Junos® OS

Multicast Protocols Feature Guide

Modified: 2018-06-25

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS Multicast Protocols Feature Guide

END USER LICENSE AGREEMENT

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at https://www.juniper.net/documentation/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at https://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- ACX Series
- M Series
- MX Series
- PTX Series
- SRX Series
- vSRX
- T Series
- EX Series
- QFX Series

- QFabric System
- NFX Series
- OCX Series

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   ```

```
load complete
```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   ```
   [edit]
   user@host# edit system scripts
   [edit system scripts]
   ```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   ```
   [edit system scripts]
   user@host# load merge relative /var/tmp/ex-script-snippet.conf
   load complete
   ```

For more information about the **load** command, see CLI Explorer.

Documentation Conventions

Table 1 on page xlii defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xlii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*

Table 2: Text and Syntax Conventions (continued)

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		  route default {
		  nexthop *address*;
		  retain;
		  }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page of the Juniper Networks TechLibrary site at https://www.juniper.net/documentation/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/documentation/feedback/.

- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or Partner Support Service support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at https://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit https://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: https://www.juniper.net/customers/support/
- Search for known bugs: https://prsearch.juniper.net/
- Find product documentation: https://www.juniper.net/documentation/
- Find solutions and answer questions using our Knowledge Base: https://kb.juniper.net/
- Download the latest versions of software and review release notes: https://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: https://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: https://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://entitlementsearch.juniper.net/entitlementsearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at https://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see https://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Understanding Multicast

Multicast Overview

IP has three fundamental types of addresses: unicast, broadcast, and multicast. A *unicast address* is used to send a packet to a single destination. A *broadcast address* is used to send a datagram to an entire subnetwork. A *multicast address* is used to send a datagram to a set of hosts that can be on different subnetworks and that are configured as members of a multicast group.

A multicast datagram is delivered to destination group members with the same best-effort reliability as a standard unicast IP datagram. This means that multicast datagrams are not guaranteed to reach all members of a group or to arrive in the same order in which they were transmitted. The only difference between a multicast IP packet and a unicast IP packet is the presence of a group address in the IP header destination address field. Multicast addresses use the Class D address format.

NOTE: **On all SRX Series devices, reordering is not supported for multicast fragments. Reordering of unicast fragments is supported.**

Individual hosts can join or leave a multicast group at any time. There are no restrictions on the physical location or the number of members in a multicast group. A host can be a member of more than one multicast group at any time. A host does not have to belong to a group to send packets to members of a group.

Routers use a group membership protocol to learn about the presence of group members on directly attached subnetworks. When a host joins a multicast group, it transmits a group membership protocol message for the group or groups that it wants to receive and sets its IP process and network interface card to receive frames addressed to the multicast group.

Comparing Multicast to Unicast

The Junos® operating system (Junos OS) routing protocol process supports a wide variety of routing protocols. These routing protocols carry network information among routing devices not only for *unicast* traffic streams sent between one pair of clients and servers, but also for *multicast* traffic streams containing video, audio, or both, between a single server source and many client receivers. The routing protocols used for multicast differ in many key ways from unicast routing protocols.

Information is delivered over a network by three basic methods: unicast, broadcast, and multicast.

The differences among unicast, broadcast, and multicast can be summarized as follows:

- Unicast: One-to-one, from one source to one destination.
- Broadcast: One-to-all, from one source to all possible destinations.
- Multicast: One-to-many, from one source to multiple destinations expressing an interest in receiving the traffic.

NOTE: This list does not include a special category for many-to-many applications, such as online gaming or videoconferencing, where there are many sources for the same receiver and where receivers often double as sources. Many-to-many is a service model that repeatedly employs one-to-many multicast and therefore requires no unique protocol. The original multicast specification, RFC 1112, supports both the any-source multicast (ASM) many-to-many model and the source-specific multicast (SSM) one-to-many model.

With unicast traffic, many streams of IP packets that travel across networks flow from a single source, such as a website server, to a single destination such as a client PC. Unicast traffic is still the most common form of information transfer on networks.

Broadcast traffic flows from a single source to all possible destinations reachable on the network, which is usually a LAN. Broadcasting is the easiest way to make sure traffic reaches its destinations.

Television networks use broadcasting to distribute video and audio. Even if the television network is a cable television (CATV) system, the source signal reaches all possible destinations, which is the main reason that some channels' content is scrambled. Broadcasting is not feasible on the Internet because of the enormous amount of unnecessary information that would constantly arrive at each end user's device, the complexities and impact of scrambling, and related privacy issues.

Multicast traffic lies between the extremes of unicast (one source, one destination) and broadcast (one source, all destinations). Multicast is a "one source, many destinations" method of traffic distribution, meaning only the destinations that explicitly indicate their need to receive the information from a particular source receive the traffic stream.

On an IP network, because destinations (clients) do not often communicate directly with sources (servers), the routing devices between source and destination must be able to determine the topology of the network from the unicast or multicast perspective to avoid routing traffic haphazardly. Multicast routing devices replicate packets received on one input interface and send the copies out on multiple output interfaces.

In IP multicast, the source and destination are almost always hosts and not routing devices. Multicast routing devices distribute the multicast traffic across the network from source to destinations. The multicast routing device must find multicast sources on the network, send out copies of packets on several interfaces, prevent routing loops, connect interested destinations with the proper source, and keep the flow of unwanted packets to a minimum. Standard multicast routing protocols provide most of these capabilities, but some router architectures cannot send multiple copies of packets and so do not support multicasting directly.

IP Multicast Uses

Multicast allows an IP network to support more than just the unicast model of data delivery that prevailed in the early stages of the Internet. Multicast, originally defined as a host extension in RFC 1112 in 1989, provides an efficient method for delivering traffic flows that can be characterized as one-to-many or many-to-many.

Unicast traffic is not strictly limited to data applications. Telephone conversations, wireless or not, contain digital audio samples and might contain digital photographs or even video and still flow from a single source to a single destination. In the same way, multicast traffic is not strictly limited to multimedia applications. In some data applications, the flow of traffic is from a single source to many destinations that require the packets, as in a news or stock ticker service delivered to many PCs. For this reason, the term *receiver* is preferred to *listener* for multicast destinations, although both terms are common.

Network applications that can function with unicast but are better suited for multicast include collaborative groupware, teleconferencing, periodic or “push” data delivery (stock quotes, sports scores, magazines, newspapers, and advertisements), server or website replication, and distributed interactive simulation (DIS) such as war simulations or virtual reality. Any IP network concerned with reducing network resource overhead for one-to-many or many-to-many data or multimedia applications with multiple receivers benefits from multicast.

If unicast were employed by radio or news ticker services, each radio or PC would have to have a separate traffic session for each listener or viewer at a PC (this is actually the method for some Web-based services). The processing load and bandwidth consumed by the server would increase linearly as more people “tune in” to the server. This is extremely inefficient when dealing with the global scale of the Internet. Unicast places the burden of packet duplication on the server and consumes more and more backbone bandwidth as the number of users grows.

If broadcast were employed instead, the source could generate a single IP packet stream using a broadcast destination address. Although broadcast eliminates the server packet duplication issue, this is not a good solution for IP because IP broadcasts can be sent only to a single subnetwork, and IP routing devices normally isolate IP subnetworks on

separate interfaces. Even if an IP packet stream could be addressed to literally go everywhere, and there were no need to “tune” to any source at all, broadcast would be extremely inefficient because of the bandwidth strain and need for uninterested hosts to discard large numbers of packets. Broadcast places the burden of packet rejection on each host and consumes the maximum amount of backbone bandwidth.

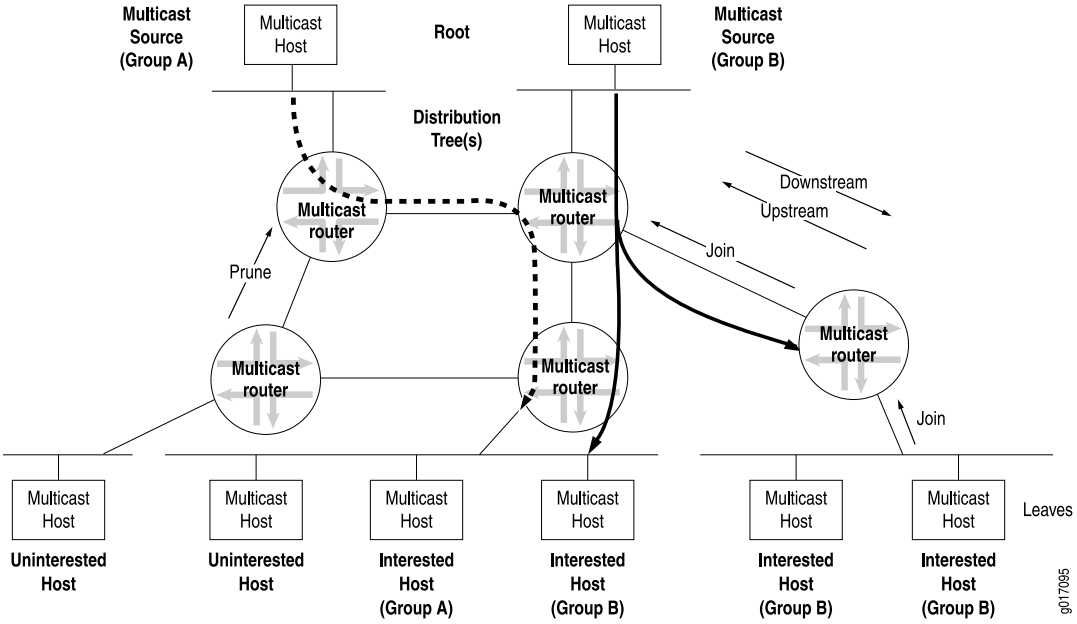
For radio station or news ticker traffic, multicast provides the most efficient and effective outcome, with none of the drawbacks and all of the advantages of the other methods. A single source of multicast packets finds its way to every *interested* receiver. As with broadcast, the transmitting host generates only a single stream of IP packets, so the load remains constant whether there is one receiver or one million. The network routing devices replicate the packets and deliver the packets to the proper receivers, but only the replication role is a new one for routing devices. The links leading to subnets consisting of entirely uninterested receivers carry no multicast traffic. Multicast minimizes the burden placed on sender, network, and receiver.

IP Multicast Terminology

Multicast has its own particular set of terms and acronyms that apply to IP multicast routing devices and networks. Figure 1 on page 7 depicts some of the terms commonly used in an IP multicast network.

In a multicast network, the key component is the *routing device*, which is able to replicate packets and is therefore multicast-capable. The routing devices in the IP multicast network, which has exactly the same topology as the unicast network it is based on, use a *multicast routing protocol* to build a *distribution tree* that connects receivers (preferred to the multimedia implications of listeners, but listeners is also used) to *sources*. In multicast terminology, the distribution tree is *rooted at the source* (the root of the distribution tree is the source). The interface on the routing device leading toward the source is the *upstream* interface, although the less precise terms *incoming* or *inbound* interface are used as well. To keep bandwidth use to a minimum, it is best for only one upstream interface on the routing device to receive multicast packets. The interface on the routing device leading toward the receivers is the *downstream* interface, although the less precise terms *outgoing* or *outbound* interface are used as well. There can be 0 to *N*–1 downstream interfaces on a routing device, where *N* is the number of logical interfaces on the routing device. To prevent looping, the upstream interface must never receive copies of downstream multicast packets.

Figure 1: Multicast Terminology in an IP Network



Routing loops are disastrous in multicast networks because of the risk of repeatedly replicated packets. One of the complexities of modern multicast routing protocols is the need to avoid routing loops, packet by packet, much more rigorously than in unicast routing protocols.

Reverse-Path Forwarding for Loop Prevention

The routing device's multicast forwarding state runs more logically based on the reverse path, from the receiver back to the root of the distribution tree. In RPF, every multicast packet received must pass an RPF check before it can be replicated or forwarded on any interface. When it receives a multicast packet on an interface, the routing device verifies that the *source* address in the multicast IP packet is the *destination* address for a unicast IP packet back to the source.

If the outgoing interface found in the unicast routing table is the same interface that the multicast packet was received on, the packet passes the RPF check. Multicast packets that fail the RPF check are dropped, because the incoming interface is not on the shortest path back to the source. Routing devices can build and maintain separate tables for RPF purposes.

Shortest-Path Tree for Loop Prevention

The distribution tree used for multicast is rooted at the source and is the shortest-path tree (SPT), but this path can be long if the source is at the periphery of the network. Providing a *shared tree* on the backbone as the distribution tree locates the multicast source more centrally in the network. Shared distribution trees with roots in the core network are created and maintained by a multicast routing device operating as a rendezvous point (RP), a feature of sparse mode multicast protocols.

Administrative Scoping for Loop Prevention

Scoping limits the routing devices and interfaces that can forward a multicast packet. Multicast scoping is *administrative* in the sense that a range of multicast addresses is reserved for scoping purposes, as described in RFC 2365, *Administratively Scoped IP Multicast*. Routing devices at the boundary must filter multicast packets and ensure that packets do not stray beyond the established limit.

Multicast Leaf and Branch Terminology

Each subnetwork with hosts on the routing device that has at least one interested receiver is a *leaf* on the distribution tree. Routing devices can have multiple leaves on different interfaces and must send a copy of the IP multicast packet out on each interface with a leaf. When a new leaf subnetwork is added to the tree (that is, the interface to the host subnetwork previously received no copies of the multicast packets), a new *branch* is built, the leaf is joined to the tree, and replicated packets are sent out on the interface. The number of leaves on a particular interface does not affect the routing device. The action is the same for one leaf or a hundred.

NOTE: On Juniper Networks security devices, if the maximum number of leaves on a multicast distribution tree is exceeded, multicast sessions are created up to the maximum number of leaves, and any multicast sessions that exceed the maximum number of leaves are ignored. The maximum number of leaves on a multicast distribution tree is device specific.

When a branch contains no leaves because there are no interested hosts on the routing device interface leading to that IP subnetwork, the branch is *pruned* from the distribution tree, and no multicast packets are sent out that interface. Packets are replicated and sent out multiple interfaces only where the distribution tree branches at a routing device, and no link ever carries a duplicate flow of packets.

Collections of hosts all receiving the same stream of IP packets, usually from the same multicast source, are called *groups*. In IP multicast networks, traffic is delivered to multicast groups based on an IP multicast address, or *group address*. The groups determine the location of the leaves, and the leaves determine the branches on the multicast network.

IP Multicast Addressing

Multicast uses the Class D IP address range (224.0.0.0 through 239.255.255.255). Class D addresses are commonly referred to as *multicast addresses* because the entire classful address concept is obsolete. Multicast addresses can never appear as the source address in an IP packet and can only be the destination of a packet.

Multicast addresses usually have a prefix length of /32, although other prefix lengths are allowed. Multicast addresses represent logical groupings of receivers and not physical collections of devices. Blocks of multicast addresses can still be described in terms of prefix length in traditional notation, but only for convenience. For example, the multicast

address range from 232.0.0.0 through 232.255.255.255 can be written as 232.0.0.0/8 or 232/8.

Internet service providers (ISPs) do not typically allocate multicast addresses to their customers because multicast addresses relate to content, not to physical devices. Receivers are not assigned their own multicast addresses, but need to know the multicast address of the content. Sources need to be assigned multicast addresses only to produce the content, not to identify their place in the network. Every source and receiver still needs an ordinary, unicast IP address.

Multicast addressing most often references the receivers, and the source of multicast content is usually not even a member of the multicast group for which it produces content. If the source needs to monitor the packets it produces, monitoring can be done locally, and there is no need to make the packets traverse the network.

Many applications have been assigned a range of multicast addresses for their own use. These applications assign multicast addresses to sessions created by that application. You do not usually need to statically assign a multicast address, but you can do so.

Multicast Addresses

Multicast host group addresses are defined to be the IP addresses whose high-order four bits are 1110, giving an address range from 224.0.0.0 through 239.255.255.255, or simply 224.0.0.0/4. (These addresses also are referred to as Class D addresses.)

The Internet Assigned Numbers Authority (IANA) maintains a list of registered IP multicast groups. The base address 224.0.0.0 is reserved and cannot be assigned to any group. The block of multicast addresses from 224.0.0.1 through 224.0.0.255 is reserved for local wire use. Groups in this range are assigned for various uses, including routing protocols and local discovery mechanisms.

The range from 239.0.0.0 through 239.255.255.255 is reserved for administratively scoped addresses. Because packets addressed to administratively scoped multicast addresses do not cross configured administrative boundaries, and because administratively scoped multicast addresses are locally assigned, these addresses do not need to be unique across administrative boundaries.

Layer 2 Frames and IPv4 Multicast Addresses

Multicasting on a LAN is a good place to start an investigation of multicasting at Layer 2. At Layer 2, multicast deals with media access control (MAC) frames and addresses instead of IPv4 or IPv6 packets and addresses. Consider a single LAN, without routing devices, with a multicast source sending to a certain group. The rest of the hosts are receivers interested in the multicast group's content. So the multicast source host generates packets with its unicast IP address as the source, and the multicast group address as the destination.

Which MAC addresses are used on the frame containing this packet? The packet source address—the unicast IP address of the host originating the multicast content—translates easily and directly to the MAC address of the source. But what about the packet's destination address? This is the IP multicast group address. Which destination MAC address for the frame corresponds to the packet's multicast group address?

One option is for LANs simply to use the LAN broadcast MAC address, which guarantees that the frame is processed by every station on the LAN. However, this procedure defeats the whole purpose of multicast, which is to limit the circulation of packets and frames to interested hosts. Also, hosts might have access to many multicast groups, which multiplies the amount of traffic to noninterested destinations. Broadcasting frames at the LAN level to support multicast groups makes no sense.

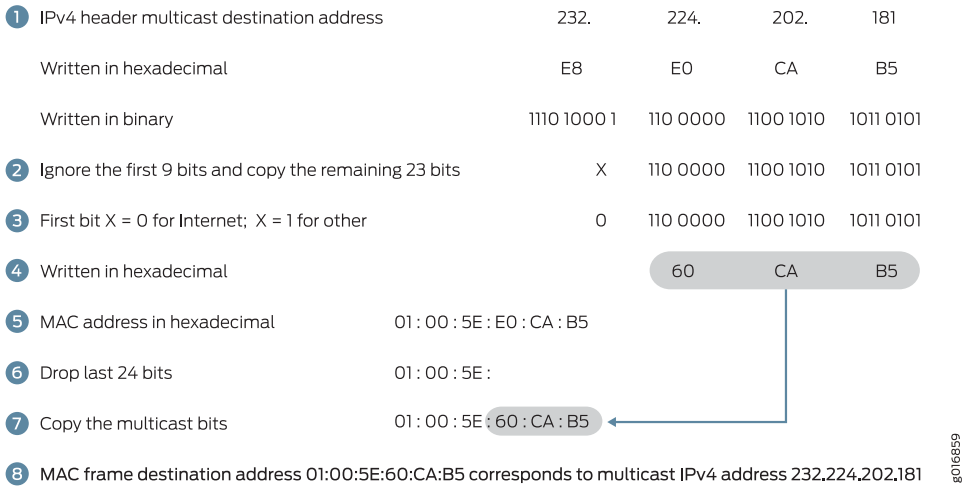
However, there is an easy way to effectively use Layer 2 frames for multicast purposes. The MAC address has a bit that is set to 0 for unicast (the LAN term is *individual address*) and set to 1 to indicate that this is a multicast address. Some of these addresses are reserved for multicast groups of specific vendors or MAC-level protocols. Internet multicast applications use the range 0x01-00-5E-00-00-00 to 0x01-00-5E-FF-FF-FF. Multicast receivers (hosts running TCP/IP) listen for frames with one of these addresses when the application joins a multicast group. The host stops listening when the application terminates or the host leaves the group at the packet layer (Layer 3).

This means that 3 bytes, or 24 bits, are available to map IPv4 multicast addresses at Layer 3 to MAC multicast addresses at Layer 2. However, all IPv4 addresses, including multicast addresses, are 32 bits long, leaving 8 IP address bits left over. Which method of mapping IPv4 multicast addresses to MAC multicast addresses minimizes the chance of “collisions” (that is, two different IP multicast groups at the packet layer mapping to the same MAC multicast address at the frame layer)?

First, it is important to realize that all IPv4 multicast addresses begin with the same 4 bits (**1110**), so there are really only 4 bits of concern, not 8. A LAN must not drop the last bits of the IPv4 address because these are almost guaranteed to be host bits, depending on the subnet mask. But the high-order bits, the leftmost address bits, are almost always network bits, and there is only one LAN (for now).

One other bit of the remaining 24 MAC address bits is reserved (an initial **0** indicates an Internet multicast address), so the 5 bits following the initial 1110 in the IPv4 address are dropped. The 23 remaining bits are mapped, one for one, into the last 23 bits of the MAC address. An example of this process is shown in Figure 2 on page 11.

Figure 2: Converting MAC Addresses to Multicast Addresses



Note that this process means that there are 32 (2^5) IPv4 multicast addresses that could map to the same MAC multicast addresses. For example, multicast IPv4 addresses 224.8.7.6 and 229.136.7.6 translate to the same MAC address (0x01-00-5E-08-07-06). This is a real concern, and because the host could be interested in frames sent to both of the those multicast groups, the IP software must reject one or the other.

NOTE: This "collision" problem does not exist in IPv6 because of the way IPv6 handles multicast groups, but it is always a concern in IPv4. The procedure for placing IPv6 multicast packets inside multicast frames is nearly identical to that for IPv4, except for the MAC destination address 0x3333 prefix (and the lack of "collisions").

Once the MAC address for the multicast group is determined, the host's operating system essentially orders the LAN interface card to join or leave the multicast group. Once joined to a multicast group, the host accepts frames sent to the multicast address as well as the host's unicast address and ignores other multicast group's frames. It is possible for a host to join and receive multicast content from more than one group at the same time, of course.

Multicast Interface Lists

To avoid multicast routing loops, every multicast routing device must always be aware of the interface that leads to the source of that multicast group content by the shortest path. This is the upstream (incoming) interface, and packets are never to be forwarded back toward a multicast source. All other interfaces are potential downstream (outgoing) interfaces, depending on the number of branches on the distribution tree.

Routing devices closely monitor the status of the incoming and outgoing interfaces, a process that determines the *multicast forwarding state*. A routing device with a multicast forwarding state for a particular multicast group is essentially "turned on" for that group's

content. Interfaces on the routing device's outgoing interface list send copies of the group's packets received on the incoming interface list for that group. The incoming and outgoing interface lists might be different for different multicast groups.

The multicast forwarding state in a routing device is usually written in either (S,G) or (*,G) notation. These are pronounced “ess comma gee” and “star comma gee,” respectively. In (S,G), the S refers to the unicast IP address of the source for the multicast traffic, and the G refers to the particular multicast group IP address for which S is the source. All multicast packets sent from this source have S as the source address and G as the destination address.

The asterisk (*) in the (*,G) notation is a wildcard indicating that the state applies to any multicast application source sending to group G. So, if two sources are originating exactly the same content for multicast group 224.1.1.2, a routing device could use (*,224.1.1.2) to represent the state of a routing device forwarding traffic from both sources to the group.

Multicast Routing Protocols

Multicast routing protocols enable a collection of multicast routing devices to build (join) distribution trees when a host on a directly attached subnet, typically a LAN, wants to receive traffic from a certain multicast group, prune branches, locate sources and groups, and prevent routing loops.

There are several multicast routing protocols:

- Distance Vector Multicast Routing Protocol (DVMRP)—The first of the multicast routing protocols and hampered by a number of limitations that make this method unattractive for large-scale Internet use. DVMRP is a dense-mode-only protocol, and uses the flood-and-prune or implicit join method to deliver traffic everywhere and then determine where the uninterested receivers are. DVMRP uses source-based distribution trees in the form (S,G), and builds its own multicast routing tables for RPF checks.
- Multicast OSPF (MOSPF)—Extends OSPF for multicast use, but only for dense mode. However, MOSPF has an explicit join message, so routing devices do not have to flood their entire domain with multicast traffic from every source. MOSPF uses source-based distribution trees in the form (S,G).
- *Bidirectional PIM mode*—A variation of PIM. Bidirectional PIM builds bidirectional shared trees that are rooted at a rendezvous point (RP) address. Bidirectional traffic does not switch to shortest path trees as in PIM-SM and is therefore optimized for routing state size instead of path length. This means that the end-to-end latency might be longer compared to PIM sparse mode. Bidirectional PIM routes are always wildcard-source (*,G) routes. The protocol eliminates the need for (S,G) routes and data-triggered events. The bidirectional (*,G) group trees carry traffic both upstream from senders toward the RP, and downstream from the RP to receivers. As a consequence, the strict reverse path forwarding (RPF)-based rules found in other PIM modes do not apply to bidirectional PIM. Instead, bidirectional PIM (*,G) routes forward traffic from all sources and the RP. Bidirectional PIM routing devices must have the ability to accept traffic on many potential incoming interfaces. Bidirectional PIM scales well because it needs no source-specific (S,G) state. Bidirectional PIM is recommended in deployments with many dispersed sources and many dispersed receivers.

- PIM *dense mode*—In this mode of PIM, the assumption is that almost all possible subnets have at least one receiver wanting to receive the multicast traffic from a source, so the network is *flooded* with traffic on all possible branches, then pruned back when branches do not express an interest in receiving the packets, explicitly (by message) or implicitly (time-out silence). This is the *dense mode* of multicast operation. LANs are appropriate networks for dense-mode operation. Some multicast routing protocols, especially older ones, support only dense-mode operation, which makes them inappropriate for use on the Internet. In contrast to DVMRP and MOSPF, PIM dense mode allows a routing device to use any unicast routing protocol and performs RPF checks using the unicast routing table. PIM dense mode has an implicit join message, so routing devices use the flood-and-prune method to deliver traffic everywhere and then determine where the uninterested receivers are. PIM dense mode uses source-based distribution trees in the form (S,G), as do all dense-mode protocols. PIM also supports sparse-dense mode, with mixed sparse and dense groups, but there is no special notation for that operational mode. If *sparse-dense mode* is supported, the multicast routing protocol allows some multicast groups to be sparse and other groups to be dense.
- PIM *sparse mode*—In this mode of PIM, the assumption is that very few of the possible receivers want packets from each source, so the network establishes and sends packets only on branches that have at least one leaf indicating (by message) an interest in the traffic. This multicast protocol allows a routing device to use any unicast routing protocol and performs reverse-path forwarding (RPF) checks using the unicast routing table. PIM sparse mode has an *explicit* join message, so routing devices determine where the interested receivers are and send join messages upstream to their neighbors, building trees from receivers to the rendezvous point (RP). PIM sparse mode uses an RP routing device as the initial source of multicast group traffic and therefore builds distribution trees in the form (*,G), as do all sparse-mode protocols. PIM sparse mode migrates to an (S,G) source-based tree if that path is shorter than through the RP for a particular multicast group's traffic. WANs are appropriate networks for sparse-mode operation, and indeed a common multicast guideline is not to run dense mode on a WAN under any circumstances.
- Core Based Trees (CBT)—Shares all of the characteristics of PIM sparse mode (sparse mode, explicit join, and shared (*,G) trees), but is said to be more efficient at finding sources than PIM sparse mode. CBT is rarely encountered outside academic discussions. There are no large-scale deployments of CBT, commercial or otherwise.
- PIM source-specific multicast (SSM)—Enhancement to PIM sparse mode that allows a client to receive multicast traffic directly from the source, without the help of an RP. Used with IGMPv3 to create a shortest-path tree between receiver and source.
- IGMPv1—The original protocol defined in RFC 1112, *Host Extensions for IP Multicasting*. IGMPv1 sends an explicit join message to the routing device, but uses a timeout to determine when hosts leave a group. Three versions of the Internet Group Management Protocol (IGMP) run between receiver hosts and routing devices.
- IGMPv2—Defined in RFC 2236, *Internet Group Management Protocol, Version 2*. Among other features, IGMPv2 adds an explicit leave message to the join message.
- IGMPv3—Defined in RFC 3376, *Internet Group Management Protocol, Version 3.* Among other features, IGMPv3 optimizes support for a single source of content for a multicast

group, or source-specific multicast (SSM). Used with PIM SSM to create a shortest-path tree between receiver and source.

- Bootstrap Router (BSR) and Auto-Rendezvous Point (RP)—Allow sparse-mode routing protocols to find RPs within the routing domain (autonomous system, or AS). RP addresses can also be statically configured.
- Multicast Source Discovery Protocol (MSDP)—Allows groups located in one multicast routing domain to find RPs in other routing domains. MSDP is not used on an RP if all receivers and sources are located in the same routing domain. Typically runs on the same routing device as PIM sparse mode RP. Not appropriate if all receivers and sources are located in the same routing domain.
- Session Announcement Protocol (SAP) and Session Description Protocol (SDP)—Display multicast session names and correlate the names with multicast traffic. SDP is a session directory protocol that advertises multimedia conference sessions and communicates setup information to participants who want to join the session. A client commonly uses SDP to announce a conference session by periodically multicasting an announcement packet to a well-known multicast address and port using SAP.
- Pragmatic General Multicast (PGM)—Special protocol layer for multicast traffic that can be used between the IP layer and the multicast application to add reliability to multicast traffic. PGM allows a receiver to detect missing information in all cases and request replacement information if the receiver application requires it.

The differences among the multicast routing protocols are summarized in Table 3 on page 14.

Table 3: Multicast Routing Protocols Compared

Multicast Routing Protocol	Dense Mode	Sparse Mode	Implicit Join	Explicit Join	(S,G) SBT	(*,G) Shared Tree
DVMRP	Yes	No	Yes	No	Yes	No
MOSPF	Yes	No	No	Yes	Yes	No
PIM dense mode	Yes	No	Yes	No	Yes	No
PIM sparse mode	No	Yes	No	Yes	Yes, maybe	Yes, initially
Bidirectional PIM	No	No	No	Yes	No	Yes
CBT	No	Yes	No	Yes	No	Yes
SSM	No	Yes	No	Yes	Yes, maybe	Yes, initially
IGMPv1	No	Yes	No	Yes	Yes, maybe	Yes, initially
IGMPv2	No	Yes	No	Yes	Yes, maybe	Yes, initially
IGMPv3	No	Yes	No	Yes	Yes, maybe	Yes, initially

Table 3: Multicast Routing Protocols Compared (continued)

Multicast Routing Protocol	Dense Mode	Sparse Mode	Implicit Join	Explicit Join	(S,G) SBT	(*,G) Shared Tree
BSR and Auto-RP	No	Yes	No	Yes	Yes, maybe	Yes, initially
MSDP	No	Yes	No	Yes	Yes, maybe	Yes, initially

It is important to realize that retransmissions due to a high bit-error rate on a link or overloaded routing device can make multicast as inefficient as repeated unicast. Therefore, there is a trade-off in many multicast applications regarding the session support provided by the Transmission Control Protocol (TCP) (but TCP always resends missing segments), or the simple drop-and-continue strategy of the User Datagram Protocol (UDP) datagram service (but reordering can become an issue). Modern multicast uses UDP almost exclusively.

T Series Router Multicast Performance

The Juniper Networks T Series Core Routers handle extreme multicast packet replication requirements with a minimum of router load. Each memory component replicates a multicast packet twice at most. Even in the worst-case scenario involving maximum fan-out, when 1 input port and 63 output ports need a copy of the packet, the T Series routing platform copies a multicast packet only six times. Most multicast distribution trees are much sparser, so in many cases only two or three replications are necessary. In no case does the T Series architecture have an impact on multicast performance, even with the largest multicast fan-out requirements.

Understanding Layer 3 Multicast Functionality on the SRX5K-MPC

Multicast is a “one source, many destinations” method of traffic distribution, meaning that only the destinations that explicitly indicate their need to receive the information from a particular source receive the traffic stream.

In the data plane of the SRX Series chassis, the SRX5000 line Module Port Concentrator (SRX5K-MPC) forwards Layer 3 IP multicast data packets, which include multicast protocol packets (for example, MLD, IGMP and PIM packets), and the data packets.

In incoming direction, the MPC receives multicast packets from an interface and forwards them to the central point or to a Services Processing Unit (SPU). The SPU performs multicast route lookup, flow-based security check, and packet replication.

In outgoing direction, the MPC receives copies of a multicast packet or Layer 3 multicast control protocol packets from SPU, and transmits them to either multicast capable routers or to hosts in a multicast group.

In the SRX Series chassis, the SPU perform multicast route lookup, if available, to forward an incoming multicast packet and replicates it for each multicast outgoing interface. After receiving replicated multicast packets and their corresponding outgoing interface information from the SPU, the MPC transmits these packets to next hops.

NOTE: On all SRX Series devices, during RG1 failover with multicast traffic and high number of multicast sessions, the failover delay is from 90 through 120 seconds for traffic to resume on the secondary node. The delay of 90 through 120 seconds is only for the first failover. For subsequent failovers, the traffic resumes within 8 through 18 seconds.

Related Documentation

- Enabling PIM Sparse Mode on page 217

Multicast Configuration Overview

You configure a router network to support multicast applications with a related family of protocols. To use multicast, you must understand the basic components of a multicast network and their relationships, and then configure the device to act as a node in the network.

To configure the device as a node in a multicast network:

1. Determine whether the router is directly attached to any multicast sources.

   Receivers must be able to locate these sources.

2. Determine whether the router is directly attached to any multicast group receivers.

   If receivers are present, IGMP is needed.

3. Determine whether to use the sparse, dense, or sparse-dense mode of multicast operation.

   Each mode has different configuration considerations.

4. Determine the address of the rendezvous point (RP) if sparse or sparse-dense mode is used.

5. Determine whether to locate the RP with the static configuration, bootstrap router (BSR), or auto-RP method.

   See:

   - Understanding Static RP on page 237
   - Understanding the PIM Bootstrap Router on page 253
   - Understanding PIM Auto-RP on page 258

6. Determine whether to configure multicast to use its own reverse-path forwarding (RPF) routing table when configuring PIM in sparse, dense, or sparse-dense modes.

   See "Understanding Multicast Reverse Path Forwarding" on page 803

7. (Optional) Configure the SAP and SDP protocols to listen for multicast session announcements.

   See "Configuring the Session Announcement Protocol" on page 415.

8. Configure IGMP.

   See "Configuring IGMP" on page 23.

9. (Optional) Configure the PIM static RP.

   See "Configuring Static RP" on page 237.

10. (Optional) Filter PIM register messages from unauthorized groups and sources.

    See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 276 and "Example: Stopping Outgoing PIM Register Messages on a Designated Router" on page 272.

11. (Optional) Configure a PIM RPF routing table.

    See "Example: Configuring a PIM RPF Routing Table" on page 809.

Related Documentation

- Multicast Overview on page 3
- *Verifying a Multicast Configuration*

IPv6 Multicast Flow

- IPv6 Multicast Flow Overview on page 17

IPv6 Multicast Flow Overview

The IPv6 multicast flow adds or enhances the following features:

- IPv6 transit multicast which includes the following packet functions:
  - Normal packet handling
  - Fragment handling
  - Packet reordering
- Protocol-Independent Multicast version 6 (PIMv6) flow handling
- Other multicast routing protocols, such as Multicast Listener Discovery (MLD)

The structure and processing of IPv6 multicast data session are the same as those of IPv4. Each data session has the following:

- One template session
- Several leaf sessions.

The reverse path forwarding (RPF) check behavior for IPv6 is the same as that for IPv4. Incoming multicast data is accepted only if the RPF check succeeds. In an IPv6 multicast flow, incoming Multicast Listener Discovery (MLD) protocol packets are accepted only if MLD or PIM is enabled in the security zone for the incoming interface. Sessions for multicast protocol packets have a default timeout value of 300 seconds. This value cannot be configured. The null register packet is sent to rendezvous point (RP).

In IPv6 multicast flow, a multicast router has the following three roles:

- Designated router

  This router receives the multicast packets, encapsulates them with unicast IP headers, and sends them for multicast flow.

- Intermediate router

  There are two sessions for the packets, the control session, for the outer unicast packets, and the data session. The security policies are applied to the data session and the control session, is used for forwarding.

- Rendezvous point

  The RP receives the unicast PIM register packet, separates the unicast header, and then forwards the inner multicast packet. The packets received by RP are sent to the pd interface for decapsulation and are later handled like normal multicast packets.

On a Services Processing Unit (SPU), the multicast session is created as a template session for matching the incoming packet's tuple. Leaf sessions are connected to the template session. On the Customer Premise Equipment (CPE), only the template session is created. Each CPE session carries the fan-out lists that are used for load-balanced distribution of multicast SPU sessions.

NOTE: **IPv6 multicast uses the IPv4 multicast behavior for session distribution.**

The network service access point identifier (nsapi) of the leaf session is set up on the multicast text traffic going into the tunnels, to point to the outgoing tunnel. The zone ID of the tunnel is used for policy lookup for the leaf session in the second stage. Multicast packets are unidirectional. Thus for multicast text session sent into the tunnels, forwarding sessions are not created.

When the multicast route ages out, the corresponding chain of multicast sessions is deleted. When the multicast route changes, then the corresponding chain of multicast sessions is deleted. This forces the next packet hitting the multicast route to take the first path and re-create the chain of sessions; the multicast route counter is not affected.

NOTE: **The IPv6 multicast packet reorder approach is same as that for IPv4.**

For the encapsulating router, the incoming packet is multicast, and the outgoing packet is unicast. For the intermediate router, the incoming packet is unicast, and the outgoing packet is unicast.

Related Documentation

- *Multicast Protocols Feature Guide*

Supported IP Multicast Protocol Standards

Junos OS substantially supports the following RFCs and Internet drafts, which define standards for IP multicast protocols, including the Distance Vector Multicast Routing Protocol (DVMRP), Internet Group Management Protocol (IGMP), Multicast Listener Discovery (MLD), Multicast Source Discovery Protocol (MSDP), Pragmatic General Multicast (PGM), Protocol Independent Multicast (PIM), Session Announcement Protocol (SAP), and Session Description Protocol (SDP).

- RFC 1112, *Host Extensions for IP Multicasting* (defines IGMP Version 1)
- RFC 2236, *Internet Group Management Protocol, Version 2*
- RFC 2327, *SDP: Session Description Protocol*
- RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*
- RFC 2858, *Multiprotocol Extensions for BGP-4*
- RFC 3031, *Multiprotocol Label Switching Architecture*
- RFC 3376, *Internet Group Management Protocol, Version 3*
- RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*
- RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*
- RFC 4601, *Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)*
- RFC 4604, *Using IGMPv3 and MLDv2 for Source-Specific Multicast*
- RFC 4607, *Source-Specific Multicast for IP*
- RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*
- RFC 5015, *Bidirectional Protocol Independent Multicast (BIDIR-PIM)*
- RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

  The scoping mechanism is not supported.
- RFC 6513, *Multicast in MPLS/BGP IP VPNs*
- RFC 6514, *BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs*
- Internet draft draft-raggarwa-l3vpn-bgp-mvpn-extranet-08.txt, *Extranet in BGP Multicast VPN (MVPN)*
- Internet draft draft-rosen-l3vpn-spmsi-joins-mldp-03.txt, *MVPN: S-PMSI Join Extensions for mLDP-Created Tunnels*

The following RFCs and Internet drafts do not define standards, but provide information about multicast protocols and related technologies. The IETF classifies them variously as “Best Current Practice,” “Experimental,” or “Informational.”

- RFC 1075, *Distance Vector Multicast Routing Protocol*
- RFC 2362, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*
- RFC 2365, *Administratively Scoped IP Multicast*
- RFC 2547, *BGP/MPLS VPNs*
- RFC 2974, *Session Announcement Protocol*
- RFC 3208, *PGM Reliable Transport Protocol Specification*
- RFC 3446, *Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*
- RFC 3569, *An Overview of Source-Specific Multicast (SSM)*
- RFC 3618, *Multicast Source Discovery Protocol (MSDP)*
- RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
- RFC 3973, *Protocol Independent Multicast – Dense Mode (PIM-DM): Protocol Specification (Revised)*
- RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*
- Internet draft draft-ietf-idmr-dvmrp-v3-11.txt, *Distance Vector Multicast Routing Protocol*
- Internet draft draft-ietf-mboned-ssm232-08.txt, *Source-Specific Protocol Independent Multicast in 232/8*
- Internet draft draft-ietf-mmusic-sap-00.txt, *SAP: Session Announcement Protocol*
- Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP VPNs*

  Only section 7, “Data MDT: Optimizing flooding,” is supported.

Related Documentation

- *Accessing Standards Documents on the Internet*

PART 2

Managing Group Membership

CHAPTER 2

Configuring IGMP and MLD

Configuring IGMP

Understanding Group Membership Protocols

There is a big difference between the multicast protocols used between host and routing device and between the multicast routing devices themselves. Hosts on a given subnetwork need to inform their routing device only whether or not they are interested in receiving packets from a certain multicast group. The source host needs to inform its routing devices only that it is the source of traffic for a particular multicast group. In other words, no detailed knowledge of the distribution tree is needed by any hosts; only a group membership protocol is needed to inform routing devices of their participation in a multicast group. Between adjacent routing devices, on the other hand, the multicast routing protocols must avoid loops as they build a detailed sense of the network topology and distribution tree from source to leaf. So, different multicast protocols are used for the host-router portion and the router-router portion of the multicast network.

Multicast group membership protocols enable a routing device to detect when a host on a directly attached subnet, typically a LAN, wants to receive traffic from a certain multicast group. Even if more than one host on the LAN wants to receive traffic for that multicast group, the routing device sends only one copy of each packet for that multicast group out on that interface, because of the inherent broadcast nature of LANs. When the multicast group membership protocol informs the routing device that there are no interested hosts on the subnet, the packets are withheld and that leaf is pruned from the distribution tree.

The Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) Protocol are the standard IP multicast group membership protocols: IGMP and MLD have several versions that are supported by hosts and routing devices:

- IGMPv1—The original protocol defined in RFC 1112. An explicit join message is sent to the routing device, but a timeout is used to determine when hosts leave a group. This process wastes processing cycles on the routing device, especially on older or smaller routing devices.
- IGMPv2—Defined in RFC 2236. Among other features, IGMPv2 adds an explicit leave message to the join message so that routing devices can more easily determine when a group has no interested listeners on a LAN.
- IGMPv3—Defined in RFC 3376. Among other features, IGMPv3 optimizes support for a single source of content for a multicast group, or *source-specific multicast* (*SSM*).
- MLDv1—Defined in RFC 2710. MLDv1 is similar to IGMPv2.
- MLDv2—Defined in RFC 3810. MLDv2 similar to IGMPv3.

The various versions of IGMP and MLD are backward compatible. It is common for a routing device to run multiple versions of IGMP and MLD on LAN interfaces. Backward compatibility is achieved by dropping back to the most basic of all versions run on a LAN. For example, if one host is running IGMPv1, any routing device attached to the LAN running IGMPv2 can drop back to IGMPv1 operation, effectively eliminating the IGMPv2 advantages. Running multiple IGMP versions ensures that both IGMPv1 and IGMPv2 hosts find peers for their versions on the routing device.

CAUTION: On MX Series platforms, IGMPv2 and IGMPv3 can or cannot be configured together on the same interface, depending on the Junos OS release at your installation. Configuring both together can cause unexpected behavior in multicast traffic forwarding.

See Also
- Examples: Configuring MLD on page 50

Understanding IGMP

The Internet Group Management Protocol (IGMP) manages the membership of hosts and routing devices in multicast groups. IP hosts use IGMP to report their multicast group memberships to any immediately neighboring multicast routing devices. Multicast routing devices use IGMP to learn, for each of their attached physical networks, which groups have members.

IGMP is also used as the transport for several related multicast protocols (for example, Distance Vector Multicast Routing Protocol [DVMRP] and Protocol Independent Multicast version 1 [PIMv1]).

A routing device receives explicit join and prune messages from those neighboring routing devices that have downstream group members. When PIM is the multicast protocol in use, IGMP begins the process as follows:

1. To join a multicast group, G, a host conveys its membership information through IGMP.
2. The routing device then forwards data packets addressed to a multicast group G to only those interfaces on which explicit join messages have been received.
3. A designated router (DR) sends periodic join and prune messages toward a group-specific rendezvous point (RP) for each group for which it has active members. One or more routing devices are automatically or statically designated as the RP, and all routing devices must explicitly join through the RP.
4. Each routing device along the path toward the RP builds a wildcard (any-source) state for the group and sends join and prune messages toward the RP.

   The term *route entry* is used to refer to the state maintained in a routing device to represent the distribution tree.

   A route entry can include such fields as:

   - source address
   - group address
   - incoming interface from which packets are accepted
   - list of outgoing interfaces to which packets are sent
   - timers
   - flag bits

The wildcard route entry's incoming interface points toward the RP.

The outgoing interfaces point to the neighboring downstream routing devices that have sent join and prune messages toward the RP as well as the directly connected hosts that have requested membership to group G.

5. This state creates a shared, RP-centered, distribution tree that reaches all group members.

IGMP is also used as the transport for several related multicast protocols (for example, Distance Vector Multicast Routing Protocol [DVMRP] and Protocol Independent Multicast version 1 [PIMv1]).

Starting in Junos OS Release 15.2, PIMv1 is not supported.

IGMP is an integral part of IP and must be enabled on all routing devices and hosts that need to receive IP multicast traffic.

For each attached network, a multicast routing device can be either a querier or a nonquerier. The querier routing device periodically sends general query messages to solicit group membership information. Hosts on the network that are members of a multicast group send report messages. When a host leaves a group, it sends a leave group message.

IGMP version 3 (IGMPv3) supports inclusion and exclusion lists. Inclusion lists enable you to specify which sources can send to a multicast group. This type of multicast group is called a source-specific multicast (SSM) group, and its multicast address is 232/8.

IGMPv3 provides support for source filtering. For example, a routing device can specify particular routing devices from which it accepts or rejects traffic. With IGMPv3, a multicast routing device can learn which sources are of interest to neighboring routing devices.

Exclusion mode works the opposite of an inclusion list. It allows any source but the ones listed to send to the SSM group.

IGMPv3 interoperates with versions 1 and 2 of the protocol. However, to remain compatible with older IGMP hosts and routing devices, IGMPv3 routing devices must also implement versions 1 and 2 of the protocol. IGMPv3 supports the following membership-report record types: mode is allowed, allow new sources, and block old sources.

See Also

- Supported IP Multicast Protocol Standards on page 19
- Enabling IGMP on page 28
- Disabling IGMP on page 49
- Configuring IGMP on page 23

Configuring IGMP

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Configuring the Session Announcement Protocol" on page 415.

To configure the Internet Group Management Protocol (IGMP), include the **igmp** statement:

```
igmp {
    accounting;
    interface interface-name {
        disable;
        (accounting | no-accounting);
        group-policy [ policy-names ];
        immediate-leave;
        oif-map map-name;
        promiscuous-mode;
        ssm-map ssm-map-name;
        static {
            group multicast-group-address {
                exclude;
                group-count number;
                group-increment increment;
                source ip-address {
                    source-count number;
                    source-increment increment;
                }
            }
        }
        version version;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
```

```
        flag flag <flag-modifier> <disable>;
    }
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**

By default, IGMP is enabled on all interfaces on which you configure Protocol Independent Multicast (PIM), and on all broadcast interfaces on which you configure the Distance Vector Multicast Routing Protocol (DVMRP).

NOTE: You can configure IGMP on an interface without configuring PIM. PIM is generally not needed on IGMP downstream interfaces. Therefore, only one "pseudo PIM interface" is created to represent all IGMP downstream (IGMP-only) interfaces on the router. This reduces the amount of router resources, such as memory, that are consumed. You must configure PIM on upstream IGMP interfaces to enable multicast routing, perform reverse-path forwarding for multicast data packets, populate the multicast forwarding table for upstream interfaces, and in the case of bidirectional PIM and PIM sparse mode, to distribute IGMP group memberships into the multicast routing domain.

Enabling IGMP

The Internet Group Management Protocol (IGMP) manages multicast groups by establishing, maintaining, and removing groups on a subnet. Multicast routing devices use IGMP to learn which groups have members on each of their attached physical networks. IGMP must be enabled for the router to receive IPv4 multicast packets. IGMP is only needed for IPv4 networks, because multicast is handled differently in IPv6 networks. IGMP is automatically enabled on all IPv4 interfaces on which you configure PIM and on all IPv4 broadcast interfaces when you configure DVMRP.

If IGMP is not running on an interface—either because PIM and DVMRP are not configured on the interface or because IGMP is explicitly disabled on the interface—you can explicitly enable IGMP.

To explicitly enable IGMP:

1. If PIM and DVMRP are not running on the interface, explicitly enable IGMP by including the interface name.

   ```
   [edit protocols igmp]
   user@host# set interface fe-0/0/0.0
   ```

2. See if IGMP is disabled on any interfaces. In the following example, IGMP is disabled on a Gigabit Ethernet interface.

   ```
   [edit protocols igmp]
   ```

```
user@host# show
interface fe-0/0/0.0;
interface ge-1/0/0.0 {
  disable;
}
```

3. Enable IGMP on the interface by deleting the **disable** statement.

```
[edit protocols igmp]
delete interface ge-1/0/0.0 disable
```

4. Verify the configuration.

```
[edit protocols igmp]
user@host# show
interface fe-0/0/0.0;
interface ge-1/0/0.0;
```

5. Verify the operation of IGMP on the interfaces by checking the output of the **show igmp interface** command.

See Also

- Understanding IGMP on page 25
- Disabling IGMP on page 49
- show igmp interface on page 1470

Modifying the IGMP Host-Query Message Interval

The objective of IGMP is to keep routers up to date with group membership of the entire subnet. Routers need not know who all the members are, only that members exist. Each host keeps track of which multicast groups are subscribed to. On each link, one router is elected the querier. The IGMP querier router periodically sends general host-query messages on each attached network to solicit membership information. The messages are sent to the all-systems multicast group address, 224.0.0.1.

The query interval, the response interval, and the robustness variable are related in that they are all variables that are used to calculate the group membership timeout. The group membership timeout is the number of seconds that must pass before a multicast router determines that no more members of a host group exist on a subnet. The group membership timeout is calculated as the (robustness variable x query-interval) + (query-response-interval). If no reports are received for a particular group before the group membership timeout has expired, the routing device stops forwarding remotely-originated multicast packets for that group onto the attached network.

By default, host-query messages are sent every 125 seconds. You can change this interval to change the number of IGMP messages sent on the subnet.

To modify the query interval:

1. Configure the interval.

```
[edit protocols igmp]
user@host# set query-interval 200
```

The value can be from 1 through 1024 seconds.

2. Verify the configuration by checking the IGMP Query Interval field in the output of the **show igmp interface** command.

3. Verify the operation of the query interval by checking the Membership Query field in the output of the **show igmp statistics** command.

See Also

- Understanding IGMP on page 25
- Modifying the IGMP Query Response Interval on page 30
- Modifying the IGMP Robustness Variable on page 34
- **show igmp interface on page 1470**
- *show igmp statistics*

Modifying the IGMP Query Response Interval

The query response interval is the maximum amount of time that can elapse between when the querier router sends a host-query message and when it receives a response from a host. Configuring this interval allows you to adjust the burst peaks of IGMP messages on the subnet. Set a larger interval to make the traffic less bursty. Bursty traffic refers to an uneven pattern of data transmission: sometimes a very high data transmission rate, whereas at other times a very low data transmission rate.

The query response interval, the host-query interval, and the robustness variable are related in that they are all variables that are used to calculate the group membership timeout. The group membership timeout is the number of seconds that must pass before a multicast router determines that no more members of a host group exist on a subnet. The group membership timeout is calculated as the (robustness variable x query-interval) + (query-response-interval). If no reports are received for a particular group before the group membership timeout has expired, the routing device stops forwarding remotely originated multicast packets for that group onto the attached network.

The default query response interval is 10 seconds. You can configure a subsecond interval up to one digit to the right of the decimal point. The configurable range is 0.1 through 0.9, then in 1-second intervals 1 through 999,999.

To modify the query response interval:

1. Configure the interval.

```
[edit protocols igmp]
user@host# set query-response-interval 0.4
```

2. Verify the configuration by checking the IGMP Query Response Interval field in the output of the **show igmp interface** command.

3. Verify the operation of the query interval by checking the Membership Query field in the output of the **show igmp statistics** command.

See Also

- Understanding IGMP on page 25
- Modifying the IGMP Host-Query Message Interval on page 29
- Modifying the IGMP Robustness Variable on page 34
- **show igmp interface on page 1470**
- *show igmp statistics*

Specifying Immediate-Leave Host Removal for IGMP

The immediate leave setting is useful for minimizing the leave latency of IGMP memberships. When this setting is enabled, the routing device leaves the multicast group immediately after the last host leaves the multicast group.

The immediate-leave setting enables host tracking, meaning that the device keeps track of the hosts that send join messages. This allows IGMP to determine when the last host sends a leave message for the multicast group.

When the immediate leave setting is enabled, the device removes an interface from the forwarding-table entry without first sending IGMP group-specific queries to the interface. The interface is pruned from the multicast tree for the multicast group specified in the IGMP leave message. The immediate leave setting ensures optimal bandwidth management for hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When immediate leave is disabled and one host sends a leave group message, the routing device first sends a group query to determine if another receiver responds. If no receiver responds, the routing device removes all hosts on the interface from the multicast group. Immediate leave is disabled by default for both IGMP version 2 and IGMP version 3.

NOTE: Although host tracking is enabled for IGMPv2 and MLDv1 when you enable immediate leave, use immediate leave with these versions only when there is one host on the interface. The reason is that IGMPv2 and MLDv1 use a report suppression mechanism whereby only one host on an interface sends a group join report in response to a membership query. The other interested hosts suppress their reports. The purpose of this mechanism is to avoid a flood of reports for the same group. But it also interferes with host tracking, because the router only knows about the one interested host and does not know about the others.

To enable immediate leave on an interface:

1. Configure immediate leave on the IGMP interface.

```
[edit protocols IGMP]
user@host# set interface ge-0/0/0.1 immediate-leave
```

2. Verify the configuration by checking the Immediate Leave field in the output of the **show igmp interface** command.

See Also
- Understanding IGMP on page 25
- show igmp interface on page 1470

Filtering Unwanted IGMP Reports at the IGMP Interface Level

Suppose you need to limit the subnets that can join a certain multicast group. The **group-policy** statement enables you to filter unwanted IGMP reports at the interface level. When this statement is enabled on a router running IGMP version 2 (IGMPv2) or version 3 (IGMPv3), after the router receives an IGMP report, the router compares the group against the specified group policy and performs the action configured in that policy (for example, rejects the report if the policy matches the defined address or network).

You define the policy to match only IGMP group addresses (for IGMPv2) by using the policy's **route-filter** statement to match the group address. You define the policy to match IGMP (source, group) addresses (for IGMPv3) by using the policy's **route-filter** statement to match the group address and the policy's **source-address-filter** statement to match the source address.

CAUTION: On MX Series platforms, IGMPv2 and IGMPv3 can or cannot be configured together on the same interface, depending on the Junos OS release at your installation. Configuring both together can cause unexpected behavior in multicast traffic forwarding.

To filter unwanted IGMP reports:

1. Configure an IGMPv2 policy.

```
[edit policy-statement reject_policy_v2]
user@host# set from route-filter 233.252.0.1/32 exact
user@host# set from route-filter 239.0.0.0/8 orlonger
user@host# set then reject
```

2. Configure an IGMPv3 policy.

```
[edit policy-statement reject_policy_v3]
user@host# set from route-filter 233.252.0.1/32 exact
user@host# set from route-filter 239.0.0.0/8 orlonger
user@host# set from source-address-filter 10.0.0.0/8 orlonger
user@host# set from source-address-filter 127.0.0.0/8 orlonger
user@host# set then reject
```

3. Apply the policies to the IGMP interfaces on which you prefer not to receive specific group or (source, group) reports. In this example, **ge-0/0/0.1** is running IGMPv2, and **ge-0/1/1.0** is running IGMPv3.

```
[edit protocols igmp]
```

```
user@host# set interface ge-0/0/0.1 group-policy reject_policy_v2
user@host# set interface ge-0/1/1.0 group-policy reject_policy_v3
```

4. Verify the operation of the filter by checking the Rejected Report field in the output of the **show igmp statistics** command.

See Also

- Understanding IGMP on page 25
- *Example: Configuring Policy Chains and Route Filters*
- *show igmp statistics*

Accepting IGMP Messages from Remote Subnetworks

By default, IGMP interfaces accept IGMP messages only from the same subnet. Including the **promiscuous-mode** statement enables the routing device to accept IGMP messages from indirectly connected subnets.

NOTE: When you enable IGMP on an unnumbered Ethernet interface that uses a /32 loopback address as a donor address, you must configure IGMP promiscuous mode to accept the IGMP packets received on this interface.

NOTE: When enabling promiscuous-mode, all routers on the ethernet segment must be configured with the promiscuous mode statement. Otherwise, only the interface configured with lowest IPv4 address acts as the querier for IGMP for this Ethernet segment.

To enable IGMP promiscuous mode on an interface:

1. Configure the IGMP interface.

```
[edit protocols igmp]
user@host# set interface ge-0/1/1.0 promiscuous-mode
```

2. Verify the configuration by checking the Promiscuous Mode field in the output of the **show igmp interface** command.

3. Verify the operation of the filter by checking the Rx non-local field in the output of the **show igmp statistics** command.

See Also

- Understanding IGMP on page 25
- *Configuring the Loopback Interface* in the *Junos OS Network Interfaces Library for Routing Devices*
- show igmp interface on page 1470

- *show igmp statistics*

Modifying the IGMP Last-Member Query Interval

The last-member query interval is the maximum amount of time between group-specific query messages, including those sent in response to leave-group messages. You can configure this interval to change the amount of time it takes a routing device to detect the loss of the last member of a group.

When the routing device that is serving as the querier receives a leave-group message from a host, the routing device sends multiple group-specific queries to the group being left. The querier sends a specific number of these queries at a specific interval. The number of queries sent is called the last-member query count. The interval at which the queries are sent is called the last-member query interval. Because both settings are configurable, you can adjust the leave latency. The IGMP leave latency is the time between a request to leave a multicast group and the receipt of the last byte of data for the multicast group.

The last-member query count x (times) the last-member query interval = (equals) the amount of time it takes a routing device to determine that the last member of a group has left the group and to stop forwarding group traffic.

The default last-member query interval is 1 second. You can configure a subsecond interval up to one digit to the right of the decimal point. The configurable range is 0.1 through 0.9, then in 1-second intervals 1 through 999,999.

To modify this interval:

1. Configure the time (in seconds) that the routing device waits for a report in response to a group-specific query.

   ```
   [edit protocols igmp]
   user@host# set query-last-member-interval 0.1
   ```

2. Verify the configuration by checking the IGMP Last Member Query Interval field in the output of the **show igmp interfaces** command.

NOTE: You can configure the last-member query count by configuring the robustness variable. The two are always equal.

See Also

- Modifying the IGMP Robustness Variable on page 34
- show pim interfaces on page 1639

Modifying the IGMP Robustness Variable

Fine-tune the IGMP robustness variable to allow for expected packet loss on a subnet. The robust count automatically changes certain IGMP message intervals for IGMPv2 and

IGMPv3. Increasing the robust count allows for more packet loss but increases the leave latency of the subnetwork.

When the query router receives an IGMP leave message on a shared network running IGMPv2, the query router must send an IGMP group query message a specified number of times. The number of IGMP group query messages sent is determined by the robust count.

The value of the robustness variable is also used in calculating the following IGMP message intervals:

- Group member interval—Amount of time that must pass before a multicast router determines that there are no more members of a group on a network. This interval is calculated as follows: (robustness variable x query-interval) + (1 x query-response-interval).
- Other querier present interval—The robust count is used to calculate the amount of time that must pass before a multicast router determines that there is no longer another multicast router that is the querier. This interval is calculated as follows: (robustness variable x query-interval) + (0.5 x query-response-interval).
- Last-member query count—Number of group-specific queries sent before the router assumes there are no local members of a group. The number of queries is equal to the value of the robustness variable.

In IGMPv3, a change of interface state causes the system to immediately transmit a state-change report from that interface. In case the state-change report is missed by one or more multicast routers, it is retransmitted. The number of times it is retransmitted is the robust count minus one. In IGMPv3, the robust count is also a factor in determining the group membership interval, the older version querier interval, and the other querier present interval.

By default, the robustness variable is set to 2. You might want to increase this value if you expect a subnet to lose packets.

The number can be from 2 through 10.

To change the value of the robustness variable:

1. Configure the robust count.

   When you set the robust count, you are in effect configuring the number of times the querier retries queries on the connected subnets.

   ```
   [edit protocols igmp]
   user@host# set robust-count 5
   ```

2. Verify the configuration by checking the IGMP Robustness Count field in the output of the **show igmp interfaces** command.

See Also

- Modifying the IGMP Host-Query Message Interval on page 29
- Modifying the IGMP Query Response Interval on page 30

- Modifying the IGMP Last-Member Query Interval on page 34
- show pim interfaces on page 1639
- RFC 2236, *Internet Group Management Protocol, Version 2*
- RFC 3376, *Internet Group Management Protocol, Version 3*

Limiting the Maximum IGMP Message Rate

This section describes how to change the limit for the maximum number of IGMP packets transmitted in 1 second by the router.

Increasing the maximum number of IGMP packets transmitted per second might be useful on a router with a large number of interfaces participating in IGMP.

To change the limit for the maximum number of IGMP packets the router can transmit in 1 second, include the **maximum-transmit-rate** statement and specify the maximum number of packets per second to be transmitted.

See Also
- maximum-transmit-rate (Protocols IGMP) on page 1143

Changing the IGMP Version

By default, the routing device runs IGMPv2. Routing devices running different versions of IGMP determine the lowest common version of IGMP that is supported by hosts on their subnet and operate in that version.

To enable source-specific multicast (SSM) functionality, you must configure version 3 on the host and the host's directly connected routing device. If a source address is specified in a multicast group that is statically configured, the version must be set to IGMPv3.

If a static multicast group is configured with the source address defined, and the IGMP version is configured to be version 2, the source is ignored and only the group is added. In this case, the join is treated as an IGMPv2 group join.

BEST PRACTICE: If you configure the IGMP version setting at the individual interface hierarchy level, it overrides the **interface all** statement. That is, the new interface does not inherit the version number that you specified with the **interface all** statement. By default, that new interface is enabled with **version 2**. You must explicitly specify a **version** *number* when adding a new interface. For example, if you specified **version 3** with **interface all**, you would need to configure the **version 3** statement for the new interface. Additionally, if you configure an interface for a multicast group at the **[edit interface** *interface-name* **static group** *multicast-group-address***]** hierarchy level, you must specify a **version** *number* as well as the other group parameters. Otherwise, the interface is enabled with the default **version 2**.

If you have already configured the routing device to use IGMP version 1 (IGMPv1) and then configure it to use IGMPv2, the routing device continues to use IGMPv1 for up to 6 minutes and then uses IGMPv2.

To change to IGMPv3 for SSM functionality:

1. Configure the IGMP interface.

   ```
   [edit protocols igmp]
   user@host# set interface ge-0/0/0 version 3
   ```

2. Verify the configuration by checking the version field in the output of the **show igmp interfaces** command. The **show igmp statistics** command has version-specific output fields, such as V1 Membership Report, V2 Membership Report, and V3 Membership Report.

CAUTION: On MX Series platforms, IGMPv2 and IGMPv3 can or cannot be configured together on the same interface, depending on the Junos OS release at your installation. Configuring both together can cause unexpected behavior in multicast traffic forwarding.

See Also

- Understanding IGMP on page 25
- show pim interfaces on page 1639
- *show igmp statistics*
- RFC 2236, *Internet Group Management Protocol, Version 2*
- RFC 3376, *Internet Group Management Protocol, Version 3*

Enabling IGMP Static Group Membership

You can create IGMP static group membership to test multicast forwarding without a receiver host. When you enable IGMP static group membership, data is forwarded to an interface without that interface receiving membership reports from downstream hosts. The router on which you enable static IGMP group membership must be the designated router (DR) for the subnet. Otherwise, traffic does not flow downstream.

When enabling IGMP static group membership, you cannot configure multiple groups using the **group-count**, **group-increment**, **source-count**, and **source-increment** statements if the **all** option is specified as the IGMP interface.

Class-of-service (CoS) adjustment is not supported with IGMP static group membership.

In this example, you create static group 233.252.0.1.

1. On the DR, configure the static groups to be created by including the **static** statement and **group** statement and specifying which IP multicast address of the group to be

created. When creating groups individually, you must specify a unique address for each group.

```
[edit protocols igmp]
user@host# set interface fe-0/1/2 static group 233.252.0.1
```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

```
user@host> show configuration protocol igmp

interface fe-0/1/2.0 {
  static {
    group 233.252.0.1 ;
  }
}
```

3. After you have committed the configuration and the source is sending traffic, use the **show igmp group** command to verify that static group 233.252.0.1 has been created.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 233.252.0.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
```

NOTE: When you configure static IGMP group entries on point-to-point links that connect routing devices to a rendezvous point (RP), the static IGMP group entries do not generate join messages toward the RP.

When you create IGMP static group membership to test multicast forwarding on an interface on which you want to receive multicast traffic, you can specify that a number of static groups be automatically created. This is useful when you want to test forwarding to multiple receivers without having to configure each receiver separately.

In this example, you create three groups.

1. On the DR, configure the number of static groups to be created by including the **group-count** statement and specifying the number of groups to be created.

```
[edit protocols igmp]
user@host# set interface fe-0/1/2 static group 233.252.0.1 group-count 3
```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

```
user@host> show configuration protocol igmp

interface fe-0/1/2.0 {
  static {
    group 233.252.0.1 {
      group-count 3;
```

```
    }
  }
}
```

3. After you have committed the configuration and after the source is sending traffic, use the **show igmp group** command to verify that static groups 233.252.0.1, 233.252.0.2, and 233.252.0.3 have been created.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 233.252.0.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 233.252.0.2
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 233.252.0.3
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
```

When you create IGMP static group membership to test multicast forwarding on an interface on which you want to receive multicast traffic, you can also configure the group address to be automatically incremented for each group created. This is useful when you want to test forwarding to multiple receivers without having to configure each receiver separately and when you do not want the group addresses to be sequential.

In this example, you create three groups and increase the group address by an increment of two for each group.

1. On the DR, configure the group address increment by including the **group-increment** statement and specifying the number by which the address should be incremented for each group. The increment is specified in dotted decimal notation similar to an IPv4 address.

```
[edit protocols igmp]
user@host# set interface fe-0/1/2 static group 233.252.0.1 group-count 3
  group-increment 0.0.0.2
```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

```
user@host> show configuration protocol igmp

interface fe-0/1/2.0 {
  version 3;
  static {
    group 233.252.0.1 {
      group-increment 0.0.0.2;
      group-count 3;
    }
  }
}
```

3. After you have committed the configuration and after the source is sending traffic, use the **show igmp group** command to verify that static groups 233.252.0.1, 233.252.0.3, and 233.252.0.5 have been created.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 233.252.0.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 233.252.0.3
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 233.252.0.5
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
```

When you create IGMP static group membership to test multicast forwarding on an interface on which you want to receive multicast traffic, and your network is operating in source-specific multicast (SSM) mode, you can also specify that the multicast source address be accepted. This is useful when you want to test forwarding to multicast receivers from a specific multicast source.

If you specify a group address in the SSM range, you must also specify a source.

If a source address is specified in a multicast group that is statically configured, the IGMP version on the interface must be set to IGMPv3. IGMPv2 is the default value.

In this example, you create group 233.252.0.1 and accept IP address 10.0.0.2 as the only source.

1. On the DR, configure the source address by including the **source** statement and specifying the IPv4 address of the source host.

```
[edit protocols igmp]
user@host# set interface fe-0/1/2 static group 233.252.0.1 source 10.0.0.2
```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

```
user@host> show configuration protocol igmp

interface fe-0/1/2.0 {
  version 3;
  static {
    group 233.252.0.1 {
      source 10.0.0.2;
    }
  }
}
```

3. After you have committed the configuration and the source is sending traffic, use the **show igmp group** command to verify that static group 233.252.0.1 has been created and that source 10.0.0.2 has been accepted.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 233.252.0.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
```

When you create IGMP static group membership to test multicast forwarding on an interface on which you want to receive multicast traffic, you can specify that a number of multicast sources be automatically accepted. This is useful when you want to test forwarding to multicast receivers from more than one specified multicast source.

In this example, you create group 233.252.0.1 and accept addresses 10.0.0.2, 10.0.0.3, and 10.0.0.4 as the sources.

1. On the DR, configure the number of multicast source addresses to be accepted by including the **source-count** statement and specifying the number of sources to be accepted.

```
[edit protocols igmp]
user@host# set interface fe-0/1/2 static group 233.252.0.1 source 10.0.0.2 source-count
  3
```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

```
user@host> show configuration protocol igmp

interface fe-0/1/2.0 {
  version 3;
  static {
    group 233.252.0.1 {
      source 10.0.0.2 {
        source-count 3;
      }
    }
  }
}
```

3. After you have committed the configuration and the source is sending traffic, use the **show igmp group** command to verify that static group 233.252.0.1 has been created and that sources 10.0.0.2, 10.0.0.3, and 10.0.0.4 have been accepted.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 233.252.0.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 233.252.0.1
         Source: 10.0.0.3
```

```
        Last reported by: Local
        Timeout: 0 Type: Static
    Group: 233.252.0.1
        Source: 10.0.0.4
        Last reported by: Local
        Timeout: 0 Type: Static
```

When you configure static groups on an interface on which you want to receive multicast traffic, and specify that a number of multicast sources be automatically accepted, you can also specify the number by which the address should be incremented for each source accepted. This is useful when you want to test forwarding to multiple receivers without having to configure each receiver separately and you do not want the source addresses to be sequential.

In this example, you create group 233.252.0.1 and accept addresses 10.0.0.2, 10.0.0.4, and 10.0.0.6 as the sources.

1. Configure the multicast source address increment by including the **source-increment** statement and specifying the number by which the address should be incremented for each source. The increment is specified in dotted decimal notation similar to an IPv4 address.

   ```
   [edit protocols igmp]
   user@host# set interface fe-0/1/2 static group 233.252.0.1 source 10.0.0.2 source-count
     3 source-increment 0.0.0.2
   ```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

   ```
   user@host> show configuration protocol igmp
   ```

   ```
   interface fe-0/1/2.0 {
     version 3;
     static {
       group 233.252.0.1 {
         source 10.0.0.2 {
           source-count 3;
           source-increment 0.0.0.2;
         }
       }
     }
   }
   ```

3. After you have committed the configuration and after the source is sending traffic, use the **show igmp group** command to verify that static group 233.252.0.1 has been created and that sources 10.0.0.2, 10.0.0.4, and 10.0.0.6 have been accepted.

   ```
   user@host> show igmp group
   Interface: fe-0/1/2
       Group: 233.252.0.1
           Source: 10.0.0.2
           Last reported by: Local
           Timeout: 0 Type: Static
       Group: 233.252.0.1
           Source: 10.0.0.4
   ```

```
            Last reported by: Local
            Timeout: 0 Type: Static
        Group: 233.252.0.1
            Source: 10.0.0.6
            Last reported by: Local
            Timeout: 0 Type: Static
```

When you configure static groups on an interface on which you want to receive multicast traffic and your network is operating in source-specific multicast (SSM) mode, you can specify that certain multicast source addresses be excluded.

By default the multicast source address configured in a static group operates in include mode. In include mode the multicast traffic for the group is accepted from the source address configured. You can also configure the static group to operate in exclude mode. In exclude mode the multicast traffic for the group is accepted from any address other than the source address configured.

If a source address is specified in a multicast group that is statically configured, the IGMP version on the interface must be set to IGMPv3. IGMPv2 is the default value.

In this example, you exclude address 10.0.0.2 as a source for group 233.252.0.1.

1. On the DR, configure a multicast static group to operate in exclude mode by including the **exclude** statement and specifying which IPv4 source address to exclude.

   ```
   [edit protocols igmp]
   user@host# set interface fe-0/1/2 static group 233.252.0.1 exclude source 10.0.0.2
   ```

2. After you commit the configuration, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

   ```
   user@host> show configuration protocol igmp
   ```

   ```
   interface fe-0/1/2.0 {
     version 3;
     static {
       group 233.252.0.1 {
         exclude;
         source 10.0.0.2;
       }
     }
   }
   ```

3. After you have committed the configuration and the source is sending traffic, use the **show igmp group detail** command to verify that static group 233.252.0.1 has been created and that the static group is operating in exclude mode.

   ```
   user@host> show igmp group detail
   Interface: fe-0/1/2
        Group: 233.252.0.1
            Group mode: Exclude
            Source: 10.0.0.2
            Last reported by: Local
            Timeout: 0 Type: Static
   ```

See Also

- Enabling MLD Static Group Membership on page 63
- group (Protocols IGMP) on page 1044
- group-count (Protocols IGMP) on page 1050
- group-increment (Protocols IGMP) on page 1052
- source-count (Protocols IGMP) on page 1294
- source-increment (Protocols IGMP) on page 1296
- static (Protocols IGMP) on page 1310

Recording IGMP Join and Leave Events

To determine whether IGMP tuning is needed in a network, you can configure the routing device to record IGMP join and leave events. You can record events globally for the routing device or for individual interfaces.

Table 4 on page 44 describes the recordable IGMP events.

Table 4: IGMP Event Messages

ERRMSG Tag	Definition
RPD_IGMP_JOIN	Records IGMP join events.
RPD_IGMP_LEAVE	Records IGMP leave events.
RPD_IGMP_ACCOUNTING_ON	Records when IGMP accounting is enabled on an IGMP interface.
RPD_IGMP_ACCOUNTING_OFF	Records when IGMP accounting is disabled on an IGMP interface.
RPD_IGMP_MEMBERSHIP_TIMEOUT	Records IGMP membership timeout events.

To enable IGMP accounting:

1. Enable accounting globally or on an IGMP interface. This example shows both options.

   ```
   [edit protocols igmp]
   user@host# set accounting
   user@host# set interface fe-0/1/0.2 accounting
   ```

2. Configure the events to be recorded and filter the events to a system log file with a descriptive filename, such as **igmp-events**.

   ```
   [edit system syslog file igmp-events]
   user@host# set any info
   user@host# set match ".*RPD_IGMP_JOIN.* | .*RPD_IGMP_LEAVE.* |
     .*RPD_IGMP_ACCOUNTING.* | .*RPD_IGMP_MEMBERSHIP_TIMEOUT.*"
   ```

3. Periodically archive the log file.

This example rotates the file size when it reaches 100 KB and keeps three files.

```
[edit system syslog file igmp-events]
user@host# set archive size 100000
user@host# set archive files 3
user@host# set archive archive-sites “ftp://user@host1//var/tmp” password
  “anonymous”
user@host# set archive archive-sites “ftp://user@host2//var/tmp” password “test”
user@host# set archive transfer-interval 24
user@host# set archive start-time 2011-01-07:12:30
```

4. You can monitor the system log file as entries are added to the file by running the **monitor start** and **monitor stop** commands.

```
user@host> monitor start igmp-events
```

```
*** igmp-events ***
Apr 16 13:08:23  host mgd[16416]: UI_CMDLINE_READ_LINE: User 'user', command
'run monitor start igmp-events '
monitor
```

See Also

- Understanding IGMP on page 25
- *Specifying Log File Size, Number, and Archiving Properties*

Limiting the Number of IGMP Multicast Group Joins on Logical Interfaces

The **group-limit** statement enables you to limit the number of IGMP multicast group joins for logical interfaces. When this statement is enabled on a router running IGMP version 2 (IGMPv2) or version 3 (IGMPv3), the limit is applied upon receipt of the group report. Once the group limit is reached, subsequent join requests are rejected.

When configuring limits for IGMP multicast groups, keep the following in mind:

- Each any-source group (*,G) counts as one group toward the limit.
- Each source-specific group (S,G) counts as one group toward the limit.
- Groups in IGMPv3 exclude mode are counted toward the limit.
- Multiple source-specific groups count individually toward the group limit, even if they are for the same group. For example, (S1, G1) and (S2, G1) would count as two groups toward the configured limit.
- Combinations of any-source groups and source-specific groups count individually toward the group limit, even if they are for the same group. For example, (*, G1) and (S, G1) would count as two groups toward the configured limit.
- Configuring and committing a group limit on a network that is lower than what already exists on the network results in the removal of all groups from the configuration. The

groups must then request to rejoin the network (up to the newly configured group limit).

- You can dynamically limit multicast groups on IGMP logical interfaces using dynamic profiles.

Starting in Junos OS Release 12.2, you can optionally configure a system log warning threshold for IGMP multicast group joins received on the logical interface. It is helpful to review the system log messages for troubleshooting purposes and to detect if an excessive amount of IGMP multicast group joins have been received on the interface. These log messages convey when the configured group limit has been exceeded, when the configured threshold has been exceeded, and when the number of groups drop below the configured threshold.

The **group-threshold** statement enables you to configure the threshold at which a warning message is logged. The range is 1 through 100 percent. The warning threshold is a percentage of the group limit, so you must configure the **group-limit** statement to configure a warning threshold. For instance, when the number of groups exceed the configured warning threshold, but remain below the configured group limit, multicast groups continue to be accepted, and the device logs the warning message. In addition, the device logs a warning message after the number of groups drop below the configured warning threshold. You can further specify the amount of time (in seconds) between the log messages by configuring the **log-interval** statement. The range is 6 through 32,767 seconds.

You might consider throttling log messages because every entry added after the configured threshold and every entry rejected after the configured limit causes a warning message to be logged. By configuring a log interval, you can throttle the amount of system log warning messages generated for IGMP multicast group joins.

NOTE: On ACX Series routers, the maximum number of multicast routes is 1024.

To limit multicast group joins on an IGMP logical interface:

1. Access the logical interface at the IGMP protocol hierarchy level.

   ```
   [edit]
   user@host# edit protocols igmp interface interface-name
   ```

2. Specify the group limit for the interface.

   ```
   [edit protocols igmp interface interface-name]
   user@host# set group-limit limit
   ```

3. (Optional) Configure the threshold at which a warning message is logged.

   ```
   [edit protocols igmp interface interface-name]
   user@host# set group-threshold value
   ```

4. (Optional) Configure the amount of time between log messages.

```
[edit protocols igmp interface interface-name]
user@host# set log-interval seconds
```

To confirm your configuration, use the **show protocols igmp** command. To verify the operation of IGMP on the interface, including the configured group limit and the optional warning threshold and interval between log messages, use the show igmp interface command.

See Also
- Enabling IGMP Static Group Membership on page 37

Tracing IGMP Protocol Traffic

Tracing operations record detailed messages about the operation of routing protocols, such as the various types of routing protocol packets sent and received, and routing policy actions. You can specify which trace operations are logged by including specific tracing flags. The following table describes the flags that you can include.

Flag	Description
all	Trace all operations.
client-notification	Trace notifications.
general	Trace general flow.
group	Trace group operations.
host-notification	Trace host notifications.
leave	Trace leave group messages (IGMPv2 only).
mtrace	Trace mtrace packets. Use the **mtrace** command to troubleshoot the software.
normal	Trace normal events.
packets	Trace all IGMP packets.
policy	Trace policy processing.
query	Trace IGMP membership query messages, including general and group-specific queries.
report	Trace membership report messages.
route	Trace routing information.
state	Trace state transitions.
task	Trace task processing.

Flag	Description
timer	Trace timer processing.

In the following example, tracing is enabled for all routing protocol packets. Then tracing is narrowed to focus only on IGMP packets of a particular type. To configure tracing operations for IGMP:

1. (Optional) Configure tracing at the routing options level to trace all protocol packets.

   ```
   [edit routing-options traceoptions]
   user@host# set file all-packets-trace
   user@host# set flag all
   ```

2. Configure the filename for the IGMP trace file.

   ```
   [edit protocols igmp traceoptions]
   user@host# set file igmp-trace
   ```

3. (Optional) Configure the maximum number of trace files.

   ```
   [edit protocols igmp traceoptions]
   user@host# set file files 5
   ```

4. (Optional) Configure the maximum size of each trace file.

   ```
   [edit protocols igmp traceoptions]
   user@host# set file size 1m
   ```

5. (Optional) Enable unrestricted file access.

   ```
   [edit protocols igmp traceoptions]
   user@host# set file world-readable
   ```

6. Configure tracing flags. Suppose you are troubleshooting issues with a particular multicast group. The following example shows how to flag all events for packets associated with the group IP address.

   ```
   [edit protocols igmp traceoptions]
   user@host# set flag group | match 233.252.0.2
   ```

7. View the trace file.

   ```
   user@host> file list /var/log
   user@host> file show /var/log/igmp-trace
   ```

See Also

- Understanding IGMP on page 25
- *Tracing and Logging Junos OS Operations*
- mtrace on page 1429

Disabling IGMP

To disable IGMP on an interface, include the **disable** statement:

```
disable;
```

You can include this statement at the following hierarchy levels:

- **[edit protocols igmp interface *interface-name*]**
- **[edit logical-systems *logical-system-name* protocols igmp interface *interface-name*]**

NOTE: ACX Series routers do not support **[edit logical-systems *logical-system-name* protocols]** hierarchy level.

See Also

- Understanding IGMP on page 25
- Configuring IGMP on page 27
- Enabling IGMP on page 28

IGMP and Nonstop Active Routing

Nonstop active routing (NSR) configurations include two Routing Engines that share information so that routing is not interrupted during Routing Engine failover. These NSR configurations include passive support with IGMP in connection with PIM. The master Routing Engine uses IGMP to determine its PIM multicast state, and this IGMP-derived information is replicated on the backup Routing Engine. IGMP on the new master Routing Engine (after failover) relearns the state information quickly through IGMP operation. In the interim, the new master Routing Engine retains the IGMP-derived PIM state as received by the replication process from the old master Routing Engine. This state information times out unless refreshed by IGMP on the new master Routing Engine. No additional IGMP configuration is required.

See Also

- Understanding Nonstop Active Routing for PIM on page 371
- Examples: Configuring MLD on page 50

Release History Table

Release	Description
15.2	Starting in Junos OS Release 15.2, PIMv1 is not supported.
12.2	Starting in Junos OS Release 12.2, you can optionally configure a system log warning threshold for IGMP multicast group joins received on the logical interface.

Related Documentation

- Examples: Configuring MLD on page 50

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From the CLI, enter the **show igmp interface** command.

Sample Output

```
user@host> show igmp interface
Interface: ge-0/0/0.0
    Querier: 192.168.4.36
    State:         Up Timeout:     197 Version:  2 Groups:      0

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

Meaning The output shows a list of the interfaces that are configured for IGMP. Verify the following information:

- Each interface on which IGMP is enabled is listed.
- Next to **Version**, the number 2 appears.

Examples: Configuring MLD

- Understanding MLD on page 51
- Configuring MLD on page 54
- Enabling MLD on page 54
- Modifying the MLD Version on page 56
- Modifying the MLD Host-Query Message Interval on page 56

Understanding MLD

The Multicast Listener Discovery (MLD) Protocol manages the membership of hosts and routers in multicast groups. IP version 6 (IPv6) multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. Each routing device maintains a list of host multicast addresses that have listeners for each subnetwork, as well as a timer for each address. However, the routing device does not need to know the address of each listener—just the address of each host. The routing device provides addresses to the multicast routing protocol it uses, which ensures that multicast packets are delivered to all subnetworks where there are interested listeners. In this way, MLD is used as the transport for the Protocol Independent Multicast (PIM) Protocol.

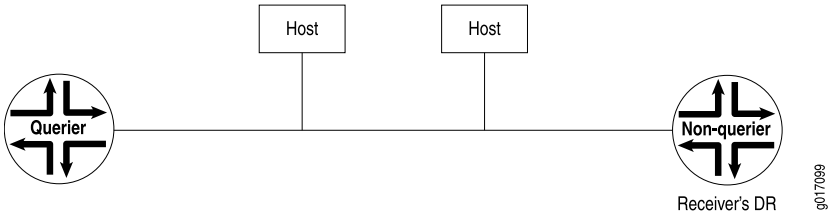
MLD is an integral part of IPv6 and must be enabled on all IPv6 routing devices and hosts that need to receive IP multicast traffic. The Junos OS supports MLD versions 1 and 2. Version 2 is supported for source-specific multicast (SSM) include and exclude modes.

In include mode, the receiver specifies the source or sources it is interested in receiving the multicast group traffic from. Exclude mode works the opposite of include mode. It allows the receiver to specify the source or sources it is not interested in receiving the multicast group traffic from.

For each attached network, a multicast routing device can be either a querier or a nonquerier. A querier routing device, usually one per subnet, solicits group membership information by transmitting MLD queries. When a host reports to the querier routing device that it has interested listeners, the querier routing device forwards the membership information to the rendezvous point (RP) routing device by means of the receiver's (host's) designated router (DR). This builds the rendezvous-point tree (RPT) connecting the host with interested listeners to the RP routing device. The RPT is the initial path used by the sender to transmit information to the interested listeners. Nonquerier routing devices do not transmit MLD queries on a subnet but can do so if the querier routing device fails.

All MLD-configured routing devices start as querier routing devices on each attached subnet (see Figure 3 on page 52). The querier routing device on the right is the receiver's DR.

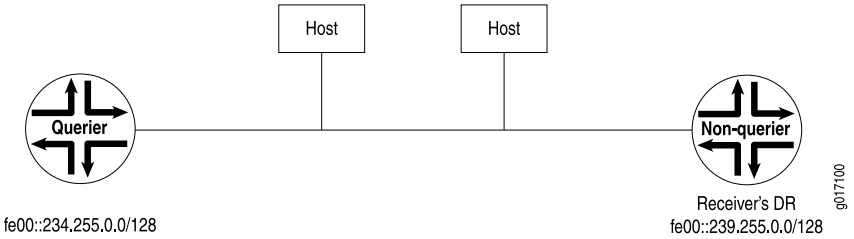
Figure 3: Routing Devices Start Up on a Subnet



To elect the querier routing device, the routing devices exchange query messages containing their IPv6 source addresses. If a routing device hears a query message whose IPv6 source address is numerically lower than its own selected address, it becomes a nonquerier. In Figure 4 on page 52, the routing device on the left has a source address numerically lower than the one on the right and therefore becomes the querier routing device.

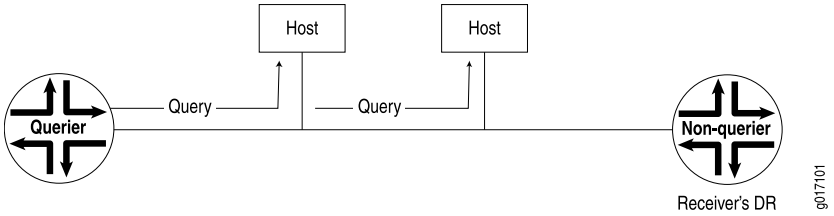
NOTE: In the practical application of MLD, several routing devices on a subnet are nonqueriers. If the elected querier routing device fails, query messages are exchanged among the remaining routing devices. The routing device with the lowest IPv6 source address becomes the new querier routing device. The IPv6 Neighbor Discovery Protocol (NDP) implementation drops incoming Neighbor Announcement (NA) messages that have a broadcast or multicast address in the target link-layer address option. This behavior is recommended by RFC 2461.

Figure 4: Querier Routing Device Is Determined



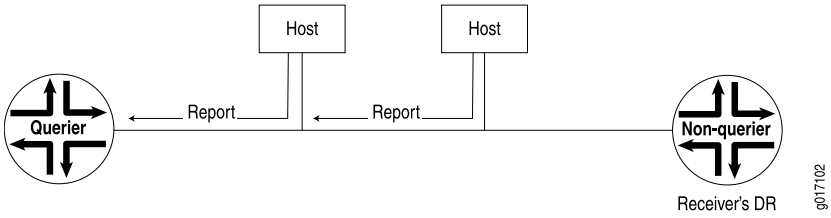
The querier routing device sends general MLD queries on the **link-scope all-nodes** multicast address FF02::1 at short intervals to all attached subnets to solicit group membership information (see Figure 5 on page 52). Within the query message is the *maximum response delay* value, specifying the maximum allowed delay for the host to respond with a report message.

Figure 5: General Query Message Is Issued



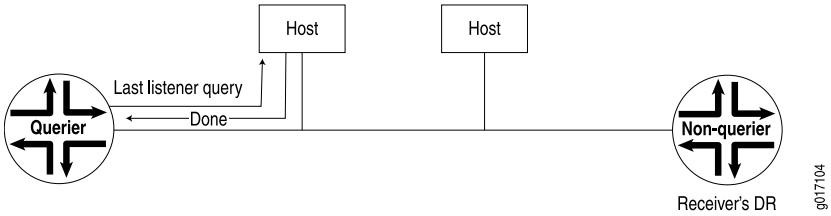
If interested listeners are attached to the host receiving the query, the host sends a report containing the host's IPv6 address to the routing device (see Figure 6 on page 53). If the reported address is not yet in the routing device's list of multicast addresses with interested listeners, the address is added to the list and a timer is set for the address. If the address is already on the list, the timer is reset. The host's address is transmitted to the RP in the PIM domain.

Figure 6: Reports Are Received by the Querier Routing Device



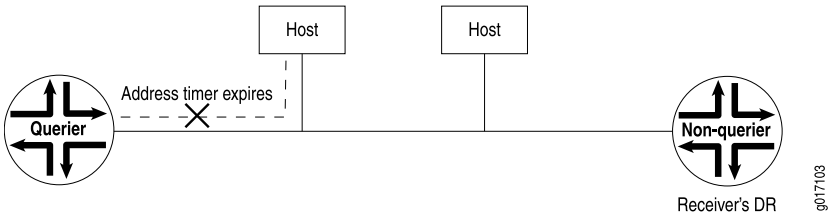
If the host has no interested multicast listeners, it sends a done message to the querier routing device. On receipt, the querier routing device issues a multicast address-specific query containing the last **listener query interval** value to the multicast address of the host. If the routing device does not receive a report from the multicast address, it removes the multicast address from the list and notifies the RP in the PIM domain of its removal (see Figure 7 on page 53).

Figure 7: Host Has No Interested Receivers and Sends a Done Message to Routing Device



If a done message is not received by the querier routing device, the querier routing device continues to send multicast address-specific queries. If the timer set for the address on receipt of the last report expires, the querier routing device assumes there are no longer interested listeners on that subnet, removes the multicast address from the list, and notifies the RP in the PIM domain of its removal (see Figure 8 on page 53).

Figure 8: Host Address Timer Expires and Address Is Removed from Multicast Address List



See Also

- Enabling MLD on page 54
- Example: Recording MLD Join and Leave Events on page 70
- Example: Modifying the MLD Robustness Variable on page 61

Configuring MLD

To configure the Multicast Listener Discovery (MLD) Protocol, include the **mld** statement:

```
mld {
  accounting;
  interface interface-name {
    disable;
    (accounting | no-accounting);
    group-policy [ policy-names ];
    immediate-leave;
    oif-map [ map-names ];
    passive;
    ssm-map ssm-map-name;
    static {
      group multicast-group-address {
        exclude;
        group-count number;
        group-increment increment;
        source ip-address {
          source-count number;
          source-increment increment;
        }
      }
    }
    version version;
  }
  maximum-transmit-rate packets-per-second;
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**

By default, MLD is enabled on all broadcast interfaces when you configure Protocol Independent Multicast (PIM) or the Distance Vector Multicast Routing Protocol (DVMRP).

Enabling MLD

The Multicast Listener Discovery (MLD) Protocol manages multicast groups by establishing, maintaining, and removing groups on a subnet. Multicast routing devices use MLD to learn which groups have members on each of their attached physical networks. MLD must be enabled for the router to receive IPv6 multicast packets. MLD is only needed for IPv6 networks, because multicast is handled differently in IPv4 networks. MLD is enabled on all IPv6 interfaces on which you configure PIM and on all IPv6 broadcast interfaces when you configure DVMRP.

MLD specifies different behaviors for multicast listeners and for routers. When a router is also a listener, the router responds to its own messages. If a router has more than one

interface to the same link, it needs to perform the router behavior over only one of those interfaces. Listeners, on the other hand, must perform the listener behavior on all interfaces connected to potential receivers of multicast traffic.

If MLD is not running on an interface—either because PIM and DVMRP are not configured on the interface or because MLD is explicitly disabled on the interface—you can explicitly enable MLD.

To explicitly enable MLD:

1. If PIM and DVMRP are not running on the interface, explicitly enable MLD by including the interface name.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/0/0.0
   ```

2. Check to see if MLD is disabled on any interfaces. In the following example, MLD is disabled on a Gigabit Ethernet interface.

   ```
   [edit protocols mld]
   user@host# show

   interface fe-0/0/0.0;
   interface ge-0/0/0.0 {
       disable;
   }
   ```

3. Enable MLD on the interface by deleting the **disable** statement.

   ```
   [edit protocols mld]
   delete interface ge-0/0/0.0 disable
   ```

4. Verify the configuration.

   ```
   [edit protocols mld]
   user@host# show

   interface fe-0/0/0.0;
   interface ge-0/0/0.0;
   ```

5. Verify the operation of MLD by checking the output of the **show mld interface** command.

See Also

- Understanding MLD on page 51
- Disabling MLD on page 74
- **show mld interface on page 1507** in the CLI Explorer
- RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*
- RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*
- RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

Modifying the MLD Version

By default, the router supports MLD version 1 (MLDv1). To enable the router to use MLD version 2 (MLDv2) for source-specific multicast (SSM) only, include the **version 2** statement.

If you configure the MLD version setting at the individual interface hierarchy level, it overrides configuring the IGMP version using the **interface all** statement.

If a source address is specified in a multicast group that is statically configured, the version must be set to MLDv2.

To change an MLD interface to version 2:

1. Configure the MLD interface.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/0/0.0 version 2
   ```

2. Verify the configuration by checking the **version** field in the output of the **show mld interface** command. The **show mld statistics** command has version-specific output fields, such as the counters in the **MLD Message type** field.

See Also

- Understanding MLD on page 51
- Source-Specific Multicast Groups Overview on page 315
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316
- Example: Configuring an SSM-Only Domain on page 319
- Example: Configuring PIM SSM on a Network on page 320
- Example: Configuring SSM Mapping on page 322
- RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*
- RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

Modifying the MLD Host-Query Message Interval

The objective of MLD is to keep routers up to date with IPv6 group membership of the entire subnet. Routers need not know who all the members are, only that members exist. Each host keeps track of which multicast groups are subscribed to. On each link, one router is elected the querier. The MLD querier router periodically sends general host-query messages on each attached network to solicit membership information. These messages solicit group membership information and are sent to the **link-scope all-nodes** address **FF02::1**. A general host-query message has a maximum response time that you can set by configuring the query response interval.

The query response timeout, the query interval, and the robustness variable are related in that they are all variables that are used to calculate the multicast listener interval. The

multicast listener interval is the number of seconds that must pass before a multicast router determines that no more members of a host group exist on a subnet. The multicast listener interval is calculated as the (robustness variable x query-interval) + (1 x query-response-interval). If no reports are received for a particular group before the multicast listener interval has expired, the routing device stops forwarding remotely-originated multicast packets for that group onto the attached network.

By default, host-query messages are sent every 125 seconds. You can change this interval to change the number of MLD messages sent on the subnet.

To modify the query interval:

1. Configure the interval.

   ```
   [edit protocols mld]
   user@host# set query-interval 200
   ```

   The value can be from 1 through 1024 seconds.

2. Verify the configuration by checking the **MLD Query Interval** field in the output of the **show mld interface** command.

3. Verify the operation of the query interval by checking the **Listener Query** field in the output of the **show mld statistics** command.

See Also

- Understanding MLD on page 51
- Modifying the MLD Query Response Interval on page 57
- Example: Modifying the MLD Robustness Variable on page 61
- **show mld interface on page 1507** in the CLI Explorer
- **show mld statistics on page 1511** in the CLI Explorer

Modifying the MLD Query Response Interval

The query response interval is the maximum amount of time that can elapse between when the querier router sends a host-query message and when it receives a response from a host. You can change this interval to adjust the burst peaks of MLD messages on the subnet. Set a larger interval to make the traffic less bursty.

The query response timeout, the query interval, and the robustness variable are related in that they are all variables that are used to calculate the multicast listener interval. The multicast listener interval is the number of seconds that must pass before a multicast router determines that no more members of a host group exist on a subnet. The multicast listener interval is calculated as the (robustness variable x query-interval) + (1 x query-response-interval). If no reports are received for a particular group before the multicast listener interval has expired, the routing device stops forwarding remotely-originated multicast packets for that group onto the attached network.

The default query response interval is 10 seconds. You can configure a subsecond interval up to one digit to the right of the decimal point. The configurable range is 0.1 through 0.9, then in 1-second intervals 1 through 999,999.

To modify the query response interval:

1. Configure the interval.

   ```
   [edit protocols mld]
   user@host# set query-response-interval 0.5
   ```

2. Verify the configuration by checking the **MLD Query Response Interval** field in the output of the **show mld interface** command.

3. Verify the operation of the query interval by checking the **Listener Query** field in the output of the **show mld statistics** command.

See Also

- Understanding MLD on page 51
- Modifying the MLD Host-Query Message Interval on page 56
- Example: Modifying the MLD Robustness Variable on page 61
- **show mld interface on page 1507** in the CLI Explorer
- **show mld statistics on page 1511** in the CLI Explorer

Modifying the MLD Last-Member Query Interval

The last-member query interval (also called the last-listener query interval) is the maximum amount of time between group-specific query messages, including those sent in response to done messages sent on the **link-scope-all-routers** address FF02::2. You can lower this interval to reduce the amount of time it takes a router to detect the loss of the last member of a group.

When the routing device that is serving as the querier receives a leave-group (done) message from a host, the routing device sends multiple group-specific queries to the group. The querier sends a specific number of these queries, and it sends them at a specific interval. The number of queries sent is called the last-listener query count. The interval at which the queries are sent is called the last-listener query interval. Both settings are configurable, thus allowing you to adjust the leave latency. The IGMP leave latency is the time between a request to leave a multicast group and the receipt of the last byte of data for the multicast group.

The last-listener query count x (times) the last-listener query interval = (equals) the amount of time it takes a routing device to determine that the last member of a group has left the group and to stop forwarding group traffic.

The default last-listener query interval is 1 second. You can configure a subsecond interval up to one digit to the right of the decimal point. The configurable range is 0.1 through 0.9, then in 1-second intervals 1 through 999,999.

To modify this interval:

1. Configure the time (in seconds) that the routing device waits for a report in response to a group-specific query.

   ```
   [edit protocols mld]
   user@host# set query-last-member-interval 0.1
   ```

2. Verify the configuration by checking the **MLD Last Member Query Interval** field in the output of the **show igmp interfaces** command.

NOTE: You can configure the last-member query count by configuring the robustness variable. The two are always equal.

See Also

- Understanding MLD on page 51
- Modifying the MLD Query Response Interval on page 57
- Example: Modifying the MLD Robustness Variable on page 61
- **show mld interface on page 1507** in the CLI Explorer

Specifying Immediate-Leave Host Removal for MLD

The immediate leave setting is useful for minimizing the leave latency of MLD memberships. When this setting is enabled, the routing device leaves the multicast group immediately after the last host leaves the multicast group.

The immediate-leave setting enables host tracking, meaning that the device keeps track of the hosts that send join messages. This allows MLD to determine when the last host sends a leave message for the multicast group.

When the immediate leave setting is enabled, the device removes an interface from the forwarding-table entry without first sending MLD group-specific queries to the interface. The interface is pruned from the multicast tree for the multicast group specified in the MLD leave message. The immediate leave setting ensures optimal bandwidth management for hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When immediate leave is disabled and one host sends a leave group message, the routing device first sends a group query to determine if another receiver responds. If no receiver responds, the routing device removes all hosts on the interface from the multicast group. Immediate leave is disabled by default for both MLD version 1 and MLD version 2.

NOTE: Although host tracking is enabled for IGMPv2 and MLDv1 when you enable immediate leave, use immediate leave with these versions only when there is one host on the interface. The reason is that IGMPv2 and MLDv1 use a report suppression mechanism whereby only one host on an interface sends a group join report in response to a membership query. The other interested hosts suppress their reports. The purpose of this mechanism is to avoid a flood of reports for the same group. But it also interferes with host tracking, because the router only knows about the one interested host and does not know about the others.

To enable immediate leave:

1. Configure immediate leave on the MLD interface.

   ```
   [edit protocols mld]
   user@host# set interface ge-0/0/0.1 immediate-leave
   ```

2. Verify the configuration by checking the **Immediate Leave** field in the output of the **show mld interface** command.

See Also

- Understanding MLD on page 51
- **show mld interface on page 1507** in the CLI Explorer

Filtering Unwanted MLD Reports at the MLD Interface Level

Suppose you need to limit the subnets that can join a certain multicast group. The **group-policy** statement enables you to filter unwanted MLD reports at the interface level.

When the **group-policy** statement is enabled on a router, after the router receives an MLD report, the router compares the group against the specified group policy and performs the action configured in that policy (for example, rejects the report if the policy matches the defined address or network).

You define the policy to match only MLD group addresses (for MLDv1) by using the policy's **route-filter** statement to match the group address. You define the policy to match MLD (source, group) addresses (for MLDv2) by using the policy's **route-filter** statement to match the group address and the policy's **source-address-filter** statement to match the source address.

To filter unwanted MLD reports:

1. Configure an MLDv1 policy.

   ```
   [edit policy-statement reject_policy_v1]
   user@host# set from route-filter fec0:1:1:4::/64 exact
   user@host# set then reject
   ```

2. Configure an MLDv2 policy.

```
[edit policy-statement reject_policy_v2]
user@host# set from route-filter fec0:1:1:4::/64 exact
user@host# set from source-address-filter fe80::2e0:81ff:fe05:1a8d/32 orlonger
user@host# set then reject
```

3. Apply the policies to the MLD interfaces where you prefer not to receive specific group or (source, group) reports. In this example, **ge-0/0/0.1** is running MLDv1 and **ge-0/1/1.0** is running MLDv2.

```
[edit protocols mld]
user@host# set interface ge-0/0/0.1 group-policy reject_policy_v1
user@host# set interface ge-0/1/1.0 group-policy reject_policy_v2
```

4. Verify the operation of the filter by checking the **Rejected Report** field in the output of the **show mld statistics** command.

See Also

- Understanding MLD on page 51
- *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*
- **show mld statistics on page 1511** in the CLI Explorer

Example: Modifying the MLD Robustness Variable

This example shows how to configure and verify the MLD robustness variable in a multicast domain.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Enable IPv6 unicast routing. See the *Junos OS Routing Protocols Library*.
- Enable PIM. See "PIM Overview" on page 185.

Overview

The MLD robustness variable can be fine-tuned to allow for expected packet loss on a subnet. Increasing the robust count allows for more packet loss but increases the leave latency of the subnetwork.

The value of the robustness variable is used in calculating the following MLD message intervals:

- Group member interval—Amount of time that must pass before a multicast router determines that there are no more members of a group on a network. This interval is calculated as follows: (robustness variable x query-interval) + (1 x query-response-interval).
- Other querier present interval—Amount of time that must pass before a multicast router determines that there is no longer another multicast router that is the querier. This interval is calculated as follows: (robustness variable x query-interval) + (0.5 x query-response-interval).
- Last-member query count—Number of group-specific queries sent before the router assumes there are no local members of a group. The default number is the value of the robustness variable.

By default, the robustness variable is set to 2. The number can be from 2 through 10. You might want to increase this value if you expect a subnet to lose packets.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols mld robust-count 5
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To change the value of the robustness variable:

1. Configure the robust count.

   ```
   [edit protocols mld]
   user@host# set robust-count 5
   ```

2. If you are done configuring the device, commit the configuration.

   ```
   [edit protocols mld]
   user@host# commit
   ```

Verification

To verify the configuration is working properly, check the **MLD Robustness Count** field in the output of the **show mld interfaces** command.

See Also

- Understanding MLD on page 51

- Modifying the MLD Query Response Interval on page 57
- Modifying the MLD Last-Member Query Interval on page 58
- **show mld interface on page 1507** in the CLI Explorer

Limiting the Maximum MLD Message Rate

You can change the limit for the maximum number of MLD packets transmitted in 1 second by the router.

Increasing the maximum number of MLD packets transmitted per second might be useful on a router with a large number of interfaces participating in MLD.

To change the limit for the maximum number of MLD packets the router can transmit in 1 second, include the **maximum-transmit-rate** statement and specify the maximum number of packets per second to be transmitted.

Enabling MLD Static Group Membership

- Create a MLD Static Group Member on page 63
- Automatically create static groups on page 64
- Automatically increment group addresses on page 65
- Specify multicast source address (in SSM mode) on page 66
- Automatically specify multicast sources on page 67
- Automatically increment source addresses on page 68
- Exclude multicast source addresses (in SSM mode) on page 69

Create a MLD Static Group Member

You can create MLD static group membership to test multicast forwarding without a receiver host. When you enable MLD static group membership, data is forwarded to an interface without that interface receiving membership reports from downstream hosts.

Class-of-service (CoS) adjustment is not supported with MLD static group membership.

When you configure static groups on an interface on which you want to receive multicast traffic, you can specify the number of static groups to be automatically created.

In this example, you create static group ff0e::1:ff05:1a8d.

1. Configure the static groups to be created by including the **static** statement and **group** statement and specifying which IPv6 multicast address of the group to be created.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d
   ```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

   ```
   user@host> show configuration protocol mld
   ```

```
interface fe-0/1/2.0 {
  static {
    group ff0e::1:ff05:1a8d;
  }
}
```

3. After you have committed the configuration and after the source is sending traffic, use the **show mld group** command to verify that static group ff0e::1:ff05:1a8d has been created.

```
user@host> show mld group
Interface: fe-0/1/2
Group: ff0e::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a8d
Last reported by: Local
Timeout: 0 Type: Static
```

NOTE: You must specify a unique address for each group.

Automatically create static groups

When you create MLD static group membership to test multicast forwarding on an interface on which you want to receive multicast traffic, you can specify that a number of static groups be automatically created. This is useful when you want to test forwarding to multiple receivers without having to configure each receiver separately.

In this example, you create three groups.

1. Configure the number of static groups to be created by including the **group-count** statement and specifying the number of groups to be created.

```
[edit protocols mld]
user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d group-count 3
```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

```
user@host> show configuration protocol mld

interface fe-0/1/2.0 {
  static {
    group ff0e::1:ff05:1a8d {
      group-count 3;
    }
  }
}
```

3. After you have committed the configuration and the source is sending traffic, use the **show mld group** command to verify that static groups ff0e::1:ff05:1a8d, ff0e::1:ff05:1a8e, and ff0e::1:ff05:1a8f have been created.

```
user@host> show mld group

Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8e
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8f
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
```

Automatically increment group addresses

When you configure static groups on an interface on which you want to receive multicast traffic and you specify the number of static groups to be automatically created, you can also configure the group address to be automatically incremented by some number of addresses.

In this example, you create three groups and increase the group address by an increment of two for each group.

1. Configure the group address increment by including the **group-increment** statement and specifying the number by which the address should be incremented for each group. The increment is specified in a format similar to an IPv6 address.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d group-count 3
     group-increment ::2
   ```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

   ```
   user@host> show configuration protocol mld

   interface fe-0/1/2.0 {
     static {
       group ff0e::1:ff05:1a8d {
         group-increment ::2;
         group-count 3;
       }
     }
   }
   ```

3. After you have committed the configuration and the source is sending traffic, use the **show mld group** command to verify that static groups ff0e::1:ff05:1a8d, ff0e::1:ff05:1a8f, and ff0e::1:ff05:1a91 have been created.

   ```
   user@host> show mld group
   ```

```
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8f
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a91
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
```

Specify multicast source address (in SSM mode)

When you configure static groups on an interface on which you want to receive multicast traffic and your network is operating in source-specific multicast (SSM) mode, you can specify the multicast source address to be accepted.

If you specify a group address in the SSM range, you must also specify a source.

If a source address is specified in a multicast group that is statically configured, the MLD version must be set to MLDv2 on the interface. MLDv1 is the default value.

In this example, you create group ff0e::1:ff05:1a8d and accept IPv6 address fe80::2e0:81ff:fe05:1a8d as the only source.

1. Configure the source address by including the **source** statement and specifying the IPv6 address of the source host.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d source
     fe80::2e0:81ff:fe05:1a8d
   ```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

   ```
   user@host> show configuration protocol mld

   interface fe-0/1/2.0 {
     static {
       group ff0e::1:ff05:1a8d {
         source fe80::2e0:81ff:fe05:1a8d;
       }
     }
   }
   ```

3. After you have committed the configuration and the source is sending traffic, use the **show mld group** command to verify that static group ff0e::1:ff05:1a8d has been created and that source fe80::2e0:81ff:fe05:1a8d has been accepted.

```
user@host> show mld group

Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
```

Automatically specify multicast sources

When you configure static groups on an interface on which you want to receive multicast traffic, you can specify a number of multicast sources to be automatically accepted.

In this example, you create static group ff0e::1:ff05:1a8d and accept fe80::2e0:81ff:fe05:1a8d, fe80::2e0:81ff:fe05:1a8e, and fe80::2e0:81ff:fe05:1a8f as the source addresses.

1. Configure the number of multicast source addresses to be accepted by including the **source-count** statement and specifying the number of sources to be accepted.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d source
     fe80::2e0:81ff:fe05:1a8d source-count 3
   ```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

   ```
   user@host> show configuration protocol mld

   interface fe-0/1/2.0 {
     static {
       group ff0e::1:ff05:1a8d {
         source fe80::2e0:81ff:fe05:1a8d {
           source-count 3;
         }
       }
     }
   }
   ```

3. After you have committed the configuration and the source is sending traffic, use the **show mld group** command to verify that static group ff0e::1:ff05:1a8d has been created and that sources fe80::2e0:81ff:fe05:1a8d, fe80::2e0:81ff:fe05:1a8e, and fe80::2e0:81ff:fe05:1a8f have been accepted.

   ```
   user@host> show mld group

   Interface: fe-0/1/2
        Group: ff0e::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   Interface: fe-0/1/2
        Group: ff0e::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8e
            Last reported by: Local
            Timeout: 0 Type: Static
   ```

```
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8f
         Last reported by: Local
         Timeout: 0 Type: Static
```

Automatically increment source addresses

When you configure static groups on an interface on which you want to receive multicast traffic, and specify a number of multicast sources to be automatically accepted, you can also specify the number by which the address should be incremented for each source accepted.

In this example, you create static group ff0e::1:ff05:1a8d and accept fe80::2e0:81ff:fe05:1a8d, fe80::2e0:81ff:fe05:1a8f, and fe80::2e0:81ff:fe05:1a91 as the sources.

1. Configure the number of multicast source addresses to be accepted by including the **source-increment** statement and specifying the number of sources to be accepted.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d source
     fe80::2e0:81ff:fe05:1a8d source-count 3 source-increment ::2
   ```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

   ```
   user@host> show configuration protocol mld

   interface fe-0/1/2.0 {
     static {
       group ff0e::1:ff05:1a8d {
         source fe80::2e0:81ff:fe05:1a8d {
           source-count 3;
           source-increment ::2;
         }
       }
     }
   }
   ```

3. After you have committed the configuration and the source is sending traffic, use the **show mld group** command to verify that static group ff0e::1:ff05:1a8d has been created and that sources fe80::2e0:81ff:fe05:1a8d, fe80::2e0:81ff:fe05:1a8f, and fe80::2e0:81ff:fe05:1a91 have been accepted.

   ```
   user@host> show mld group

   Interface: fe-0/1/2
        Group: ff0e::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   Interface: fe-0/1/2
        Group: ff0e::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8f
   ```

```
        Last reported by: Local
        Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff0e2::1:ff05:1a8d
        Source: fe80::2e0:81ff:fe05:1a91
        Last reported by: Local
        Timeout: 0 Type: Static

Interface: fe-0/1/2
Group: ff0e::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a8d
Last reported by: Local
Timeout: 0 Type: Static
Group: ff0e::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a8f
Last reported by: Local
Timeout: 0 Type: Static
Group: ff0e::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a91
Last reported by: Local
Timeout: 0 Type: Static
```

Exclude multicast source addresses (in SSM mode)

When you configure static groups on an interface on which you want to receive multicast traffic and your network is operating in source-specific multicast (SSM) mode, you can specify that certain multicast source addresses be excluded.

By default the multicast source address configured in a static group operates in include mode. In include mode the multicast traffic for the group is accepted from the configured source address. You can also configure the static group to operate in exclude mode. In exclude mode the multicast traffic for the group is accepted from any address other than the configured source address.

If a source address is specified in a multicast group that is statically configured, the MLD version must be set to MLDv2 on the interface. MLDv1 is the default value.

In this example, you exclude address fe80::2e0:81ff:fe05:1a8d as a source for group ff0e::1:ff05:1a8d.

1. Configure a multicast static group to operate in exclude mode by including the **exclude** statement and specifying which IPv6 source address to be excluded.

   ```
   [edit protocols mld]
   user@host# set interface fe-0/1/2 static group ff0e::1:ff05:1a8d exclude source
     fe80::2e0:81ff:fe05:1a8d
   ```

2. After you commit the configuration, use the **show configuration protocol mld** command to verify the MLD protocol configuration.

   ```
   user@host> show configuration protocol mld

   interface fe-0/1/2.0 {
     static {
   ```

```
        group ff0e::1:ff05:1a8d {
          exclude;
          source fe80::2e0:81ff:fe05:1a8d;
        }
      }
    }
```

3. After you have committed the configuration and the source is sending traffic, use the **show mld group detail** command to verify that static group ff0e::1:ff05:1a8d has been created and that the static group is operating in exclude mode.

```
user@host> show mld group detail
Interface: fe-0/1/2
     Group: ff0e::1:ff05:1a8d
         Group mode: Exclude
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
```

Similar configuration is available for IPv4 multicast traffic using the IGMP protocol.

See Also
- Enabling IGMP Static Group Membership on page 37

Example: Recording MLD Join and Leave Events

This example shows how to determine whether MLD tuning is needed in a network by configuring the routing device to record MLD join and leave events.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Enable IPv6 unicast routing. See the *Junos OS Routing Protocols Library*.
- Enable PIM. See "PIM Overview" on page 185.

Overview

Table 5 on page 71 describes the recordable MLD join and leave events.

Table 5: MLD Event Messages

ERRMSG Tag	Definition
RPD_MLD_JOIN	Records MLD join events.
RPD_MLD_LEAVE	Records MLD leave events.
RPD_MLD_ACCOUNTING_ON	Records when MLD accounting is enabled on an MLD interface.
RPD_MLD_ACCOUNTING_OFF	Records when MLD accounting is disabled on an MLD interface.
RPD_MLD_MEMBERSHIP_TIMEOUT	Records MLD membership timeout events.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols mld interface fe-0/1/0.2 accounting
set system syslog file mld-events any info
set system syslog file mld-events match ".*RPD_MLD_JOIN.* | .*RPD_MLD_LEAVE.* |
  .*RPD_MLD_ACCOUNTING.* | .*RPD_MLD_MEMBERSHIP_TIMEOUT.*"
set system syslog file mld-events archive size 100000
set system syslog file mld-events archive files 3
set system syslog file mld-events archive transfer-interval 1440
set system syslog file mld-events archive archive-sites "ftp://user@host1//var/tmp"
  password "anonymous"
set system syslog file mld-events archive archive-sites "ftp://user@host2//var/tmp"
  password "test"
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure recording of MLD join and leave events:

1. Enable accounting globally or on an MLD interface. This example shows the interface configuration.

   ```
   [edit protocols mld]
   user@host# set interface fe–0/1/0.2 accounting
   ```

2. Configure the events to be recorded, and filter the events to a system log file with a descriptive filename, such as **mld-events**.

   ```
   [edit system syslog file mld-events]
   user@host# set any info
   [edit system syslog file mld-events]
   ```

```
user@host# set match ".*RPD_MLD_JOIN.* | .*RPD_MLD_LEAVE.* |
  .*RPD_MLD_ACCOUNTING.* | .*RPD_MLD_MEMBERSHIP_TIMEOUT.*"
```

3. Periodically archive the log file.

   This example rotates the file every 24 hours (1440 minutes) when it reaches 100 KB and keeps three files.

```
[edit system syslog file mld-events]
user@host# set archive size 100000
[edit system syslog file mld-events]
user@host# set archive files 3
[edit system syslog file mld-events]
user@host# set archive archive-sites "ftp://user@host1//var/tmp" password
  "anonymous"
[edit system syslog file mld-events]
user@host# set archive archive-sites "ftp://user@host2//var/tmp" password "test"
[edit system syslog file mld-events]
user@host# set archive transfer-interval 1440
[edit system syslog file mld-events]
user@host# set archive start-time 2011–01–07:12:30
```

4. If you are done configuring the device, commit the configuration.

```
[edit system syslog file mld-events]]
user@host# commit
```

Verification

You can view the system log file by running the **file show** command.

```
user@host> file show mld-events
```

You can monitor the system log file as entries are added to the file by running the **monitor start** and **monitor stop** commands.

```
user@host> monitor start mld-events
```

```
*** mld-events ***
Apr 16 13:08:23  host mgd[16416]: UI_CMDLINE_READ_LINE: User 'user', command 'run
 monitor start mld-events '
monitor
```

See Also

- Understanding MLD on page 51

Configuring the Number of MLD Multicast Group Joins on Logical Interfaces

The **group-limit** statement enables you to limit the number of MLD multicast group joins for logical interfaces. When this statement is enabled on a router running MLD version 2, the limit is applied upon receipt of the group report. Once the group limit is reached, subsequent join requests are rejected.

When configuring limits for MLD multicast groups, keep the following in mind:

- Each any-source group (*,G) counts as one group toward the limit.
- Each source-specific group (S,G) counts as one group toward the limit.
- Groups in MLDv2 exclude mode are counted toward the limit.
- Multiple source-specific groups count individually toward the group limit, even if they are for the same group. For example, (S1, G1) and (S2, G1) would count as two groups toward the configured limit.
- Combinations of any-source groups and source-specific groups count individually toward the group limit, even if they are for the same group. For example, (*, G1) and (S, G1) would count as two groups toward the configured limit.
- Configuring and committing a group limit on a network that is lower than what already exists on the network results in the removal of all groups from the configuration. The groups must then request to rejoin the network (up to the newly configured group limit).
- You can dynamically limit multicast groups on MLD logical interfaces by using dynamic profiles. For detailed information about creating dynamic profiles, see the *Junos OS Broadband Subscriber Management and Services Library*.

Beginning with Junos OS 12.2, you can optionally configure a system log warning threshold for MLD multicast group joins received on the logical interface. It is helpful to review the system log messages for troubleshooting purposes and to detect if an excessive amount of MLD multicast group joins have been received on the interface. These log messages convey when the configured group limit has been exceeded, when the configured threshold has been exceeded, and when the number of groups drop below the configured threshold.

The **group-threshold** statement enables you to configure the threshold at which a warning message is logged. The range is 1 through 100 percent. The warning threshold is a percentage of the group limit, so you must configure the **group-limit** statement to configure a warning threshold. For instance, when the number of groups exceed the configured warning threshold, but remain below the configured group limit, multicast groups continue to be accepted, and the device logs a warning message. In addition, the device logs a warning message after the number of groups drop below the configured warning threshold. You can further specify the amount of time (in seconds) between the log messages by configuring the **log-interval** statement. The range is 6 through 32,767 seconds.

You might consider throttling log messages because every entry added after the configured threshold and every entry rejected after the configured limit causes a warning message to be logged. By configuring a log interval, you can throttle the amount of system log warning messages generated for MLD multicast group joins.

To limit multicast group joins on an MLD logical interface:

1. Access the logical interface at the MLD protocol hierarchy level.

   ```
   [edit]
   user@host# edit protocols mld interface interface-name
   ```

2. Specify the group limit for the interface.

   [edit protocols mld interface *interface-name*]
   user@host# **set** group-limit ***limit***

3. (Optional) Configure the threshold at which a warning message is logged.

   [edit protocols mld interface *interface-name*]
   user@host# **set** group-threshold ***value***

4. (Optional) Configure the amount of time between log messages.

   [edit protocols mld interface *interface-name*]
   user@host# **set** log-interval ***seconds***

To confirm your configuration, use the **show protocols mld** command. To verify the operation of MLD on the interface, including the configured group limit and the optional warning threshold and interval between log messages, use the show mld interface command.

See Also
- Enabling MLD Static Group Membership on page 63

Disabling MLD

To disable MLD on an interface, include the **disable** statement:

```
interface interface-name {
  disable;
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols** mld**]**
- **[edit logical-systems** *logical-system-name* **protocols** mld**]**

See Also
- Enabling MLD on page 54

Release History Table

Release	Description
12.2	Beginning with Junos OS 12.2, you can optionally configure a system log warning threshold for MLD multicast group joins received on the logical interface.

Related Documentation
- Configuring IGMP on page 23

Understanding Distributed IGMP

By default, Internet Group Management Protocol (IGMP) processing takes place on the Routing Engine for MX Series routers. This centralized architecture may lead to reduced performance in scaled environments or when the Routing Engine undergoes CLI changes or route updates. You can improve system performance for IGMP processing by enabling *distributed IGMP*, which utilizes the Packet Forwarding Engine to maintain a higher system-wide processing rate for join and leave events.

Distributed IGMP Overview

Distributed IGMP works by moving IGMP processing from the Routing Engine to the Packet Forwarding Engine. When distributed IGMP is not enabled, IGMP processing is centralized on the routing protocol process (rpd) running on the Routing Engine. When you enable distributed IGMP, join and leave events are processed across Modular Port Concentrators (MPCs) on the Packet Forwarding Engine. Because join and leave processing is distributed across multiple MPCs instead of being processed through a centralized rpd on the Routing Engine, performance improves and join and leave latency decreases.

When you enable distributed IGMP, each Packet Forwarding Engine processes reports and generates queries, maintains local group membership to the interface mapping table and updates the forwarding state based on this table, runs distributed IGMP independently, and implements the **group-policy** and **ssm-map-policy** IGMP interface options.

NOTE: **Information from group-policy and ssm-map-policy IGMP interface options passes from the Routing Engine to the Packet Forwarding Engine.**

When you enable distributed IGMP, the **rpd** on the Routing Engine synchronizes all IGMP configurations (including global and interface-level configurations) from the **rpd** to each Packet Forwarding Engine, runs passive IGMP on distributed interfaces, and notifies Protocol Independent Multicast (PIM) of all group memberships per distributed IGMP interface.

Guidelines for Configuring Distributed IGMP

Consider the following guidelines when you configure distributed IGMP on an MX Series router with MPCs:

- Distributed IGMP increases network performance by reducing the maximum join and leave latency and by increasing join and leave events.

  NOTE: Join and leave latency may increase if multicast traffic is not preprovisioned and destined for an MX Series router when a join or leave event is received from a client interface.

- Distributed IGMP is supported for Ethernet interfaces. It does not improve performance on PIM interfaces.
- Starting in Junos OS release 18.2, distributed IGMP is supported on aggregated Ethernet interfaces, and for enhanced subscriber management. As such, IGMP processing for subscriber flows is moved from the Routing Engine to the Packet Forwarding Engine of supported line cards. Multicast groups can be comprised of mixed receivers, that is, some centralized IGMP and some distributed IGMP.
- You can reduce initial join delays by enabling Protocol Independent Multicast (PIM) static joins or IGMP static joins. You can reduce initial delays even more by preprovisioning multicast traffic. When you preprovision multicast traffic, MPCs with distributed IGMP interfaces receive multicast traffic.
- For distributed IGMP to function properly, you must enable enhanced IP network services on a single-chassis MX Series router. Virtual Chassis is not supported.
- When you enable distributed IGMP, the following interface options are not supported on the Packet Forwarding Engine: **oif-map**, **group-limit**, **ssm-map**, and **static**. The **traceoptions** and **accounting** statements can only be enabled for IGMP operations still performed on the Routing Engine; they are not supported on the Packet Forwarding Engine. The **clear igmp membership** command is not supported when distributed IGMP is enabled.

Release History Table

Release	Description
18.2	Starting in Junos OS release 18.2, distributed IGMP is supported on aggregated Ethernet interfaces, and for enhanced subscriber management. As such, IGMP processing for subscriber flows is moved from the Routing Engine to the Packet Forwarding Engine of supported line cards. Multicast groups can be comprised of mixed receivers, that is, some centralized IGMP and some distributed IGMP.

Related Documentation

- Understanding IGMP on page 25
- For general information about IGMP, see the *Multicast Protocols Feature Guide*

Enabling Distributed IGMP

Configuring distributed IGMP improves performance by reducing join and leave latency. This works by moving IGMP processing from the Routing Engine to the Packet Forwarding

Engine. In contrast to centralized IGMP processing on the Routing Engine, the Packet Forwarding Engine disperses traffic across multiple Modular Port Concentrators (MPCs).

You can enable distributed IGMP on static interfaces or dynamic interfaces. As a prerequisite, you must enable enhanced IP network services on a single-chassis MX Series router.

- Enabling Distributed IGMP on Static Interfaces on page 77
- Enabling Distributed IGMP on Dynamic Interfaces on page 77
- Configuring Multicast Traffic for Distributed IGMP on page 78

Enabling Distributed IGMP on Static Interfaces

You can enable distributed IGMP on a static interface by configuring enhanced IP network services and including the **distributed** statement at the **[edit protocols igmp interface *interface-name*]** hierarchy level. Enhanced IP network services must be enabled (at the **[chassis network-services enhanced-ip]** hierarchy).

To enable distributed IGMP on a static interface:

1. Configure the IGMP static interface.

   ```
   [edit protocols igmp ]
   user@host# set interface interface-name
   ```

2. Enable distributed IGMP on a static interface.

   ```
   [edit protocols igmp interface interface-name]
   user@host# set distributed
   ```

3. Commit the configuration.

Enabling Distributed IGMP on Dynamic Interfaces

You can enable distributed IGMP on a dynamic interface by configuring enhanced IP network services and including the **distributed** statement at the **[edit dynamic profiles *profile-name* protocols]** hierarchy level. Enhanced IP network services must be enabled (at the **[chassis network-services enhanced-ip]** hierarchy).

1. Configure the IGMP interface.

   ```
   [edit dynamic profiles profile-name protocols]
   user@host# set interface $junos-interface-name
   ```

2. Enable distributed IGMP on a dynamic interface.

   ```
   [edit dynamic profiles profile-name protocols interface $junos-interface-name]
   user@host# set distributed
   ```

3. Commit the configuration.

Configuring Multicast Traffic for Distributed IGMP

Configuring static source and group (S,G) addresses for distributed IGMP reduces join delays and sends multicast traffic to the last-hop router. You can configure static multicast groups (S,G) for distributed IGMP at the **[edit protocols pim]** hierarchy level. You can issue the **distributed** keyword at one of the following three hierarchy levels:

- **[edit protocols pim static]**

  Issuing the **distributed** keyword at this hierarchy level enables static joins for specific multicast (S,G) groups and preprovisions all of them so that all distributed IGMP Packet Forwarding Engines receive traffic.

- **[edit protocols pim static group *multicast-group-address*]**

  Issuing the **distributed** keyword at this hierarchy level enables static joins for multicast (S,G) groups so that all distributed IGMP Packet Forwarding Engines receive traffic and preprovisions a specific multicast group address (G).

- **[edit protocols pim static group *multicast-group-address* source *source-address*]**

  Issuing the **distributed** keyword at this hierarchy level enables static joins for multicast (S,G) groups so that all Packet Forwarding Engines receive traffic, but preprovisions a specific multicast (S,G) group.

To configure static multicast (S,G) addresses for distributed IGMP:

1. Configure static PIM.

   ```
   [edit protocols pim]
   user@host# set static
   ```

2. (Optional) Enable static joins for specific (S,G) addresses and preprovision all of them so that all distributed IGMP Packet Forwarding Engines receive traffic. In the example, multicast traffic for all of the groups (225.0.0.1, 10.10.10.1), (225.0.0.1, 10.10.10.2), and (225.0.0.2, *) is preprovisioned.

   ```
   [edit protocols pim]
   user@host# set protocols pim static distributed
   user@host# set protocols pim static group 225.0.0.1 source 10.10.10.1
   user@host# set protocols pim static group225.0.0.1 source10.10.10.2
   user@host# set protocols pim static group 225.0.0.2
   ```

3. (Optional) Enable static joins for specific multicast (S,G) groups so that all distributed IGMP Packet Forwarding Engines receive traffic and preprovision a specific multicast group address (G). In the example, multicast traffic for groups (225.0.0.1, 10.10.10.1) and (225.0.0.1, 10.10.10.2) is preprovisioned, but group (225.0.0.2, *) is not preprovisioned.

   ```
   [edit protocols pim]
   user@host# set protocols pim static
   user@host# set protocols pim static group 225.0.0.1  distributed
   user@host# set protocols pim static group 225.0.0.1 source10.10.10.1
   user@host# set protocols pim static group 225.0.0.1 source10.10.10.2
   ```

```
user@host# set protocols pim static group 225.0.0.2
```

4. (Optional) Enable a static join for specific multicast (S,G) groups so that all Packet Forwarding Engines receive traffic, but preprovision only one specific multicast address group. In the example, multicast traffic for group (225.0.0.1, 10.10.10.1) is preprovisioned, but all other groups are not preprovisioned.

```
[edit protocols pim]
user@host# set protocols pim static
user@host# set protocols pim static group 225.0.0.1
user@host# set protocols pim static group 225.0.0.1 source 10.10.10.1 distributed
user@host# set protocols pim static group225.0.0.1 source10.10.10.2
user@host# set protocols pim static group 225.0.0.2
```

5. Commit the configuration.

See Also

- *Configuring Dynamic DHCP Client Access to a Multicast Network*
- For information about enabling IGMP, see "Enabling IGMP" in the *Multicast Protocols Feature Guide*
- For general information about configuring IGMP, see the *Multicast Protocols Feature Guide*

CHAPTER 3

Configuring IGMP Snooping

IGMP Snooping Overview

With IGMP snooping enabled, the Juniper Networks device monitors the IGMP (Internet Group Management Protocol) traffic between hosts and multicast routers and uses what it learns to forward multicast traffic to only the downstream interfaces that are connected to interested receivers. This conserves bandwidth by allowing the device to send multicast traffic to only those interfaces that are connected to devices that want to receive the traffic (instead of flooding the traffic to all the downstream VLAN interfaces).

This IGMP snooping topic includes:

Benefits of IGMP Snooping

- **Optimized bandwidth utilization**—The main benefit of IGMP snooping is to reduce flooding of packets. IPv4 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN.
- **Improved security**—Denial of service attacks from unknown sources are prevented.

How IGMP Snooping Works

The device usually learns unicast MAC addresses by checking the source address field of the frames it receives and then sends any traffic for that unicast address only to the appropriate interface. However, a multicast MAC address can never be the source address for a packet. As a result, when the device receives traffic for a multicast destination address, it floods the traffic on the relevant VLAN, which can cause a significant amount of traffic to be sent unnecessarily.

IGMP snooping prevents this flooding. When you enable IGMP snooping, the device monitors IGMP packets between receivers and multicast routers and uses the content of the packets to build a multicast cache table—a database of multicast groups and the interfaces that are connected to members of the groups. When the device receives multicast packets, it uses the cache table to selectively forward the traffic to only the interfaces that are connected to members of the appropriate multicast groups.

NOTE: IGMP snooping is enabled by default on the default VLAN only. With versions of Junos OS for the QFX Series previous to 13.2, IGMP snooping is enabled by default on all VLANs.

NOTE: You cannot configure IGMP snooping on a secondary (private) VLAN.

How IGMP Snooping Works with Routed VLAN Interfaces

The device can use a routed VLAN interface (RVI) to forward traffic between VLANs that connect to it. IGMP snooping works with Layer 2 interfaces and RVIs to forward multicast traffic in a switched network.

When the device receives a multicast packet, its Packet Forwarding Engines perform a multicast lookup on the packet to determine how to forward the packet to its local interfaces. From the results of the lookup, each Packet Forwarding Engine extracts a list of Layer 3 interfaces that have ports local to the Packet Forwarding Engine. If the list includes an RVI, the device provides a bridge multicast group ID for the RVI to the Packet Forwarding Engine.

For VLANs that include multicast receivers, the bridge multicast ID includes a sub-next-hop ID, which identifies the Layer 2 interfaces in the VLAN that are interested in receiving the multicast stream. The Packet Forwarding Engine then forwards multicast traffic to bridge multicast IDs that have multicast receivers for a given multicast group.

IGMP Message Types

Multicast routers use IGMP to learn which groups have interested listeners, for each of their attached physical networks. In any given subnet, one multicast router acts as an IGMP querier. The IGMP querier sends out the following types of queries to hosts:

- General query—Asks whether any host is listening to any group.
- Group-specific query—(IGMPv2 and IGMPv3 only) Asks whether any host is listening to a specific multicast group. This query is sent in response to a host leaving the multicast group and allows the router to quickly determine if any remaining hosts are interested in the group.
- Group-and-source-specific query—(IGMPv3 only) Asks whether any host is listening to group multicast traffic from a specific multicast source. This query is sent in response to a host indicating that it is not longer interested in receiving group multicast traffic from the multicast source and allows the router to quickly determine any remaining hosts are interested in receiving group multicast traffic from that source.

Hosts that are multicast listeners send the following kinds of messages:

- Membership report—Indicates that the host wants to join a particular multicast group.
- Leave report—(IGMPv2 and IGMPv3 only) Indicates that the host wants to leave a particular multicast group.

How Hosts Join and Leave Multicast Groups

Hosts can join multicast groups in two ways:

- By sending an unsolicited IGMP join message to a multicast router that specifies the IP multicast group that the host is attempting to join.
- By sending an IGMP join message in response to a general query from a multicast router.

A multicast router continues to forward multicast traffic to a VLAN provided that at least one host on that VLAN responds to the periodic general IGMP queries. For a host to remain a member of a multicast group, therefore, it must continue to respond to the periodic general IGMP queries.

To leave a multicast group, either a host cannot respond to the periodic general IGMP queries, which results in a "silent leave" (the only leave option for IGMPv1), or a host can send a group-specific IGMPv2 leave message.

Support for IGMPv3 Multicast Sources

In IGMPv3, a host can send a membership report that includes a list of source addresses. When the host sends a membership report in INCLUDE mode, the host is interested in group multicast traffic only from those sources in the source address list. If host sends a membership report in EXCLUDE mode, the host is interested in group multicast traffic from any source *except* the sources in the source address list. A host can also send an EXCLUDE report in which the source-list parameter is empty, which is known as an

EXCLUDE NULL report. An EXCLUDE NULL report indicates that the host wants to join the multicast group and receive packets from all sources.

A Layer 2 devices support IGMPv3 membership reports that are in INCLUDE and EXCLUDE mode. However, SRX Series devices do not support forwarding on a per-source basis. Instead, the device consolidates all INCLUDE and EXCLUDE mode reports it receives on a VLAN for a specified group into a single route that includes all multicast sources for that group, with the next hop being all interfaces that have interested receivers for the group. As a result, interested receivers on the VLAN can receive traffic from a source that they did not include in their INCLUDE report or from a source they excluded in their EXCLUDE report. For example, if Host 1 wants traffic for G from Source A and Host 2 wants traffic for G from Source B, they both receive traffic for G regardless of whether A or B sends the traffic.

IGMP Snooping and Forwarding Interfaces

To determine how to forward multicast traffic, the device with IGMP snooping enabled maintains information about the following interfaces in its multicast forwarding table:

- Multicast-router interfaces—These interfaces lead toward multicast routers or IGMP queriers.
- Group-member interfaces—These interfaces lead toward hosts that are members of multicast groups.

The device learns about these interfaces by monitoring IGMP traffic. If an interface receives IGMP queries or Protocol Independent Multicast (PIM) updates, the device adds the interface to its multicast forwarding table as a multicast-router interface. If an interface receives membership reports for a multicast group, the device adds the interface to its multicast forwarding table as a group-member interface.

Table entries for interfaces that the device learns about are subject to aging. For example, if a learned multicast-router interface does not receive IGMP queries or PIM hellos within a certain interval, the device removes the entry for that interface from its multicast forwarding table.

NOTE: **For the device to learn multicast-router interfaces and group-member interfaces, an IGMP querier must exist in the network. This is often a multicast router, but if there is no multicast router on the local network, you can configure the device itself to be an IGMP querier.**

You can statically configure an interface to be a multicast-router interface or a group-member interface. The device adds a static interface to its multicast forwarding table without having to learn about the interface, and the entry in the table is not subject to aging. You can have a mix of statically configured and dynamically learned interfaces on the device.

General Forwarding Rules

Multicast traffic received on the device interface in a VLAN on which IGMP snooping is enabled is forwarded according to the following rules.

IGMP traffic is forwarded as follows:

- IGMP general queries received on a multicast-router interface are forwarded to all other interfaces in the VLAN.
- IGMP group-specific queries received on a multicast-router interface are forwarded to only those interfaces in the VLAN that are members of the group.
- IGMP reports received on a host interface are forwarded to multicast-router interfaces in the same VLAN, but not to the other host interfaces in the VLAN.

Multicast traffic that is not IGMP traffic is forwarded as follows:

- A multicast packet with a destination address of 233.252.0.0/24 is flooded to all other interfaces on the VLAN.
- An unregistered multicast packet—that is, a packet for a group that has no current members—is forwarded to all multicast-router interfaces in the VLAN.
- A registered multicast packet is forwarded only to those host interfaces in the VLAN that are members of the multicast group and to all multicast-router interfaces in the VLAN.

Using the Device as an IGMP Querier

If IGMP snooping is enabled on a pure Layer 2 a local network (that is, Layer 3 is not enabled on the network), and there is no multicast router in the network, multicast traffic might not be properly forwarded through the network. This problem occurs if the local network is configured such that multicast traffic must be forwarded between devices in order to reach a multicast receiver. In this case, an upstream device does not forward multicast traffic to a downstream device (and therefore to the multicast receivers attached to the downstream device) because the downstream device does not forward IGMP reports to the upstream device. You can solve this problem by configuring one of the devices to be an IGMP querier. This device sends periodic general query packets to all the devices in the network, which ensures that the snooping membership tables are updated and prevents any multicast traffic loss.

If you configure multiple devices to be IGMP queriers, the device with the lowest (smallest) IGMP querier source address takes precedence and acts as the querier. The devices with higher IGMP querier source addresses stop sending IGMP queries unless they do not receive IGMP queries for 255 seconds. If the device with a higher IGMP querier source address does not receive any IGMP queries during that period, it starts sending queries again.

NOTE: The **igmp-querier** statement is supported on QFabric systems in Junos OS Release 14.1X53-D15 but is not supported in Junos OS 15.1.

The **igmp-querier** statement is supported on QFX systems in Junos OS releases up to but not including Junos OS Release 15.1. It is supported in Junos OS Release 15.2 and later releases. It is not supported in Junos OS Release 15.1.

To configure a standalone device to act as an IGMP querier, enter the following:

```
[edit protocols]
user@host# set igmp-snooping vlan vlan-name l2-querier source-address source address
```

To configure a QFabric Node device switch to act as an IGMP querier, enter the following:

```
[edit protocols]
user@host# set igmp-snooping vlan vlan-name igmp-querier source-address source address
```

Release History Table

Release	Description
15.1	The **igmp-querier** statement is supported on QFX systems in Junos OS releases up to but not including Junos OS Release 15.1. It is supported in Junos OS Release 15.2 and later releases. It is not supported in Junos OS Release 15.1.
14.1X53-D15	The **igmp-querier** statement is supported on QFabric systems in Junos OS Release 14.1X53-D15 but is not supported in Junos OS 15.1.

Related Documentation

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- Example: Configuring IGMP Snooping on Switches on page 93
- Configuring IGMP Snooping on Switches on page 88
- Monitoring IGMP Snooping on page 96
- Configuring IGMP on page 27
- RFC 3171, *IANA Guidelines for IPv4 Multicast Address Assignments*
- IGMPv1—See RFC 1112, *Host extensions for IP multicasting*.
- IGMPv2—See RFC 2236, *Internet Group Management Protocol, Version 2*.
- IGMPv3—See RFC 3376, *Internet Group Management Protocol, Version 3*.

Overview of IGMP Snooping in an EVPN-VXLAN Environment

IGMP snooping is used to constrain multicast traffic in a broadcast domain to interested receivers and multicast devices. In an environment with significant multicast traffic, using IGMP snooping preserves bandwidth because multicast traffic is forwarded only on those interfaces where there are IGMP listeners. Starting with Junos OS Release 17.2R1, IGMP snooping is supported in an Ethernet VPN (EVPN) environment on QFX10000 switches. Previously, IGMP snooping was supported only with virtual LANs. As a result, you can now configure IGMP snooping in an EVPN-Virtual Extensible LAN (VXLAN) environment.

IGMP snooping in an EVPN-VXLAN multihomed environment is useful in data center interconnect (DCI) scenarios with a high level of multicast traffic. Every customer edge (CE) device does not need to receive every instance of multicast traffic. Enabling a provider edge (PE) device to send multicast traffic to a CE device only as needed utilizes the links between CE and PE devices more efficiently. When multicast traffic arrives at the VXLAN core, a remote PE device configured with EVPN forwards traffic only to the access interfaces where there are IGMP listeners. The multicast senders and receivers can be on the same site or on different sites. A site can have either only receiver, only source or both source and receiver attached to it. This implementation relies on IRB interfaces. Both intra-VLAN and inter-VLAN forwarding are supported. Use Protocol Independent Multicast (PIM) for inter-VLAN forwarding.

In this implementation, every link between a CE and a PE device must support IGMP snooping. Only proxy mode is supported with IGMP snooping. All multihomed interfaces must have the same configuration. Only the active-active mode of multihoming is supported. Active-standby mode is not supported. Only IGMP version 2 is supported. IGMP versions 1 and 3 are not supported. Additionally, the **multicast-options** configuration stanza is supported.

Starting with Junos OS Release 17.3R1, you can configure the PE device to perform inter-VLAN forwarding of multicast traffic without having to configure IRB interfaces. In such a scenario, an external multicast router is used to send IGMP queries to solicit reports and to forward VLAN traffic through a Layer 3 multicast protocol such as PIM. IRB interfaces are not supported with the use of an external multicast router.

Also starting with Junos OS Release 17.3R1, you can connect a PE device to an external Layer 3 device running PIM. This implementation relies on IRB interfaces and supports sending and receiving multicast traffic to and from the data center through an external gateway running PIM. To enable the PEs to forward traffic to the external domain, configure PIM-to-IGMP translation by including the **pim-to-igmp-proxy upstream-interface** ***irb-interface-name*** statements at the **[edit routing-options multicast]** hierarchy level. Additionally, you an now configure PIM on the PE so that it functions only to forward inter-VLAN traffic within the data center. This means that you do not need to configure a PIM rendezvous point since forming PIM adjacencies is not required. The gateway device only needs to view the data center as a Layer 2 multicast domain. Include the **passive** statement at the **[edit protocols pim]** hierarchy level.

Release History Table

Release	Description
17.3R1	Starting with Junos OS Release 17.3R1, you can configure the PE device to perform inter-VLAN forwarding of multicast traffic without having to configure IRB interfaces.
17.3R1	Also starting with Junos OS Release 17.3R1, you can connect a PE device to an external Layer 3 device running PIM.
17.2R1	Starting with Junos OS Release 17.2R1, IGMP snooping is supported in an Ethernet VPN (EVPN) environment on QFX10000 switches.

Related Documentation

- distributed-dr on page 1014
- igmp-snooping on page 1077
- multicast-router-interface on page 1172
- *Example: Preserving Bandwidth with IGMP Snooping in an EVPN-VXLAN Environment*

Configuring IGMP Snooping on Switches

With IGMP snooping enabled, the Juniper Networks device monitors the IGMP (Internet Group Management Protocol) traffic between hosts and multicast routers and uses what it learns to forward multicast traffic to only the downstream interfaces that are connected to interested receivers. This conserves bandwidth by allowing the device to send multicast traffic to only those interfaces that are connected to devices that want to receive the traffic (instead of flooding the traffic to all the downstream VLAN interfaces).

NOTE: You cannot configure IGMP snooping on a secondary VLAN.

NOTE: Starting in Junos OS Release 14.1X53 support for the igmp-querier statement is provided, The igmp-querier statement is not supported on QFabric switches. In Junos OS Release 15.2, support for the igmp-querier statement is restored.

To enable IGMP snooping and configure individual options as needed for your network by using the CLI:

1. Enable IGMP snooping on a VLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan employee-vlan
```

2. Configure the switch to immediately remove group membership from interfaces on a VLAN when it receives a leave message through that VLAN, and have it not forward any membership queries for the multicast group to the VLAN (IGMPv2 only):

```
[edit protocols]
user@switch# set igmp-snooping vlan vlan-name immediate-leave
```

3. Configure an interface to belong to a multicast group:

```
[edit protocols]
user@switch# set igmp-snooping vlan-name interface interface-name static group
group-address
```

4. Configure an interface to forward IGMP queries received from multicast routers.

```
[edit protocols]
user@switch# set igmp-snooping vlan vlan-name interface interface-name
multicast-router-interface
```

5. Configure the switch to wait for four timeout intervals before timing out a multicast group on a VLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan vlan-name robust-count 4
```

6. If you want a standalone switch to act as an IGMP querier, enter the following:

```
[edit protocols]
user@switch# set igmp-snooping vlan vlan-name l2-querier source-address source address
```

The switch uses the address that you configure as the source address in the IGMP queries that it sends. If there are any multicast routers on the same local network, make sure the source address for the IGMP querier is greater (a higher number) than the IP addresses for those routers on the network. This ensures that switch is always the IGMP querier on the network.

7. If you want a QFabric Node device to act as an IGMP querier, enter the following:

```
[edit protocols]
user@switch# set igmp-snooping vlan vlan-name igmp-querier source-address source address
```

Release History Table

Release	Description
14.1X53	Starting in Junos OS Release 14.1X53 support for the **igmp-querier** statement is provided, The **igmp-querier** statement is not supported on QFabric switches. In Junos OS Release 15.2, support for the **igmp-querier** statement is restored.

Related Documentation

- IGMP Snooping Overview on page 81
- Example: Configuring IGMP Snooping on Switches on page 93
- Monitoring IGMP Snooping on page 96

Example: Configuring IGMP Snooping on EX Series Switches

You can enable IGMP snooping on a VLAN to constrain the flooding of IPv4 multicast traffic on a VLAN. When IGMP snooping is enabled, a switch examines IGMP messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the switch then forwards multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

This example describes how to configure IGMP snooping:

Requirements

This example uses the following software and hardware components:

- One EX4300 Series switch
- Junos OS Release 13.2 or later for EX Series switches

Before you configure IGMP snooping, be sure you have:

- Configured the **vlan100** VLAN on the switch
- Assigned interfaces **ge-0/0/0**, **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/12** to vlan100
- Configure **ge-0/0/12** as a trunk interface.

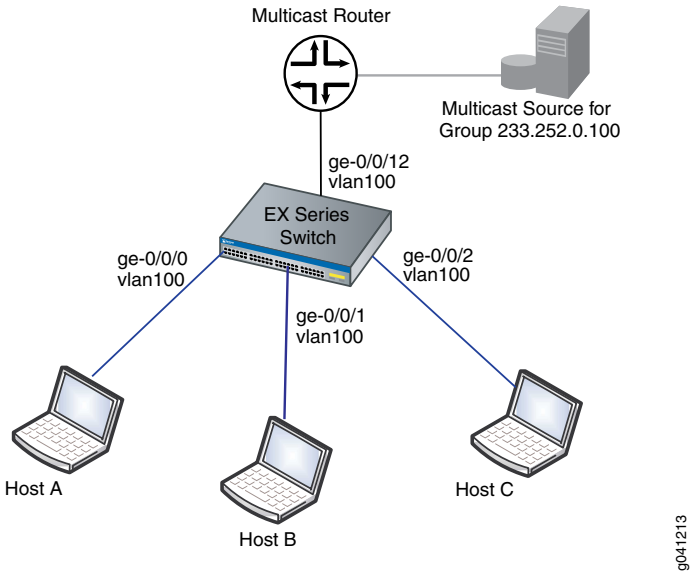
See <add topic-ref to the Configuring VLANs topic>.

Overview and Topology

In this example, interfaces **ge-0/0/0**, **ge-0/0/1**, and **ge-0/0/2** on the switch are in **vlan100** and are connected to hosts that are potential multicast receivers. Interface **ge-0/0/12**, a trunk interface also in **vlan100**, is connected to a multicast router. The router acts as the IGMP querier and forwards multicast traffic for group **255.100.100.100** to the switch from a multicast source.

The example topology is illustrated in Figure 9 on page 91.

Figure 9: Example IGMP Snooping Topology



In this example topology, the multicast router forwards multicast traffic to the switch from the source when it receives a memberhsip report for group **255.100.100.100** from one of the hosts—for example, Host B. If IGMP snooping is not enabled on **vlan100**, the switch floods the multicast traffic on all interfaces in **vlan100** (except for interface **ge-0/0/12**). If IGMP snooping is enabled on **vlan100**, the switch monitors the IGMP messages between the hosts and router, allowing it to determine that only Host B is interested in receiving the multicast traffic. The switch then forwards the multicast traffic only to interface **ge-0/0/1**.

IGMP snooping is enabled on all VLANs in the default factory configuration. For many implementations, IGMP snooping requires no additional configuration. This example shows how to perform the following optional configurations, which can reduce group join and leave latency:

- Configure immediate leave on the VLAN. When immediate leave is configured, the switch stops forwarding multicast traffic on an interface when it detects that the last member of the multicast group has left the group. If immediate leave is not configured, the switch waits until the group-specific queries time out before it stops forwarding traffic.

  Immediate leave is supported by IGMP version 2 (IGMPv2) and IGMPv3. With IGMPv2, we recommend that you configure immediate leave only when there is only one IGMP host on an interface. In IGMPv2, only one host on a interface sends a membership report in response to a group-specifc query—any other interested hosts suppress their reports to avoid a flood of reports for the same group. This report-suppression feature means that the switch only knows about one interested host at any given time.

- Configure **ge-0/0/12** as a static multicast-router interface. In this topology, **ge-0/0/12** always leads to the multicast router. By statically configuring **ge-0/0/12** as a multicast-router interface, you avoid any delay imposed by the switch having to learn that **ge-0/0/12** is a multicast-router interface.

Configuration

To configure IGMP snooping on a switch:

CLI Quick Configuration

To quickly configure IGMP snooping, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols igmp-snooping vlan vlan100 immediate-leave
set protocols igmp-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
```

Step-by-Step Procedure

To configure IGMP snooping on vlan100:

1. Configure the switch to immediately remove a group membership from an interface when it receives a leave report from the last member of the group on the interface:

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan vlan100 immediate-leave
   ```

2. Statically configure interface **ge-0/0/12** as a multicast-router interface:

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan vlan100 interface ge-0/0/12
   multicast-router-interface
   ```

Results

Check the results of the configuration:

```
[edit protocols]
user@switch# show igmp-snooping
vlan all;
vlan vlan100 {
    immediate-leave;
    interface ge-0/0/12.0 {
        multicast-router-interface;
    }
}
```

Verifying IGMP Snooping Operation

To verify that IGMP snooping is operating as configured, perform the following task:

- Displaying IGMP Snooping Information for VLAN vlan100 on page 92

Displaying IGMP Snooping Information for VLAN vlan100

Purpose

Verify that IGMP snooping is enabled on **vlan100** and that **ge-0/0/12** is recognized as a multicast-router interface.

Action Enter the following command:

```
user@switch> show igmp-snooping vlans vlan vlan100 detail
VLAN: vlan100, Tag: 100
    Interface: ge-0/0/12.0, tagged, Groups: 0, Router
```

Meaning By showing information for **vlan100**, the command output confirms that IGMP snooping is configured on the VLAN. Interface **ge-0/0/12.0** is listed as multicast-router interface, as configured. Because none of the host interfaces are listed, none of the hosts are currently receivers for the multicast group.

Related Documentation

- Configuring IGMP Snooping on Switches on page 88
- Verifying IGMP Snooping on EX Series Switches on page 97
- IGMP Snooping Overview on page 81

Example: Configuring IGMP Snooping on Switches

With IGMP snooping enabled, the Juniper Networks device monitors the IGMP (Internet Group Management Protocol) traffic between hosts and multicast routers and uses what it learns to forward multicast traffic to only the downstream interfaces that are connected to interested receivers. This conserves bandwidth by allowing the device to send multicast traffic to only those interfaces that are connected to devices that want to receive the traffic (instead of flooding the traffic to all the downstream VLAN interfaces).

This example describes how to configure IGMP snooping:

- Requirements on page 93
- Overview and Topology on page 93
- Configuration on page 94

Requirements

This example requires Junos OS Release 11.1 or later on a QFX Series product.

Before you configure IGMP snooping, be sure you have:

- Configured the **employee-vlan** VLAN
- Assigned interfaces **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/3** to **employee-vlan**

Overview and Topology

In this example you configure an interface to receive multicast traffic from a source and configure some multicast-related behavior for downstream interfaces. The example assumes that IGMP snooping was previously disabled for the VLAN.

Table 6 on page 94 shows the components of the topology for this example.

Table 6: Components of the IGMP Snooping Topology

Components	Settings
VLAN name	**employee-vlan**, tag **20**
Interfaces in **employee-vlan**	**ge-0/0/1**, **ge-0/0/2**, **ge-0/0/3**
Multicast IP address for **employee-vlan**	**225.100.100.100**

Configuration

To configure basic IGMP snooping on a switch:

CLI Quick Configuration

To quickly configure IGMP snooping, copy the following commands and paste them into a terminal window:

```
[edit protocols]
set igmp-snooping vlan employee-vlan
set igmp-snooping vlan employee-vlan interface ge-0/0/3 static group 225.100.100.100
set igmp-snooping vlan employee-vlan interface ge-0/0/2 multicast-router-interface
set igmp-snooping vlan employee-vlan robust-count 4
```

Step-by-Step Procedure

Configure IGMP snooping:

1. Enable and configure IGMP snooping on the VLAN **employee-vlan**:

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan employee-vlan
   ```

2. Configure a interface to belong to a multicast group:

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan employee-vlan interface ge-0/0/3 static group
   225.100.100.100
   ```

3. Configure an interface to forward IGMP queries received from multicast routers.

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan employee-vlan interface ge-0/0/2
   multicast-router-interface
   ```

4. Configure the switch to wait for four timeout intervals before timing out a multicast group on a VLAN:

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan employee-vlan robust-count 4
   ```

Results

Check the results of the configuration:

```
user@switch# show protocols igmp-snooping
  vlan employee-vlan {
```

```
    robust-count 4;
    }
    interface ge-0/0/2 {
      multicast-router-interface;
    }
    interface ge-0/0/3 {
      static {
        group 255.100.100.100;
      }
    }
  }
```

Related Documentation

- IGMP Snooping Overview on page 81
- Configuring IGMP Snooping on Switches on page 88
- Changing the IGMP Snooping Group Timeout Value on Switches on page 95
- Monitoring IGMP Snooping on page 96

Changing the IGMP Snooping Group Timeout Value on Switches

The IGMP snooping group timeout value determines how long a switch waits to receive an IGMP query from a multicast router before removing a multicast group from its multicast cache table. A switch calculates the timeout value by using the **query-interval** and **query-response-interval** values.

When you enable IGMP snooping, the **query-interval** and **query-response-interval** values are applied to all VLANs on the switch. The values are:

- **query-interval**—125 seconds
- **query-response-interval**—10 seconds

The switch automatically calculates the group timeout value for an IGMP snooping-enabled switch by multiplying the **query-interval** value by 2 (the default **robust-count** value) and then adding the **query-response-interval** value. By default, the switch waits 260 seconds to receive an IGMP query before removing a multicast group from its multicast cache table: (125 x 2) + 10 = 260.

You can modify the group timeout value by changing the **robust-count** value. For example, if you want the system to wait 510 seconds before timing groups out—(125 x 4) + 10 = 510—enter this command:

```
[edit protocols]
user@switch# set igmp-snooping vlan employee-vlan robust-count 4
```

Related Documentation

- Verifying IGMP Snooping on EX Series Switches on page 97
- Example: Configuring IGMP Snooping on Switches on page 93
- Configuring IGMP Snooping on Switches on page 88

Monitoring IGMP Snooping

Purpose Use the monitoring feature to view status and information about the IGMP snooping configuration.

Action To display details about IGMP snooping, enter the following operational commands:

- **show igmp snooping interface**—Display information about interfaces enabled with IGMP snooping, including which interfaces are being snooped in a learning domain and the number of groups on each interface.
- **show igmp snooping membership**—Display IGMP snooping membership information, including the multicast group address and the number of active multicast groups.
- **show igmp snooping options**—Display brief or detailed information about IGMP snooping.
- **show igmp snooping statistics**—Display IGMP snooping statistics, including the number of messages sent and received.

The **show igmp snooping interface**, **show igmp snooping membership**, and **show igmp snooping statistics** commands also support the following options:

- **instance** *instance-name*
- **interface** *interface-name*
- **qualified-vlan** *vlan-identifier*
- **vlan** *vlan-name*

Meaning Table 7 on page 96 summarizes the IGMP snooping details displayed.

Table 7: Summary of IGMP Snooping Output Fields

Field	Values
IGMP Snooping Monitor	
VLAN	VLAN for which IGMP snooping is enabled.
Interfaces	Interface connected to a multicast router.
Groups	Number of the multicast groups learned by the VLAN.
MRouters	Multicast router.
Receivers	Multicast receiver.
IGMP Route Information	
VLAN	VLAN for which IGMP snooping is enabled.
Next-Hop	Next hop assigned by the switch after performing the route lookup.

Table 7: Summary of IGMP Snooping Output Fields (continued)

Field	Values
Group	Multicast groups learned by the VLAN.

Related Documentation

- IGMP Snooping Overview on page 81
- Example: Configuring IGMP Snooping on Switches on page 93
- Configuring IGMP Snooping on Switches on page 88
- Changing the IGMP Snooping Group Timeout Value on Switches on page 95

Verifying IGMP Snooping on EX Series Switches

Internet Group Management Protocol (IGMP) snooping constrains the flooding of IPv4 multicast traffic on VLANs on a switch. This topic describes how to verify IGMP snooping operation on the switch.

It covers:

- Verifying IGMP Snooping Memberships on page 97
- Viewing IGMP Snooping Statistics on page 98
- Viewing IGMP Snooping Routing Information on page 99

Verifying IGMP Snooping Memberships

Purpose Determine group memberships, multicast-router interfaces, host IGMP versions, and the current values of timeout counters.

Action Enter the following command:

```
user@switch> show igmp snooping membership detail
VLAN: vlan2 Tag: 2 (Index: 3)
    Router interfaces:
        ge-1/0/0.0 dynamic Uptime: 00:14:24 timeout: 253
    Group: 233.252.0.1
      ge-1/0/17.0 259 Last reporter: 13.0.0.90 Receiver count: 1
      Uptime: 00:00:19 timeout: 259 Flags: <V3-hosts>
        Include source: 10.2.11.5, 10.2.11.12
```

Meaning The switch has multicast membership information for one VLAN on the switch, **vlan2**. IGMP snooping might be enabled on other VLANs, but the switch does not have any multicast membership information for them. The following information is provided:

- Information on the multicast-router interfaces for the VLAN—in this case, **ge-1/0/0.0**. The multicast-router interface has been learned by IGMP snooping, as indicated by the dynamic value. The timeout value shows how many seconds from now the interface

will be removed from the multicast forwarding table if the switch does not receive IGMP queries or Protocol Independent Multicast (PIM) updates on the interface.

- Information about the group memberships for the VLAN:
  - Currently, the VLAN has membership in only one multicast group, **233.252.0.1**.
  - The host or hosts that have reported membership in the group are on interface **ge-1/0/17.0**. The last host that reported membership in the group has address **10.0.0.90**. The number of hosts belonging to the group on the interface is shown in the Receiver count field, which is displayed only when host tracking is enabled if immediate leave is configured on the VLAN.
  - The Uptime field shows that the multicast group has been active on the interface for 19 seconds. The interface group membership will time out in 259 seconds if no hosts respond to membership queries during this interval. The Flags field shows the lowest version of IGMP used by a host that is currently a member of the group, which in this case is IGMP version 3 (IGMPv3).
  - Because the interface has IGMPv3 hosts on it, the source addresses from which the IGMPv3 hosts want to receive group multicast traffic are shown (addresses **10.2.11.5** and **10.2.11.12**). The timeout value for the interface group membership is derived from the largest timeout value for all sources addresses for the group.

Viewing IGMP Snooping Statistics

Purpose Display IGMP snooping statistics, such as number of IGMP queries, reports, and leaves received and how many of these IGMP messages contained errors.

Action Enter the following command:

```
user@switch> show igmp snooping statistics
Bad length: 0 Bad checksum: 0 Invalid interface: 0
Not local: 0 Receive unknown: 0 Timed out: 0

    IGMP Type           Received      Transmitted    Recv Errors
    Queries:               74295                0              0
    Reports:            18148423                0       16333523
    Leaves:                    0                0              0
    Other:                     0                0              0
```

Meaning The output shows how many IGMP messages of each type—**Queries**, **Reports**, **Leaves**—the switch received or transmitted on interfaces on which IGMP snooping is enabled. For each message type, it also shows the number of IGMP packets the switch received that had errors—for example, packets that do not conform to the IGMPv1, IGMPv2, or IGMPv3 standards. If the **Recv Errors** count increases, verify that the hosts are compliant with IGMP standards. If the switch is unable to recognize the IGMP message type for a packet, it counts the packet under **Receive unknown**.

Viewing IGMP Snooping Routing Information

Purpose Display the next-hop information maintained in the multicast forwarding table.

Action Enter the following command:

```
user@switch> show multicast snooping route vlan
```

Meaning The output shows the next-hop interfaces for a given multicast group on a VLAN.

Related Documentation

- clear igmp snooping membership on page 1397
- Example: Configuring IGMP Snooping on EX Series Switches on page 90
- Configuring IGMP Snooping on Switches on page 88

Example: Configuring IGMP Snooping

Understanding Multicast Snooping

Network devices such as routers operate mainly at the packet level, or Layer 3. Other network devices such as bridges or LAN switches operate mainly at the frame level, or Layer 2. Multicasting functions mainly at the packet level, Layer 3, but there is a way to map Layer 3 IP multicast group addresses to Layer 2 MAC multicast group addresses at the frame level.

Routers can handle both Layer 2 and Layer 3 addressing information because the frame and its addresses must be processed to access the encapsulated packet inside. Routers can run Layer 3 multicast protocols such as PIM or IGMP and determine where to forward multicast content or when a host on an interface joins or leaves a group. However, bridges

and LAN switches, as Layer 2 devices, are not supposed to have access to the multicast information inside the packets that their frames carry.

How then are bridges and other Layer 2 devices to determine when a device on an interface joins or leaves a multicast tree, or whether a host on an attached LAN wants to receive the content of a particular multicast group?

The answer is for the Layer 2 device to implement multicast snooping. Multicast snooping is a general term and applies to the process of a Layer 2 device "snooping" at the Layer 3 packet content to determine which actions are taken to process or forward a frame. There are more specific forms of snooping, such as IGMP snooping or PIM snooping. In all cases, snooping involves a device configured to function at Layer 2 having access to normally "forbidden" Layer 3 (packet) information. Snooping makes multicasting more efficient in these devices.

See Also

- Layer 2 Frames and IPv4 Multicast Addresses on page 9

Understanding IGMP Snooping

Snooping is a general way for Layer 2 devices, such as Juniper Networks MX Series Ethernet Services Routers, to implement a series of procedures to "snoop" at the Layer 3 packet content to determine which actions are to be taken to process or forward a frame. More specific forms of snooping, such as Internet Group Membership Protocol (IGMP) snooping or Protocol Independent Multicast (PIM) snooping, are used with multicast.

Layer 2 devices (LAN switches or bridges) handle multicast packets and the frames that contain them much in the same way the Layer 3 devices (routers) handle broadcasts. So, a Layer 2 switch processes an arriving frame having a multicast destination media access control (MAC) address by forwarding a copy of the packet (frame) onto each of the other network interfaces of the switch that are in a forwarding state.

However, this approach (sending multicast frames everywhere the device can) is not the most efficient use of network bandwidth, particularly for IPTV applications. IGMP snooping functions by "snooping" at the IGMP packets received by the switch interfaces and building a multicast database similar to that a multicast router builds in a Layer 3 network. Using this database, the switch can forward multicast traffic only onto downstream interfaces with interested receivers, and this technique allows more efficient use of network bandwidth.

You configure IGMP snooping for each bridge on the router. A bridge instance without qualified learning has just one learning domain. For a bridge instance with qualified learning, snooping will function separately within each learning domain in the bridge. That is, IGMP snooping and multicast forwarding will proceed independently in each learning domain in the bridge.

This discussion focuses on bridge instances without qualified learning (those forming one learning domain on the device). Therefore, all the interfaces mentioned are logical interfaces of the bridge or VPLS instance.

Several related concepts are important when discussing IGMP snooping:

- Bridge or VPLS instance interfaces are either multicast-router interfaces or host-side interfaces.
- IGMP snooping supports proxy mode or without-proxy mode.

NOTE: When integrated routing and bridging (IRB) is used, if the router is an IGMP querier, any leave message received on any Layer 2 interface will cause a group-specific query on all Layer 2 interfaces (as a result of this practice, some corresponding reports might be received on all Layer 2 interfaces). However, if some of the Layer 2 interfaces are also router (Layer 3) interfaces, reports and leaves from other Layer 2 interfaces will not be forwarded on those interfaces.

If an IRB interface is used as an outgoing interface in a multicast forwarding cache entry (as determined by the routing process), then the output interface list is expanded into a subset of the Layer 2 interface in the corresponding bridge. The subset is based on the snooped multicast membership information, according to the multicast forwarding cache entry installed by the snooping process for the bridge.

If no snooping is configured, the IRB output interface list is expanded to all Layer 2 interfaces in the bridge.

The Junos OS does not support IGMP snooping in a VPLS configuration on a virtual switch. This configuration is disallowed in the CLI.

NOTE: IGMP snooping is supported on AE interfaces, however, it is not supported on AE interfaces in combination with IRB interfaces.

See Also

- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99
- Example: Configuring IGMP Snooping on page 106
- *IGMP Snooping in MC-LAG Active-Active Mode*

IGMP Snooping Interfaces and Forwarding

IGMP snooping divides the device interfaces into multicast-router interfaces and host-side interfaces. A multicast-router interface is an interface in the direction of a multicasting router. An interface on the bridge is considered a multicast-router interface if it meets at least one of the following criteria:

- It is statically configured as a multicast-router interface in the bridge instance.
- IGMP queries are being received on the interface.

All other interfaces that are not multicast-router interfaces are considered host-side interfaces.

Any multicast traffic received on a bridge interface with IGMP snooping configured will be forwarded according to following rules:

- Any IGMP packet is sent to the Routing Engine for snooping processing.
- Other multicast traffic with destination address 224.0.0/24 is flooded onto all other interfaces of the bridge.
- Other multicast traffic is sent to all the multicast-router interfaces but only to those host-side interfaces that have hosts interested in receiving that multicast group.

See Also

- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

IGMP Snooping and Proxies

Without a proxy arrangement, IGMP snooping does not generate or introduce queries and reports. It will only “snoop” reports received from all of its interfaces (including multicast-router interfaces) to build its state and group (S,G) database.

Without a proxy, IGMP messages are processed as follows:

- Query—All general and group-specific IGMP query messages received on a multicast-router interface are forwarded to all other interfaces (both multicast-router interfaces and host-side interfaces) on the bridge.
- Report—IGMP reports received on any interface of the bridge are forwarded toward other multicast-router interfaces. The receiving interface is added as an interface for that group if a multicast routing entry exists for this group. Also, a group timer is set for the group on that interface. If this timer expires (that is, there was no report for this group during the IGMP group timer period), then the interface is removed as an interface for that group.
- Leave—Any IGMP leave message received on any interface of the bridge. The Leave Group message reduces the time it takes for the multicast router to stop forwarding multicast traffic when there are no longer any members in the host group.

Proxy snooping reduces the number of IGMP reports sent toward an IGMP router.

NOTE: **With proxy snooping configured, an IGMP router is not able to perform host tracking.**

As proxy for its host-side interfaces, IGMP snooping in proxy mode replies to the queries it receives from an IGMP router on a multicast-router interface. On the host-side interfaces, IGMP snooping in proxy mode behaves as an IGMP router and sends general and group-specific queries on those interfaces.

NOTE: Only group-specific queries are generated by IGMP snooping directly. General queries received from the multicast-router interfaces are flooded to host-side interfaces.

All the queries generated by IGMP snooping are sent using 0.0.0.0 as the source address. Also, all reports generated by IGMP snooping are sent with 0.0.0.0 as the source address unless there is a configured source address to use.

Proxy mode functions differently on multicast-router interfaces than it does on host-side interfaces.

See Also

- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

Multicast-Router Interfaces and IGMP Snooping Proxy Mode

On multicast-router interfaces, in response to IGMP queries, IGMP snooping in proxy mode sends reports containing aggregate information on groups learned on all host-side interfaces of the bridge.

Besides replying to queries, IGMP snooping in proxy mode forwards all queries, reports, and leaves received on a multicast-router interface to other multicast-router interfaces. IGMP snooping keeps the membership information learned on this interface but does not send a group-specific query for leave messages received on this interface. It simply times out the groups learned on this interface if there are no reports for the same group within the timer duration.

NOTE: For the hosts on all the multicast-router interfaces, it is the IGMP router, not the IGMP snooping proxy, that generates general and group-specific queries.

See Also

- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

Host-Side Interfaces and IGMP Snooping Proxy Mode

No reports are sent on host-side interfaces by IGMP snooping in proxy mode. IGMP snooping processes reports received on these interfaces and sends group-specific queries onto host-side interfaces when it receives a leave message on the interface. Host-side interfaces do not generate periodic general queries, but forwards or floods general queries received from multicast-router interfaces.

If a group is removed from a host-side interface and this was the last host-side interface for that group, a leave is sent to the multicast-router interfaces. If a group report is received

on a host-side interface and this was the first host-side interface for that group, a report is sent to all multicast-router interfaces.

See Also
- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

IGMP Snooping and Bridge Domains

IGMP snooping on a VLAN is only allowed for the legacy **vlan-id all** case. In other cases, there is a specific bridge domain configuration that determines the VLAN-specific configuration for IGMP snooping.

See Also
- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

Configuring IGMP Snooping

To configure Internet Group Management Protocol (IGMP) snooping, include the **igmp-snooping** statement:

```
igmp-snooping {
    immediate-leave;
    interface interface-name {
        group-limit limit;
        host-only-interface;
        immediate-leave;
        multicast-router-interface;
        static {
            group ip-address {
                source ip-address;
            }
        }
    }
    proxy {
        source-address ip-address;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    vlan vlan-id {
        immediate-leave;
        interface interface-name {
            group-limit limit;
            host-only-interface;
            immediate-leave;
            multicast-router-interface;
            static {
                group ip-address {
                    source ip-address;
                }
```

```
        }
      }
      proxy {
        source-address ip-address;
      }
      query-interval seconds;
      query-last-member-interval seconds;
      query-response-interval seconds;
      robust-count number;
    }
  }
```

You can include this statement at the following hierarchy levels:

- **[edit bridge-domains *bridge-domain-name* protocols]**
- **[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols]**

By default, IGMP snooping is not enabled. Statements configured at the VLAN level apply only to that particular VLAN.

See Also

- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

Configuring VLAN-Specific IGMP Snooping Parameters

All of the IGMP snooping statements configured with the **igmp-snooping** statement, with the exception of the **traceoptions** statement, can be qualified with the same statement at the VLAN level. To configure IGMP snooping parameters at the VLAN level, include the **vlan** statement:

```
vlan vlan-id;
  immediate-leave;
  interface interface-name {
    group-limit limit;
    host-only-interface;
    multicast-router-interface;
    static {
      group ip-address {
        source ip-address;
      }
    }
  }
  proxy {
    source-address ip-address;
  }
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
}
```

You can include this statement at the following hierarchy levels:

- **[edit bridge-domains *bridge-domain-name* protocols igmp-snooping]**
- **[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping]**

See Also
- Layer 2 Frames and IPv4 Multicast Addresses on page 9
- Understanding Multicast Snooping on page 99

Example: Configuring IGMP Snooping

This example shows how to configure IGMP snooping. IGMP snooping can reduce unnecessary traffic from IP multicast applications.

Requirements

This example uses the following hardware components:

- One MX Series router
- One Layer 3 device functioning as a multicast router

Before you begin:

- Configure the interfaces. See the *Interfaces Feature Guide for Security Devices*.
- Configure an interior gateway protocol. See the *Junos OS Routing Protocols Library*.
- Configure a multicast protocol. This feature works with the following multicast protocols:
  - DVMRP
  - PIM-DM
  - PIM-SM
  - PIM-SSM

Overview and Topology

IGMP snooping controls multicast traffic in a switched network. When IGMP snooping is not enabled, the Layer 2 device broadcasts multicast traffic out of all of its ports, even if the hosts on the network do not want the multicast traffic. With IGMP snooping enabled, a Layer 2 device monitors the IGMP join and leave messages sent from each connected host to a multicast router. This enables the Layer 2 device to keep track of the multicast groups and associated member ports. The Layer 2 device uses this information to make

intelligent decisions and to forward multicast traffic to only the intended destination hosts.

This example includes the following statements:

- **proxy**—Enables the Layer 2 device to actively filter IGMP packets to reduce load on the multicast router. Joins and leaves heading upstream to the multicast router are filtered so that the multicast router has a single entry for the group, regardless of how many active listeners have joined the group. When a listener leaves a group but other listeners remain in the group, the leave message is filtered because the multicast router does not need this information. The status of the group remains the same from the router's point of view.
- **immediate-leave**—When only one IGMP host is connected, the **immediate-leave** statement enables the multicast router to immediately remove the group membership from the interface and suppress the sending of any group-specific queries for the multicast group.

  When you configure this feature on IGMPv2 interfaces, ensure that the IGMP interface has only one IGMP host connected. If more than one IGMPv2 host is connected to a LAN through the same interface, and one host sends a leave message, the router removes all hosts on the interface from the multicast group. The router loses contact with the hosts that properly remain in the multicast group until they send join requests in response to the next general multicast listener query from the router.

  When IGMP snooping is enabled on a router running IGMP version 3 (IGMPv3) snooping, after the router receives a report with the type BLOCK_OLD_SOURCES, the router suppresses the sending of group-and-source queries but relies on the Junos OS host-tracking mechanism to determine whether or not it removes a particular source group membership from the interface.
- **query-interval**—Enables you to change the number of IGMP messages sent on the subnet by configuring the interval at which the IGMP querier router sends general host-query messages to solicit membership information.

  By default, the query interval is 125 seconds. You can configure any value in the range 1 through 1024 seconds.
- **query-last-member-interval**—Enables you to change the amount of time it takes a device to detect the loss of the last member of a group.

  The last-member query interval is the maximum amount of time between group-specific query messages, including those sent in response to leave-group messages.

  By default, the last-member query interval is 1 second. You can configure any value in the range 0.1 through 0.9 seconds, and then 1-second intervals from 1 through 1024 seconds.
- **query-response-interval**—Configures how long the router waits to receive a response from its host-query messages.

  By default, the query response interval is 10 seconds. You can configure any value in the range 1 through 1024 seconds. This interval should be less than the interval set in the **query-interval** statement.

- **robust-count**—Provides fine-tuning to allow for expected packet loss on a subnet. It is basically the number of intervals to wait before timing out a group. You can wait more intervals if subnet packet loss is high and IGMP report messages might be lost.

  By default, the robust count is 2. You can configure any value in the range 2 through 10 intervals.

- **group-limit**—Configures a limit for the number of multicast groups (or [S,G] channels in IGMPv3) that can join an interface. After this limit is reached, new reports are ignored and all related flows are discarded, not flooded.

  By default, there is no limit to the number of groups that can join an interface. You can configure a limit in the range 0 through a 32-bit number.

- **host-only-interface**—Configure an IGMP snooping interface to be an exclusively host-side interface. On a host-side interface, received IGMP queries are dropped.

  By default, an interface can face either other multicast routers or hosts.

- **multicast-router-interface**—Configures an IGMP snooping interface to be an exclusively router-facing interface.

  By default, an interface can face either other multicast routers or hosts.

- **static**—Configures an IGMP snooping interface with multicast groups statically.

  By default, the router learns about multicast groups on the interface dynamically.

Figure 10 on page 109 shows networks without IGMP snooping. Suppose host A is an IP multicast sender and hosts B and C are multicast receivers. The router forwards IP multicast traffic only to those segments with registered receivers (hosts B and C). However, the Layer 2 devices flood the traffic to all hosts on all interfaces.

Figure 10: Networks Without IGMP Snooping Configured

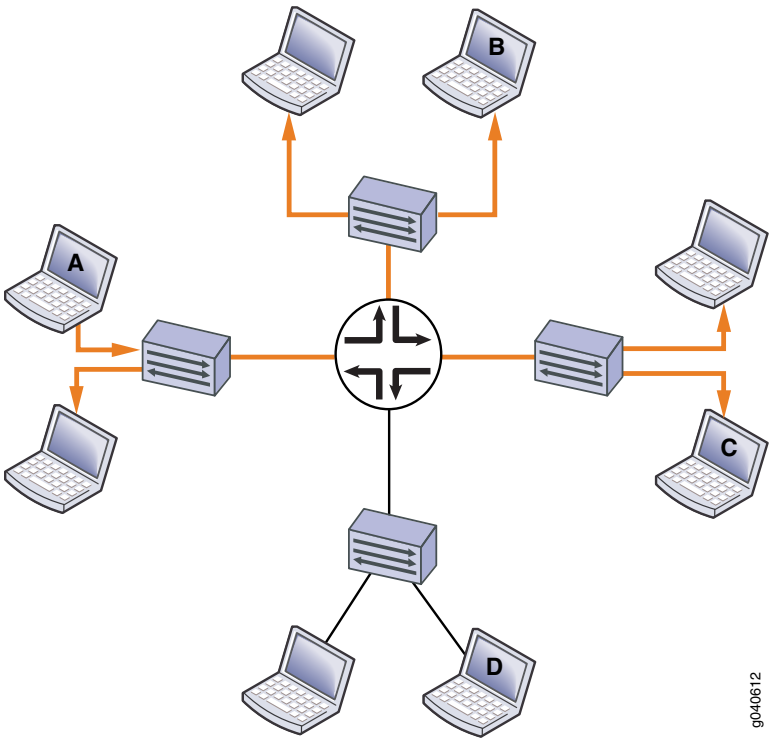
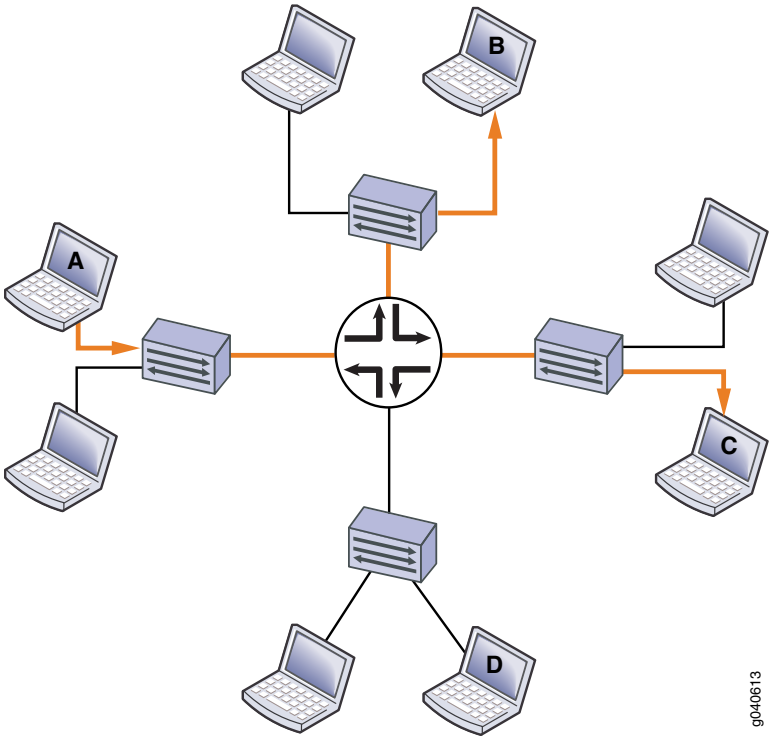


Figure 11 on page 110 shows the same networks with IGMP snooping configured. The Layer 2 devices forward multicast traffic to registered receivers only.

Figure 11: Networks with IGMP Snooping Configured



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set bridge-domains domain1 domain-type bridge
set bridge-domains domain1 interface ge-0/0/1.1
set bridge-domains domain1 interface ge-0/0/2.1
set bridge-domains domain1 interface ge-0/0/3.1
set bridge-domains domain1 protocols igmp-snooping query-interval 200
set bridge-domains domain1 protocols igmp-snooping query-response-interval 0.4
set bridge-domains domain1 protocols igmp-snooping query-last-member-interval 0.1
set bridge-domains domain1 protocols igmp-snooping robust-count 4
set bridge-domains domain1 protocols igmp-snooping immediate-leave
set bridge-domains domain1 protocols igmp-snooping proxy
set bridge-domains domain1 protocols igmp-snooping interface ge-0/0/1.1
  host-only-interface
set bridge-domains domain1 protocols igmp-snooping interface ge-0/0/1.1 group-limit
  50
set bridge-domains domain1 protocols igmp-snooping interface ge-0/0/3.1 static group
  225.100.100.100
set bridge-domains domain1 protocols igmp-snooping interface ge-0/0/2.1
  multicast-router-interface
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure IGMP snooping:

1. Configure the bridge domain.

```
[edit bridge-domains domain1]
user@host# set domain-type bridge
user@host# set interface ge-0/0/1.1
user@host# set interface ge-0/0/2.1
user@host# set interface ge-0/0/3.1
```

2. Enable IGMP snooping and configure the router to serve as a proxy.

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snooping proxy
```

3. Configure the limit for the number of multicast groups allowed on the **ge-0/0/1.1** interface to 50.

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snooping interface ge-0/0/1.1group-limit 50
```

4. Configure the router to immediately remove a group membership from an interface when it receives a leave message from that interface without waiting for any other IGMP messages to be exchanged.

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snooping immediate-leave
```

5. Statically configure IGMP group membership on a port.

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snooping interface ge-0/0/3.1 static group
  225.100.100.100
```

6. Configure an interface to be an exclusively router-facing interface (to receive multicast traffic).

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snooping interface ge-0/0/2.1
  multicast-router-interface
```

7. Configure an interface to be an exclusively host-facing interface (to drop IGMP query messages).

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snooping interface ge-0/0/1.1 host-only-interface
```

8. Configure the IGMP message intervals and robustness count.

```
[edit bridge-domains domain1]
user@host# set protocols igmp-snoopingrobust-count 4
user@host# set protocols igmp-snooping query-last-member-interval 0.1
user@host# set protocols igmp-snooping query-interval 200
user@host# set protocols igmp-snooping query-response-interval 0.4
```

9. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results Confirm your configuration by entering the **show bridge-domains** command.

```
user@host# show bridge-domains
domain1 {
  domain-type bridge;
  interface ge-0/0/1.1;
  interface ge-0/0/2.1;
  interface ge-0/0/3.1;
  protocols {
    igmp-snooping {
      query-interval 200;
      query-response-interval 0.4;
      query-last-member-interval 0.1;
      robust-count 4;
      immediate-leave;
      proxy;
      interface ge-0/0/1.1 {
        host-only-interface;
        group-limit 50;
      }
      interface ge-0/0/3.1 {
        static {
          group 225.100.100.100;
        }
      }
      interface ge-0/0/2.1 {
        multicast-router-interface;
      }
    }
  }
}
```

Verification

To verify the configuration, run the following commands:

- show igmp snooping interface
- show igmp snooping membership
- show igmp snooping statistics

See Also

- Understanding IGMP Snooping on page 100
- Host-Side Interfaces and IGMP Snooping Proxy Mode on page 103
- Multicast-Router Interfaces and IGMP Snooping Proxy Mode on page 103

Configuring IGMP Snooping Trace Operations

Tracing operations record detailed messages about the operation of routing protocols, such as the various types of routing protocol packets sent and received, and routing policy actions. You can specify which trace operations are logged by including specific tracing flags. The following table describes the flags that you can include.

Flag	Description
all	Trace all operations.
client-notification	Trace notifications.
general	Trace general flow.
group	Trace group operations.
host-notification	Trace host notifications.
leave	Trace leave group messages (IGMPv2 only).
normal	Trace normal events.
packets	Trace all IGMP packets.
policy	Trace policy processing.
query	Trace IGMP membership query messages.
report	Trace membership report messages.
route	Trace routing information.
state	Trace state transitions.
task	Trace routing protocol task processing.
timer	Trace timer processing.

You can configure tracing operations for IGMP snooping globally or in a routing instance. The following example shows the global configuration.

To configure tracing operations for IGMP snooping:

1. Configure the filename for the trace file.

   ```
   [edit bridge-domains domain1 protocols igmp-snooping traceoptions]
   user@host# set file igmp-snoop-trace
   ```

2. (Optional) Configure the maximum number of trace files.

   ```
   [edit bridge-domains domain1 protocols igmp-snooping traceoptions]
   user@host# set file files 5
   ```

3. (Optional) Configure the maximum size of each trace file.

   ```
   [edit bridge-domains domain1 protocols igmp-snooping traceoptions]
   user@host# set file size 1m
   ```

4. (Optional) Enable unrestricted file access.

   ```
   [edit bridge-domains domain1 protocols igmp-snooping traceoptions]
   user@host# set file world-readable
   ```

5. Configure tracing flags. Suppose you are troubleshooting issues with a policy related to received packets on a particular logical interface with an IP address of 192.168.0.1. The following example shows how to flag all policy events for received packets associated with the IP address.

   ```
   [edit bridge-domains domain1 protocols igmp-snooping traceoptions]
   user@host# set flag policy receive | match 192.168.0.1
   ```

6. View the trace file.

   ```
   user@host> file list /var/log
   user@host> file show /var/log/igmp-snoop-trace
   ```

See Also

- *Tracing and Logging Junos OS Operations*
- Configuring IGMP Snooping on page 104

Related Documentation

- Understanding Multicast Snooping on page 99

Example: Configuring IGMP Snooping on SRX Series Devices

You can enable IGMP snooping on a VLAN to constrain the flooding of IPv4 multicast traffic on a VLAN. When IGMP snooping is enabled, the device examines IGMP messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the device then forwards multicast traffic only to those interfaces that are connected to relevant receivers instead of flooding the traffic to all interfaces.

This example describes how to configure IGMP snooping:

Requirements

This example uses the following hardware and software components:

- One SRX Series device
- Junos OS Release 18.1R1

Before you configure IGMP snooping, be sure you have:

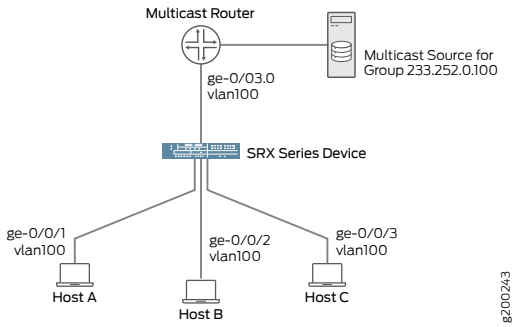
- Configured a VLAN, vlan100, on the device
- Assigned interfaces ge-0/0/1, ge-0/0/2, and ge-0/0/3 to vlan100
- Configured ge-0/0/3 as a trunk interface

Overview and Topology

IGMP snooping controls multicast traffic in a switched network. When IGMP snooping is not enabled, the SRX Series device broadcasts multicast traffic out of all of its ports, even if the hosts on the network do not want the multicast traffic. With IGMP snooping enabled, the SRX Series device monitors the IGMP join and leave messages sent from each connected host to a multicast router. This enables the SRX Series device to keep track of the multicast groups and associated member ports. The SRX Series device uses this information to make intelligent decisions and to forward multicast traffic to only the intended destination hosts.

The sample topology is illustrated in Figure 9 on page 91.

Figure 12: IGMP Snooping Sample Topology



In this sample topology, the multicast router forwards multicast traffic to the device from the source when it receives a membership report for group 233.252.0.100 from one of the hosts—for example, Host B. If IGMP snooping is not enabled on vlan100, the device

floods the multicast traffic on all interfaces in vlan100 (except for interface ge-0/0/2.0). If IGMP snooping is enabled on vlan100, the device monitors the IGMP messages between the hosts and router, allowing it to determine that only Host B is interested in receiving the multicast traffic. The device then forwards the multicast traffic only to interface ge-0/0/2.

Configuration

To configure IGMP snooping on a device:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interface ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interface ge-0/0/1 unit 0 family ethernet-switching vlan members v1
set interface ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interface ge-0/0/2 unit 0 family ethernet-switching vlan members v1
set interface ge-0/0/3 unit 0 family ethernet-switching interface-mode access
set interface ge-0/0/3 unit 0 family ethernet-switching vlan members v1
set vlans v1 vlan-id 100
set protocols igmp-snooping vlan v1 query-interval 200
set protocols igmp-snooping vlan v1 query-response-interval 0.4
set protocols igmp-snooping vlan v1 query-last-member-interval 0.1
set protocols igmp-snooping vlan v1 robust-count 4
set protocols igmp-snooping vlan v1 immediate-leave
set protocols igmp-snooping vlan v1 proxy
set protocols igmp-snooping vlan v1 interface ge-0/0/1.0 host-only-interface
set protocols igmp-snooping vlan v1 interface ge-0/0/1.0 group-limit 50
set protocols igmp-snooping vlan v1 interface ge-0/0/3.0 static group 233.252.0.100
set protocols igmp-snooping vlan v1 interface ge-0/0/2.0 multicast-router-interface
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure IGMP snooping:

1. Configure the access mode interfaces.

```
[edit interfaces]
user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/1 unit 0 family ethernet-switching vlan members v1
user@host# set ge-0/0/2 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/2 unit 0 family ethernet-switching vlan members v1
user@host# set ge-0/0/3 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/3 unit 0 family ethernet-switching vlan members v1
```

2. Configure the VLAN.

```
[edit vlans V1]
user@host# set vlans v1 vlan-id 100
```

3. Enable IGMP snooping and configure the device to serve as a proxy.

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 proxy
```

4. Configure the limit for the number of multicast groups allowed on the ge-0/0/1.0 interface to 50.

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 interface ge-0/0/1.0 group-limit
  50
```

5. Configure the device to immediately remove a group membership from an interface when it receives a leave message from that interface without waiting for any other IGMP messages to be exchanged.

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 immediate-leave
```

6. Statically configure interface ge-0/0/3.0 as a multicast-router interface.

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 interface ge-0/0/3.0 static group
  233.252.0.100
```

7. Configure an interface to be an exclusively router-facing interface (to receive multicast traffic).

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 interface ge-0/0/2.0
  multicast-router-interface
```

8. Configure an interface to be an exclusively host-facing interface (to drop IGMP query messages).

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 interface ge-0/0/1.0
  host-only-interface
```

9. Configure the IGMP message intervals and robustness count.

```
[edit vlans v1]
user@host# set protocols igmp-snooping vlan v1 query-interval 200
user@host# set protocols igmp-snooping vlan v1 query-response-interval 0.4
user@host# set protocols igmp-snooping vlan v1 query-last-member-interval 0.1
user@host# set protocols igmp-snooping vlan v1 robust-count 4
```

10. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results From configuration mode, confirm your configuration by entering the **show protocols igmp-snooping** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show protocols igmp-snooping
  vlan v1 {
    query-interval 200;
    query-response-interval 0.4;
    query-last-member-interval 0.1;
    robust-count 4;
    immediate-leave;
    proxy;
    interface ge-0/0/1.0 {
      host-only-interface;
      group-limit 50;
    }
    interface ge-0/0/3.0 {
      static {
        group 233.252.0.100;
      }
    }
    interface ge-0/0/2.0 {
      multicast-router-interface;
    }
  }
```

Verifying IGMP Snooping Operation

To verify that IGMP snooping is operating as configured, perform the following task:

- Displaying IGMP Snooping Information for VLAN v1 on page 118

Displaying IGMP Snooping Information for VLAN v1

Purpose Verify that IGMP snooping is enabled on vlanv1 and that ge-0/0/3 is recognized as a multicast-router interface.

Action From operational mode, enter the **show igmp snooping membership** command.

```
user@host> show igmp snooping membership
Instance: default-switch

Vlan: v1

Learning-Domain: default
Interface: ge-0/0/3.0, Groups: 1
Group: 233.252.0.100
Group mode: Exclude
Source: 0.0.0.0
Last reported by: Local
Group timeout: 0 Type: Static
```

Meaning By showing information for vlanv1, the command output confirms that IGMP snooping is configured on the VLAN. Interface ge-0/0/3.0 is listed as a multicast-router interface, as configured. Because none of the host interfaces are listed, none of the hosts are currently receivers for the multicast group.

Related Documentation

- IGMP Snooping Overview on page 81
- igmp-snooping on page 1077

Configuring Point-to-Multipoint LSP with IGMP Snooping

By default, IGMP snooping in VPLS uses multiple parallel streams when forwarding multicast traffic to PE routers participating in the VPLS. However, you can enable point-to-multipoint LSP for IGMP snooping to have multicast data traffic in the core take the point-to-multipoint path rather than using a pseudowire path. The effect is a reduction in the amount of traffic generated on the PE router when sending multicast packets for multiple VPLS sessions.

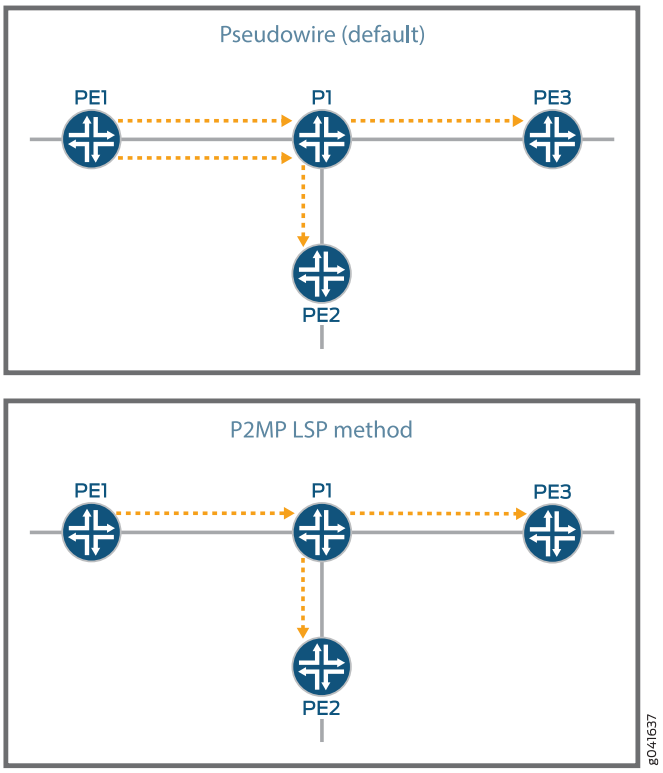
Figure 1 shows the effect on multicast traffic generated on the PE1 router (the device where the setting is enabled). When pseudowire LSP is used, the PE1 router sends multiple packets whereas with point-to-multipoint LSP enabled, only a single copy of the packets on the PE1 router is sent.

The options configured for IGMP snooping are applied on a per routing-instance, so all IGMP snooping routes in the same instance will use the same mode, point-to-multipoint or pseudowire.

NOTE: The point-to-multipoint option is available on MX960, MX480, MX240, and MX80 routers running Junos OS 13.3 and later.

NOTE: IGMP snooping is not supported on the core-facing pseudowire interfaces; all PE routers participating in VPLS will continue to receive multicast data traffic even when this option is enabled.

Figure 13: Point-to-multipoint LSP generates less traffic on the PE router than pseudowire.



In a VPLS instance with IGMP-snooping that uses a point-to-multipoint LSP, mcsnoopd (the multicast snooping process that allows Layer 3 inspection from Layer 2 device) will start listening for point-to-multipoint next-hop notifications and then manage the IGMP snooping routes accordingly. Enabling the **use-p2mp-lsp** command in Junos allows the IGMP snooping routes to start using this next-hop. In short, if point-to-multipoint is configured for a VPLS instance, multicast data traffic in the core can avoid ingress replication by taking the point-to-multipoint path. If the point-to-multipoint next-hop is unavailable, packets are handled in the VPLS instance in the same way as broadcast packets or unknown unicast frames. Note that IGMP snooping is not supported on the core-facing pseudowire interfaces. PE routers participating in VPLS will continue to receive multicast data traffic regardless of how Point-to-Multipoint is set.

To enable point-to-multipoint LSP, type the following CLI command:

```
[edit]
user@host> set routing-instances instance name instance-type vpls
igmp-snooping-options use-p2mp-lsp
```

The following output shows the hierarchical presence of igmp-snooping-options:

```
routing-instances {
    <instance-name> {
         instance-type vpls;
         igmp-snooping-options {
             use-p2mp-lsp;
         }
     }
 }
```

To show the operational status of point-to-multipoint LSP for IGMP snooping routes, use the following CLI command:

```
user@host> show igmp snooping options

Instance: master
     P2MP LSP in use: no
Instance: default-switch
     P2MP LSP in use: no
Instance: name
     P2MP LSP in use: yes
```

Related Documentation

- use-p2mp-lsp on page 1366
- show igmp snooping options on page 1488
- multicast-snooping-options on page 1174

CHAPTER 4

Configuring MLD Snooping

Understanding MLD Snooping

Multicast Listener Discovery (MLD) snooping constrains the flooding of IPv6 multicast traffic on VLANs. When MLD snooping is enabled on a VLAN, a Juniper Networks device examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving traffic for a multicast group. On the basis of what it learns, the device then forwards multicast traffic only to those interfaces in the VLAN that are connected to interested receivers instead of flooding the traffic to all interfaces.

MLD snooping supports MLD version 1 (MLDv1) and MLDv2. For details on MLDv1 and MLDv2, see the following standards:

- MLDv1—See RFC 2710, *Multicast Listener Discovery (MLD) for IPv6.*
- MLDv2—See RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6.*

This topic covers:

Benefits of MLD Snooping

- **Optimized bandwidth utilization**—The main benefit of MLD snooping is to reduce flooding of packets. IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN.
- **Improved security**—Denial of service attacks from unknown sources are prevented.

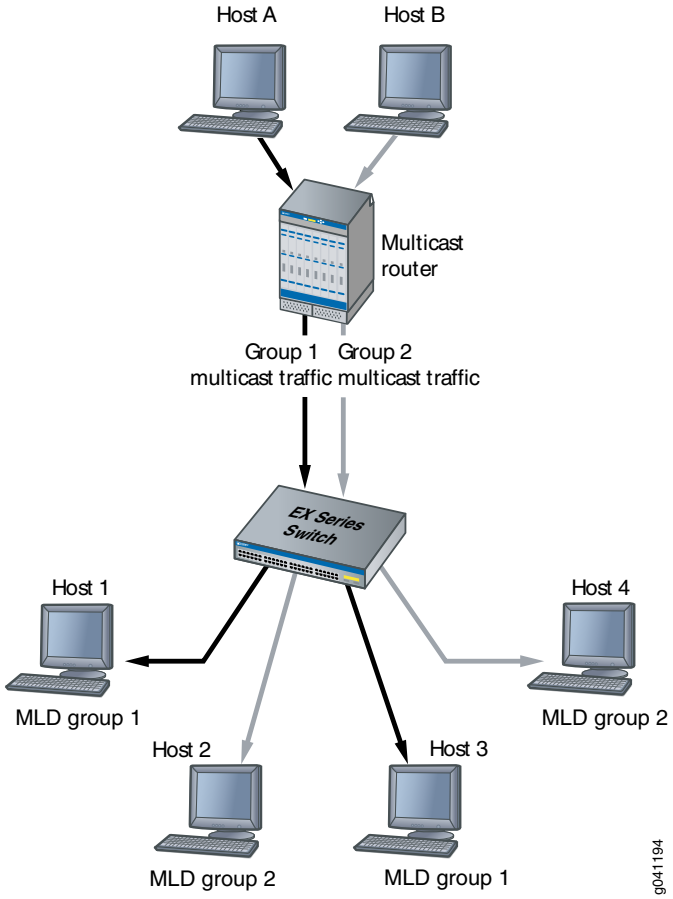
How MLD Snooping Works

By default, the device floods Layer 2 multicast traffic on all of the interfaces belonging to that VLAN on the device, except for the interface that is the source of the multicast traffic. This behavior can consume significant amounts of bandwidth.

You can enable MLD snooping to avoid this flooding. When you enable MLD snooping, the device monitors MLD messages between receivers (hosts) and multicast routers and uses the content of the messages to build an IPv6 multicast forwarding table—a database of IPv6 multicast groups and the interfaces that are connected to the interested members of each group. When the device receives multicast traffic for a multicast group, it uses the forwarding table to forward the traffic only to interfaces that are connected to receivers that belong to the multicast group.

Figure 14 on page 127 shows an example of multicast traffic flow with MLD snooping enabled.

Figure 14: Multicast Traffic Flow with MLD Snooping Enabled



MLD Message Types

Multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. In any given subnet, one multicast router is elected to act as an MLD querier. The MLD querier sends out the following types of queries to hosts:

- General query—Asks whether any host is listening to any group.
- Group-specific query—Asks whether any host is listening to a specific multicast group. This query is sent in response to a host leaving the multicast group and allows the router to quickly determine if any remaining hosts are interested in the group.
- Group-and-source-specific query—(MLD version 2 only) Asks whether any host is listening to group multicast traffic from a specific multicast source. This query is sent in response to a host indicating that it is no longer interested in receiving group multicast traffic from the multicast source and allows the router to quickly determine any remaining hosts are interested in receiving group multicast traffic from that source.

Hosts that are multicast listeners send the following kinds of messages:

- Membership report—Indicates that the host wants to join a particular multicast group.

- Leave report—Indicates that the host wants to leave a particular multicast group.

Only MLDv1 hosts use two different kinds of reports to indicate whether they want to join or leave a group. MLDv2 hosts send only one kind of report, the contents of which indicate whether they want to join or leave a group. However, for simplicity's sake, the MLD snooping documentation uses the term *membership report* for a report that indicates that a host wants to join a group and uses the term *leave report* for a report that indicates a host wants to leave a group.

How Hosts Join and Leave Multicast Groups

Hosts can join multicast groups in either of two ways:

- By sending an unsolicited membership report that specifies the multicast group that the host is attempting to join.
- By sending a membership report in response to a query from a multicast router.

A multicast router continues to forward multicast traffic to an interface provided that at least one host on that interface responds to the periodic general queries indicating its membership. For a host to remain a member of a multicast group, therefore, it must continue to respond to the periodic general queries.

Hosts can leave multicast groups in either of two ways:

- By not responding to periodic queries within a set interval of time. This results in what is known as a "silent leave."
- By sending a leave report.

NOTE: If a host is connected to the device through a hub, the host does not automatically leave the multicast group if it disconnects from the hub. The host remains a member of the group until group membership times out and a silent leave occurs. If another host connects to the hub port before the silent leave occurs, the new host might receive the group multicast traffic until the silent leave, even though it never sent an membership report.

Support for MLDv2 Multicast Sources

In MLDv2, a host can send a membership report that includes a list of source addresses. When the host sends a membership report in INCLUDE mode, the host is interested in group multicast traffic only from those sources in the source address list. If host sends a membership report in EXCLUDE mode, the host is interested in group multicast traffic from any source *except* the sources in the source address list. A host can also send an EXCLUDE report in which the source-list parameter is empty, which is known as an EXCLUDE NULL report. An EXCLUDE NULL report indicates that the host wants to join the multicast group and receive packets from all sources.

Devices that support MLD snooping support MLDv2 membership reports that are in INCLUDE and EXCLUDE mode. However, SRX Series devices, QFX Series switches, and EX Series switches running MLD snooping, except for EX9200 switches, do not support

forwarding on a per-source basis. Instead, the device consolidates all INCLUDE and EXCLUDE mode reports it receives on a VLAN for a specified group into a single route that includes all multicast sources for that group, with the next hop being all interfaces that have interested receivers for the group. As a result, interested receivers on the VLAN can receive traffic from a source that they did not include in their INCLUDE report or from a source they excluded in their EXCLUDE report. For example, if Host 1 wants traffic for group G from Source A and Host 2 wants traffic for group G from Source B, they both receive traffic for group G regardless of whether A or B sends the traffic.

MLD Snooping and Forwarding Interfaces

To determine how to forward multicast traffic, the device with MLD snooping enabled maintains information about the following interfaces in its multicast forwarding table:

- Multicast-router interfaces—These interfaces lead toward multicast routers or MLD queriers.
- Group-member interfaces—These interfaces lead toward hosts that are members of multicast groups.

The device learns about these interfaces by monitoring MLD traffic. If an interface receives MLD queries, the device adds the interface to its multicast forwarding table as a multicast-router interface. If an interface receives membership reports for a multicast group, the device adds the interface to its multicast forwarding table as a group-member interface.

Table entries for interfaces that the device learns about are subject to aging. For example, if a learned multicast-router interface does not receive MLD queries within a certain interval, the device removes the entry for that interface from its multicast forwarding table.

NOTE: **For the device to learn multicast-router interfaces and group-member interfaces, an MLD querier must exist in the network. For the device itself to function as an MLD querier, MLD must be enabled on the device.**

You can statically configure an interface to be a multicast-router interface or a group-member interface. The device adds a static interface to its multicast forwarding table without having to learn about the interface, and the entry in the table is not subject to aging. You can have a mix of statically configured and dynamically learned interfaces on the device.

General Forwarding Rules

Multicast traffic received on the device interface in a VLAN on which MLD snooping is enabled is forwarded according to the following rules.

MLD protocol traffic is forwarded as follows:

- MLD general queries received on a multicast-router interface are forwarded to all other interfaces in the VLAN.

- MLD group-specific queries received on a multicast-router interface are forwarded to only those interfaces in the VLAN that are members of the group.
- MLD reports received on a host interface are forwarded to multicast-router interfaces in the same VLAN, but not to the other host interfaces in the VLAN.

Multicast traffic that is not MLD protocol traffic is forwarded as follows:

- An unregistered multicast packet—that is, a packet for a group that has no current members—is forwarded to all multicast-router interfaces in the VLAN.
- A registered multicast packet is forwarded only to those host interfaces in the VLAN that are members of the multicast group and to all multicast-router interfaces in the VLAN.

NOTE: When IGMP and MLD snooping are both enabled on the same VLAN, multicast-router interfaces are created as part of IGMP and MLD snooping configuration. Unregistered multicast traffic is not blocked and can be passed through router interfaces, so due to hardware limitations, unregistered IPv4 multicast traffic might be passed through the multicast router interfaces created as part of MLD snooping configuration, and unregistered IPv6 multicast traffic might pass through multicast-router interfaces created as part of IGMP snooping configuration.

Examples of MLD Snooping Multicast Forwarding

The following examples are provided to illustrate how MLD snooping forwards multicast traffic in different topologies:

- Scenario 1: Device Forwarding Multicast Traffic to a Multicast Router and Hosts on page 130
- Scenario 2: Device Forwarding Multicast Traffic to Another Device on page 131
- Scenario 3: Device Connected to Hosts Only (No MLD Querier) on page 132
- Scenario 4: Layer 2/Layer 3 Device Forwarding Multicast Traffic Between VLANs on page 133

Scenario 1: Device Forwarding Multicast Traffic to a Multicast Router and Hosts

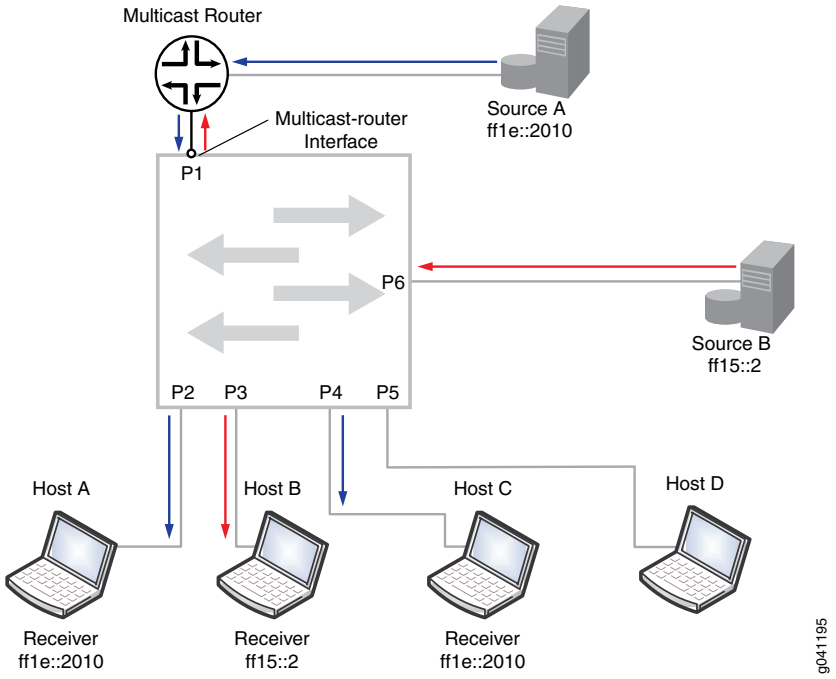
In the topology shown in Figure 15 on page 131, the device acting as a Layer 2 device receives multicast traffic belonging to multicast group **ff1e::2010** from Source A, which is connected to the multicast router. It also receives multicast traffic belonging to multicast group **ff15::2** from Source B, which is connected directly to the device. All interfaces on the device belong to the same VLAN.

Because the device receives MLD queries from the multicast router on interface P1, MLD snooping learns that interface P1 is a multicast-router interface and adds the interface to its multicast forwarding table. It forwards any MLD general queries it receives on this interface to all host interfaces on the device, and, in turn, forwards membership reports it receives from hosts to the multicast-router interface.

In the example, Hosts A and C have responded to the general queries with membership reports for group **ff1e::2010**. MLD snooping adds interfaces P2 and P4 to its multicast forwarding table as member interfaces for group **ff1e::2010**. It forwards the group multicast traffic received from Source A to Hosts A and C, but not to Hosts B and D.

Host B has responded to the general queries with a membership report for group **ff15::2**. The device adds interface P3 to its multicast forwarding table as a member interface for group **ff15::2** and forwards multicast traffic it receives from Source B to Host B. The device also forwards the multicast traffic it receives from Source B to the multicast-router interface P1.

Figure 15: Scenario 1: Device Forwarding Multicast Traffic to a Multicast Router and Hosts

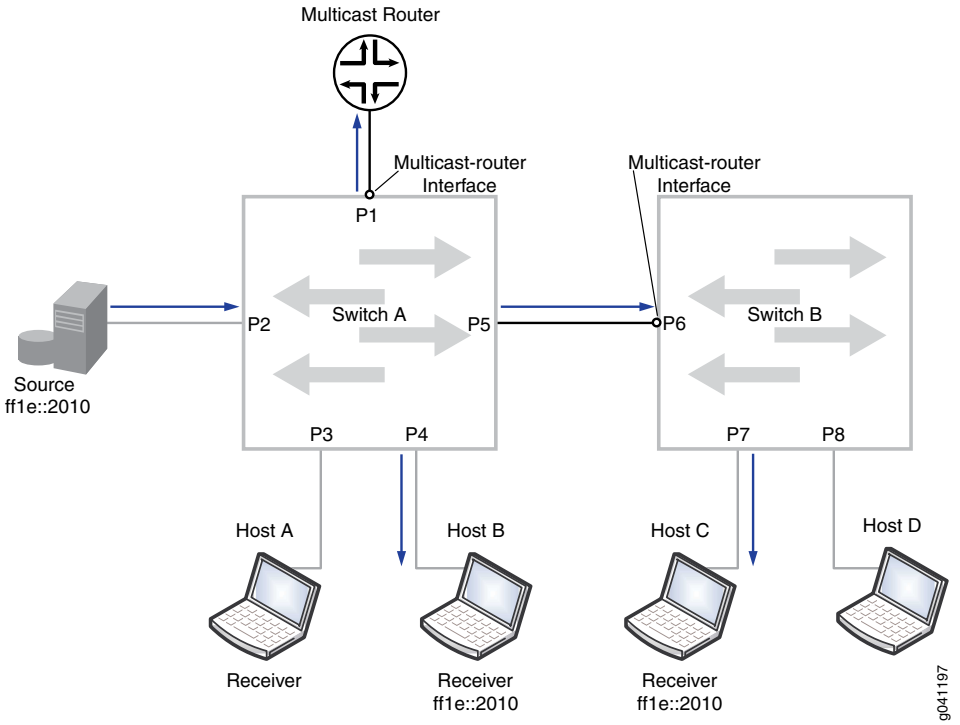


Scenario 2: Device Forwarding Multicast Traffic to Another Device

In the topology show in Figure 16 on page 132, a multicast source is connected to Device A. Device A in turn is connected to another device, Device B. Hosts on both Device A and B are potential members of the multicast group. Both devices are acting as Layer 2 devices, and all interfaces on the devices are members of the same VLAN.

Device A receives MLD queries from the multicast router on interface P1, making interface P1 a multicast-router interface for Device A. Device A forwards all general queries it receives on this interface to the other interfaces on the device, including the interface connecting Device B. Because Device B receives the forwarded MLD queries on interface P6, P6 is the multicast-router interface for Device B. Device B forwards the membership report it receives from Host C to Device A through its multicast-router interface. Device A forwards the membership report to its multicast-router interface, includes interface P5 in its multicast forwarding table as a group-member interface, and forwards multicast traffic from the source to Device B.

Figure 16: Scenario 2: Device Forwarding Multicast Traffic to Another Device



In certain implementations, you might have to configure P6 on Device B as a static multicast-router interface to avoid a delay in a host receiving multicast traffic. For example, if Device B receives unsolicited membership reports from its hosts before it learns which interface is its multicast-router interface, it does not forward those reports to Device A. If Device A then receives multicast traffic, it does not forward the traffic to Device B, because it has not received any membership reports on interface P5. This issue will resolve when the multicast router sends out its next general query; however, it can cause a delay in the host receiving multicast traffic. You can statically configure interface P6 as a multicast-router interface to solve this issue.

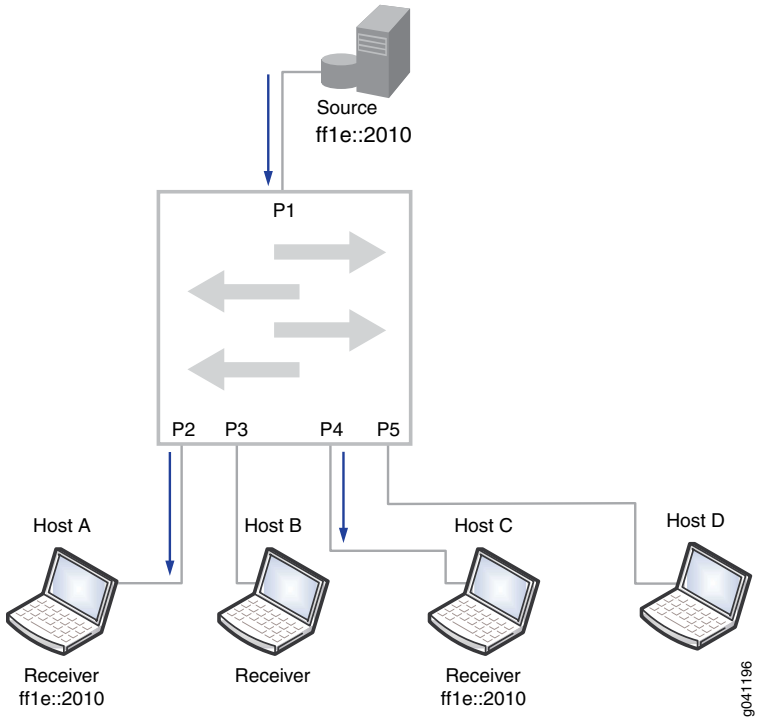
Scenario 3: Device Connected to Hosts Only (No MLD Querier)

In the topology shown in Figure 17 on page 133, the device is connected to a multicast source and to hosts. There is no multicast router in this topology—hence there is no MLD querier. Without an MLD querier to respond to, a host does not send periodic membership reports. As a result, even if the host sends an unsolicited membership report to join a multicast group, its membership in the multicast group will time out.

For MLD snooping to work correctly in this network so that the device forwards multicast traffic to Hosts A and C only, you can either:

- Configure interfaces P2 and P4 as static group-member interfaces.
- Configure a routed VLAN interface (RVI), also referred to as an integrated routing and bridging (IRB) interface, on the VLAN and enable MLD on it. In this case, the device itself acts as an MLD querier, and the hosts can dynamically join the multicast group and refresh their group membership by responding to the queries.

Figure 17: Scenario 3: Device Connected to Hosts Only (No MLD Querier)

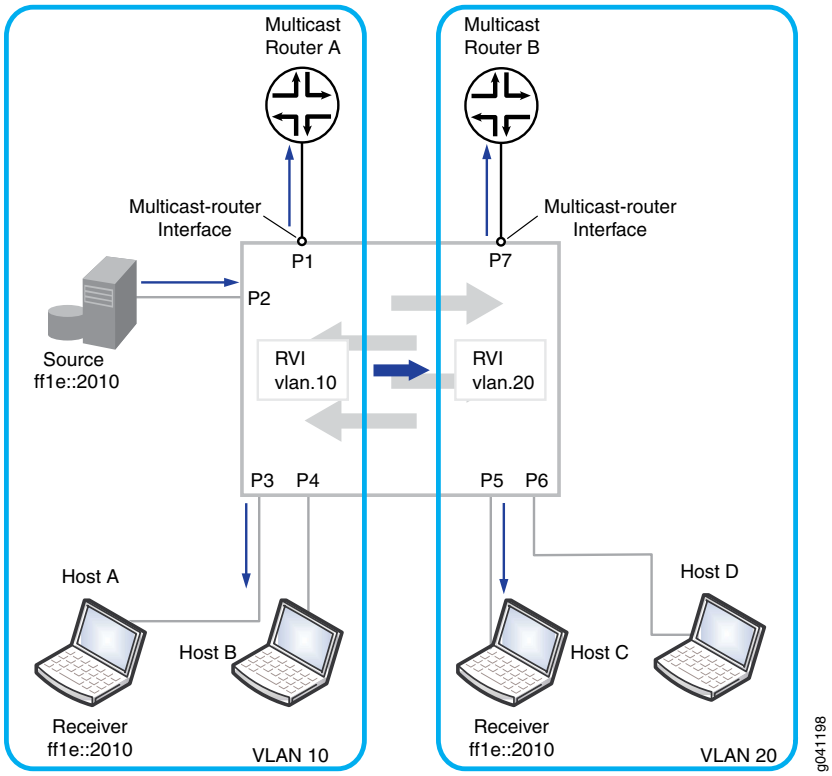


Scenario 4: Layer 2/Layer 3 Device Forwarding Multicast Traffic Between VLANs

In the topology shown in Figure 18 on page 134, a multicast source, Multicast Router A, and Hosts A and B are connected to the device and are in VLAN 10. Multicast Router B and Hosts C and D are also connected to the device and are in VLAN 20.

In a pure Layer 2 environment, traffic is not forwarded between VLANs. For Host C to receive the multicast traffic from the source on VLAN 10, RVIs (or IRB interfaces) must be created on VLAN 10 and VLAN 20 to permit routing of the multicast traffic between the VLANs.

Figure 18: Scenario 4: Layer 2/Layer 3 device Forwarding Multicast Traffic Between VLANs



Related Documentation

- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure) on page 142
- Example: Configuring MLD Snooping on Switches with ELS Support on page 163
- Verifying MLD Snooping on Switches on page 170
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Example: Configuring MLD Snooping on EX Series Switches on page 148
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167

Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure)

You can enable MLD snooping on a VLAN to constrain the flooding of IPv6 multicast traffic on a VLAN. When MLD snooping is enabled, a switch examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the switch then forwards IPv6 multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

MLD snooping is not enabled on the switch by default. To enable MLD snooping on all VLANs:

```
[edit]
user@switch# set protocols mld-snooping vlan all
```

For many networks, MLD snooping requires no further configuration.

You can perform the following optional configurations per VLAN:

- Selectively enable MLD snooping on specific VLANs.

NOTE: You cannot configure MLD snooping on a secondary VLAN.

- Specify the MLD version for the general query that the switch sends on an interface when the interface comes up.
- Enable immediate leave on a VLAN or all VLANs. Immediate leave reduces the length of time it takes the switch to stop forwarding multicast traffic when the last member host on the interface leaves the group.
- Configure an interface as a static multicast-router interface for a VLAN or for all VLANs so that the switch does not need to dynamically learn that the interface is a multicast-router interface.
- Configure an interface as a static member of a multicast group so that the switch does not need to dynamically learn the interface's membership.
- Change the value for certain timers and counters to match the values configured on the multicast router serving as the MLD querier.

TIP: When you configure MLD snooping using the vlan all statement, any VLAN that is not individually configured for MLD snooping inherits the vlan all configuration. Any VLAN that is individually configured for MLD snooping, on the other hand, inherits none of its configuration from vlan all. Any parameters that are not explicitly defined for the individual VLAN assume their default values, not the values specified in the vlan all configuration. For example, in the following configuration:

```
protocols {
    mld-snooping {
        vlan all {
            robust-count 8;
        }
        vlan employee {
            interface ge-0/0/8.0 {
                static {
                    group ff1e::1;
                }
            }
        }
    }
}
```

all VLANs, except employee, have a robust count of 8. Because employee has been individually configured, its robust count value is not determined by the value set under vlan all. Instead, its robust count is the default value of 2.

This topic covers:

Enabling or Disabling MLD Snooping on VLANs

MLD snooping is not enabled on any VLAN by default. You must explicitly configure a VLAN or all VLANs for MLD snooping.

This topic describes how you can enable or disable MLD snooping on specific VLANs or on all VLANs on the switch.

- To enable MLD snooping on all VLANs:

```
[edit protocols mld-snooping]
user@switch# set vlan all
```

- To enable MLD snooping on a specific VLAN:

```
[edit protocols mld-snooping]
user@switch# set vlan vlan-name
```

NOTE: You cannot configure MLD snooping on a secondary VLAN.

For example, to enable MLD snooping on VLAN **education**:

```
[edit protocols mld-snooping]
 user@switch# set vlan education
```

- To enable MLD snooping on all VLANs except a few VLANs:

1. Enable MLD snooping on all VLANs:

```
[edit protocols mld-snooping]
user@switch# set vlan all
```

2. Disable MLD snooping on individual VLANs:

```
[edit protocols mld-snooping]
user@switch# set vlan vlan-name disable
```

For example, to enable MLD snooping on all VLANs except **vlan100** and **vlan200**:

```
[edit protocols mld-snooping]
user@switch# set vlan all

[edit protocols mld-snooping]
user@switch# set vlan vlan100 disable

[edit protocols mld-snooping]
user@switch# set vlan vlan200 disable
```

You can also deactivate the MLD snooping protocol on the switch without changing the MLD snooping VLAN configurations:

```
[edit]
user@switch# deactivate protocols mld-snooping
```

Configuring the MLD Version

You can configure the version of MLD queries sent by a switch when MLD snooping is enabled. By default, the switch uses MLD version 1 (MLDv1). If you are using Protocol-Independent Multicast source-specific multicast (PIM-SSM), we recommend that you configure the switch to use MLDv2.

Typically, a switch passively monitors MLD messages sent between multicast routers and hosts and does not send MLD queries. The exception is when a switch detects that an interface has come up. When an interface comes up, the switch sends an immediate general membership query to all hosts on the interface. By doing so, the switch enables the multicast routers to learn group memberships more quickly than they would if they had to wait until the MLD querier sent its next general query.

The MLD version of the general query determines the MLD version of the host membership reports as follows:

- MLD version 1 (MLDv1) general query—Both MLDv1 and MLDv2 hosts respond with an MLDv1 membership report.
- MLDv2 general query—MLDv2 hosts respond with an MLDv2 membership report, while MLDv1 hosts are unable to respond to the query.

By default, the switch sends MLDv1 queries. This ensures compatibility with hosts and multicast routers that support MLDv1 only and cannot process MLDv2 reports. However, if your VLAN contains MLDv2 multicast routers and hosts and the routers are running PIM-SSM, we recommend that you configure MLD snooping for MLDv2. Doing so enables the routers to quickly learn which multicast sources the hosts on the interface want to receive traffic from.

NOTE: Configuring the MLD version does not limit the version of MLD messages that the switch can snoop. A switch can snoop both MLDv1 and MLDv2 messages regardless of the MLD version configured.

To configure the MLD version on a switch:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name version number
```

For example, to set the MLD version to version 2 for VLAN **marketing**:

```
[edit protocols]
user@switch# set mld-snooping vlan marketing version 2
```

Enabling Immediate Leave

By default, when a switch with MLD snooping enabled receives an MLD leave report on a member interface, it waits for hosts on the interface to respond to MLD group-specific queries to determine whether there still are hosts on the interface interested in receiving the group multicast traffic. If the switch does not see any membership reports for the group within a set interval of time, it removes the interface's group membership from the multicast forwarding table and stops forwarding multicast traffic for the group to the interface.

You can decrease the leave latency created by this default behavior by enabling immediate leave on a VLAN.

When you enable immediate leave on a VLAN, host tracking is also enabled, allowing the switch to keep track of the hosts on a interface that have joined a multicast group. When the switch receives a leave report from the last member of the group, it immediately stops forwarding traffic to the interface and does not wait for the interface group membership to time out.

Immediate leave is supported for both MLD version 1 (MLDv1) and MLDv2. However, with MLDv1, we recommend that you configure immediate leave only when there is only one MLD host on an interface. In MLDv1, only one host on a interface sends a membership report in response to a group-specifc query—any other interested hosts suppress their reports. This report-suppression feature means that the switch only knows about one interested host at any given time.

To enable immediate leave on a VLAN:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name immediate-leave
```

To enable immediate leave on all VLANs:

```
[edit protocols]
user@switch# set mld-snooping vlan all immediate-leave
```

Configuring an Interface as a Multicast-Router Interface

When MLD snooping is enabled on a switch, the switch determines which interfaces face a multicast router by monitoring interfaces for MLD queries or Protocol Independent Multicast (PIM) updates. If the switch receives these messages on an interface, it adds the interface to its multicast forwarding table as a multicast-router interface.

In addition to dynamically learned interfaces, the multicast forwarding table can include interfaces that you explicitly configure to be multicast router interfaces. Unlike the table entries for dynamically learned interfaces, table entries for statically configured interfaces are not subject to aging and deletion from the forwarding table.

Examples of when you might want to configure a static multicast-router interface include:

- You have an unusual network configuration that prevents MLD snooping from reliably learning about a multicast-router interface through monitoring MLD queries or PIM updates.
- Your implementation does not require an MLD querier.
- You have a stable topology and want to avoid the delay the dynamic learning process entails.

NOTE: If the interface you are configuring as a multicast-router interface is a trunk port, the interface becomes a multicast-router interface for all VLANs configured on the trunk port even if you have not explicitly configured it for all the VLANs. In addition, all unregistered multicast packets, whether they are IPv4 or IPv6 packets, are forwarded to the multicast-router interface, even if the interface is configured as a multicast-router interface only for MLD snooping.

To configure an interface as a static multicast-router interface:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name interface interface-name
multicast-router-interface
```

For example, to configure **ge-0/0/5.0** as a multicast-router interface for all VLANs on the switch:

```
[edit protocols]
user@switch# set mld-snooping vlan all interface ge-0/0/5.0
multicast-router-interface
```

Configuring Static Group Membership on an Interface

To determine how to forward multicast packets, a switch with MLD snooping enabled maintains a multicast forwarding table containing a list of host interfaces that have interested listeners for a specific multicast group. The switch learns which host interfaces to add or delete from this table by examining MLD membership reports as they arrive on interfaces on which MLD snooping is enabled.

In addition to such dynamically learned interfaces, the multicast forwarding table can include interfaces that you statically configure to be members of multicast groups. When you configure a static group interface, the switch adds the interface to the forwarding table as a host interface for the group. Unlike an entry for a dynamically learned interface, a static interface entry is not subject to aging and deletion from the forwarding table.

Examples of when you might want to configure static group membership on an interface include:

- You want to simulate an attached multicast receiver for testing purposes.
- The interface has receivers that cannot send MLD membership reports.

- You want the multicast traffic for a specific group to be immediately available to a receiver without any delay imposed by the dynamic join process.

You cannot configure multicast source addresses for a static group interface. The MLD version of a static group interface is always MLD version 1.

NOTE: The switch does not simulate MLD membership reports on behalf of a statically configured interface. Thus a multicast router might be unaware that the switch has an interface that is a member of the multicast group. You can configure a static group interface on the router to ensure that the switch receives the group multicast traffic.

To configure a host interface as a static member of a multicast group:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name interface  interface-name static
group ip-address
```

For example, to configure interface **ge-0/0/11.0** in VLAN **ip-camera-vlan** as a static member of multicast group **ff1e::1**:

```
[edit protocols]
user@switch# set mld-snooping vlan ip-camera-vlan interface ge-0/0/11.0 static
 group ff1e::1
```

Changing the Timer and Counter Values

MLD uses various timers and counters to determine how often an MLD querier sends out membership queries and when group memberships time out. On Juniper Networks EX Series switches, the MLD and MLD snooping timers and counters default values are set to the values recommended in RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*. These values work well for most multicast implementations.

There might be cases, however, where you might want to adjust the timer and counter values—for example, to reduce burstiness, to reduce leave latency, or to adjust for expected packet loss on a subnet. If you change a timer or counter value for the MLD querier on a VLAN, we recommend that you change the value for all multicast routers and switches on the VLAN so that all devices time out group memberships at approximately the same time.

The following timers and counters are configurable on a switch:

- **query-interval**—The length of time the MLD querier waits between sending general queries (the default is 125 seconds). You can change this interval to tune the number of MLD messages on the subnet; larger values cause general queries to be sent less often.

  You cannot configure this value directly for MLD snooping. MLD snooping inherits the value from the MLD value configured on the switch, which is applied to all VLANs on the switch.

  To configure the MLD **query-interval**:

```
[edit protocols]
user@switch# set mld query-interval seconds
```

- **query-response-interval**—The maximum length of time the host can wait until it responds (the default is 10 seconds). You can change this interval to adjust the burst peaks of MLD messages on the subnet. Set a larger interval to make the traffic less bursty.

  You cannot configure this value directly for MLD snooping. MLD snooping inherits the value from the MLD value configured on the switch, which is applied to all VLANs on the switch.

  To configure the MLD **query-response-interval**:

  ```
  [edit protocols]
  user@switch# set mld query-response-interval seconds
  ```

- **query-last-member-interval**—The length of time the MLD querier waits between sending group-specific membership queries (the default is 1 second). The MLD querier sends a group-specific query after receiving a leave report from a host. You can decrease this interval to reduce the amount of time it takes for multicast traffic to stop forwarding after the last member leaves a group.

  You cannot configure this value directly for MLD snooping. MLD snooping inherits the value from the MLD value configured on the switch, which is applied to all VLANs on the switch.

  To configure the MLD **query-last-member-interval**:

  ```
  [edit protocols]
  user@switch# set mld query-last-member-interval seconds
  ```

- **robust-count**—The number of times the querier resends a general membership query or a group-specific membership query (the default is 2 times). You can increase this count to tune for higher expected packet loss.

  For MLD snooping, you can configure **robust-count** for a specific VLAN. If a VLAN does not have **robust-count** configured, the **robust-count** value is inherited from the value configured for MLD.

  To configure **robust-count** for MLD snooping on a VLAN:

  ```
  [edit protocols]
  user@switch# set mld-snooping vlan vlan-name robust-count number
  ```

The values configured for **query-interval**, **query-response-interval**, and **robust-count** determine the multicast listener interval—the length of time the switch waits for a group membership report after a general query before removing a multicast group from its multicast forwarding table. The switch calculates the multicast listener interval by multiplying **query-interval** by **robust-count** and then adding **query-response-interval**:

(**query-interval** x **robust-count**) + **query-response-interval** = multicast listener interval

For example, the multicast listener interval is 260 seconds when the default settings for **query-interval**, **query-response-interval**, and **robust-count** are used:

(125 x 2) + 10 = 260

You can display the time remaining in the multicast listener interval before a group times out by using the **show mld-snooping membership** command.

Related Documentation

- Examples: Configuring MLD on page 50

Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure)

NOTE: This task uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure)" on page 134. For ELS details, see *Using the Enhanced Layer 2 Software CLI*.

You can enable MLD snooping on a VLAN to constrain the flooding of IPv6 multicast traffic on the VLAN. When MLD snooping is enabled, a switch examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the switch then forwards IPv6 multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

You can perform the following configurations for each VLAN:

- Selectively enable MLD snooping on specific VLANs.
- Specify the MLD version for the general query that the switch sends on an interface when the interface comes up.
- Enable immediate leave to reduce the length of time it takes the switch to stop forwarding multicast traffic when the last member host on the interface leaves the group.
- Configure an interface as a static multicast-router interface so that the switch does not need to dynamically learn that the interface is a multicast-router interface.
- Configure an interface as a static member of a multicast group so that the switch does not need to dynamically learn the interface's membership.
- Change the value for certain timers and counters to match the values configured on the multicast router serving as the MLD querier.

This topic covers:

- Enabling or Disabling MLD Snooping on VLANs on page 143
- Configuring the MLD Version on page 143
- Enabling Immediate Leave on page 144
- Configuring an Interface as a Multicast-Router Interface on page 145
- Configuring Static Group Membership on an Interface on page 145
- Changing the Timer and Counter Values on page 146

Enabling or Disabling MLD Snooping on VLANs

MLD snooping is not enabled on any VLAN by default. You must explicitly enable MLD snooping on specific interfaces.

- To enable MLD snooping on a specific VLAN:

```
[edit protocols mld-snooping]
user@switch# set vlan vlan-name
```

NOTE: You cannot enable MLD snooping on a secondary VLAN.

For example, to enable MLD snooping on VLAN education:

```
[edit protocols mld-snooping]
 user@switch# set vlan education
```

- To disable MLD snooping on a specific VLAN:

```
[edit protocols mld-snooping]
user@switch# delete vlan vlan-name
```

You can also deactivate the MLD snooping protocol on the switch without changing the MLD snooping VLAN configurations:

```
[edit]
user@switch# deactivate protocols mld-snooping
```

Configuring the MLD Version

You can configure the version of MLD queries sent by a switch when MLD snooping is enabled. By default, the switch uses MLD version 1 (MLDv1). If you are using Protocol-Independent Multicast source-specific multicast (PIM-SSM), we recommend that you configure the switch to use MLDv2.

Typically, a switch passively monitors MLD messages sent between multicast routers and hosts and does not send MLD queries. The exception is when a switch detects that an interface has come up. When an interface comes up, the switch sends an immediate general membership query to all hosts on the interface. By doing so, the switch enables the multicast routers to learn group memberships more quickly than they would if they had to wait until the MLD querier sent its next general query.

The MLD version of the general query determines the MLD version of the host membership reports as follows:

- MLD version 1 (MLDv1) general query—Both MLDv1 and MLDv2 hosts respond with an MLDv1 membership report.
- MLDv2 general query—MLDv2 hosts respond with an MLDv2 membership report, while MLDv1 hosts are unable to respond to the query.

By default, the switch sends MLDv1 queries. This ensures compatibility with hosts and multicast routers that support MLDv1 only and cannot process MLDv2 reports. However, if your VLAN contains MLDv2 multicast routers and hosts and the routers are running

PIM-SSM, we recommend that you configure MLD snooping for MLDv2. Doing so enables the routers to quickly learn which multicast sources the hosts on the interface want to receive traffic from.

NOTE: Configuring the MLD version does not limit the version of MLD messages that the switch can snoop. A switch can snoop both MLDv1 and MLDv2 messages regardless of the MLD version configured.

To configure the MLD version on an interface:

```
[edit protocols]
user@switch# set mld interface interface-name version number
```

For example, to set the MLD version to version 2 on interface ge-0/0/2:

```
[edit protocols]
user@switch# set mld interface ge-0/0/2 version 2
```

Enabling Immediate Leave

By default, when a switch with MLD snooping enabled receives an MLD leave report on a member interface, it waits for hosts on the interface to respond to MLD group-specific queries to determine whether there still are hosts on the interface interested in receiving the group multicast traffic. If the switch does not see any membership reports for the group within a set interval of time, it removes the interface's group membership from the multicast forwarding table and stops forwarding multicast traffic for the group to the interface.

You can decrease the leave latency created by this default behavior by enabling immediate leave on a VLAN.

When you enable immediate leave on a VLAN, host tracking is also enabled, allowing the switch to keep track of the hosts on a interface that have joined a multicast group. When the switch receives a leave report from the last member of the group, it immediately stops forwarding traffic to the interface and does not wait for the interface group membership to time out.

Immediate leave is supported for both MLD version 1 (MLDv1) and MLDv2. However, with MLDv1, we recommend that you configure immediate leave only when there is only one MLD host on an interface. In MLDv1, only one host on a interface sends a membership report in response to a group-specifc query—any other interested hosts suppress their reports. This report-suppression feature means that the switch only knows about one interested host at any given time.

To enable immediate leave on a VLAN:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name immediate-leave
```

Configuring an Interface as a Multicast-Router Interface

When MLD snooping is enabled on a switch, the switch determines which interfaces face a multicast router by monitoring interfaces for MLD queries or Protocol Independent Multicast (PIM) updates. If the switch receives these messages on an interface, it adds the interface to its multicast forwarding table as a multicast-router interface.

In addition to dynamically learned interfaces, the multicast forwarding table can include interfaces that you explicitly configure to be multicast router interfaces. Unlike the table entries for dynamically learned interfaces, table entries for statically configured interfaces are not subject to aging and deletion from the forwarding table.

Examples of when you might want to configure a static multicast-router interface include:

- You have an unusual network configuration that prevents MLD snooping from reliably learning about a multicast-router interface through monitoring MLD queries or PIM updates.
- Your implementation does not require an MLD querier.
- You have a stable topology and want to avoid the delay the dynamic learning process entails.

To configure an interface as a static multicast-router interface:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name interface interface-name
multicast-router-interface
```

For example, to configure ge-0/0/5.0 as a multicast-router interface for VLAN employee:

```
[edit protocols]
user@switch# set mld-snooping vlan employee interface ge-0/0/5.0
multicast-router-interface
```

Configuring Static Group Membership on an Interface

To determine how to forward multicast packets, a switch with MLD snooping enabled maintains a multicast forwarding table containing a list of host interfaces that have interested listeners for a specific multicast group. The switch learns which host interfaces to add or delete from this table by examining MLD membership reports as they arrive on interfaces on which MLD snooping is enabled.

In addition to such dynamically learned interfaces, the multicast forwarding table can include interfaces that you statically configure to be members of multicast groups. When you configure a static group interface, the switch adds the interface to the forwarding table as a host interface for the group. Unlike an entry for a dynamically learned interface, a static interface entry is not subject to aging and deletion from the forwarding table.

Examples of when you might want to configure static group membership on an interface include:

- You want to simulate an attached multicast receiver for testing purposes.
- The interface has receivers that cannot send MLD membership reports.

- You want the multicast traffic for a specific group to be immediately available to a receiver without any delay imposed by the dynamic join process.

You cannot configure multicast source addresses for a static group interface. The MLD version of a static group interface is always MLD version 1.

NOTE: The switch does not simulate MLD membership reports on behalf of a statically configured interface. Thus a multicast router might be unaware that the switch has an interface that is a member of the multicast group. You can configure a static group interface on the router to ensure that the switch receives the group multicast traffic.

To configure a host interface as a static member of a multicast group:

```
[edit protocols]
user@switch# set mld-snooping vlan vlan-name interface  interface-name static
group ip-address
```

For example, to configure interface ge-0/0/11.0 in VLAN employee as a static member of multicast group ff1e::1:

```
[edit protocols]
user@switch# set mld-snooping vlan ip-camera-vlan interface ge-0/0/11.0 static
 group ff1e::1
```

Changing the Timer and Counter Values

MLD uses various timers and counters to determine how often an MLD querier sends out membership queries and when group memberships time out. On Juniper Networks switches, the MLD and MLD snooping timers and counters default values are set to the values recommended in RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*. These values work well for most IPv6 multicast deployments.

There might be cases, however, where you might want to adjust the timer and counter values—for example, to reduce burstiness, to reduce leave latency, or to adjust for expected packet loss on a subnet. If you change a timer or counter value for the MLD querier on a VLAN, we recommend that you change the value for all multicast routers and switches on the VLAN so that all devices time out group memberships at approximately the same time.

The following timers and counters are configurable on a switch:

- **query-interval**—The length of time in seconds the MLD querier waits between sending general queries (the default is 125 seconds). You can change this interval to tune the number of MLD messages on the subnet; larger values cause general queries to be sent less often.

  To configure the MLD query interval:

  ```
  [edit protocols]
  user@switch# set mld-snooping vlan vlan-name query-interval seconds
  ```

- **query-response-interval**—The maximum length of time in seconds the host waits before it responds (the default is 10 seconds). You can change this interval to accommodate the burst peaks of MLD messages on the subnet. Set a larger interval to make the traffic less bursty.

  To configure the MLD query response interval:

  ```
  [edit protocols]
  user@switch# set mld-snooping vlan  vlan-name query-response-interval seconds
  ```

- **query-last-member-interval**—The length of time the MLD querier waits between sending group-specific membership queries (the default is 1 second). The MLD querier sends a group-specific query after receiving a leave report from a host. You can decrease this interval to reduce the amount of time it takes for multicast traffic to stop forwarding after the last member leaves a group.

  To configure the MLD query last member interval:

  ```
  [edit protocols]
  user@switch# set mld-snooping vlan  vlan-name query-last-member-interval seconds
  ```

- **robust-count**—The number of times the querier resends a general membership query or a group-specific membership query (the default is 2 times). You can increase this count to tune for higher anticipated packet loss.

  For MLD snooping, you can configure **robust-count** for a specific VLAN. If a VLAN does not have **robust-count** configured, the value is inherited from the value configured for MLD.

  To configure **robust-count** for MLD snooping on a VLAN:

  ```
  [edit protocols]
  user@switch# set mld-snooping vlan vlan-name robust-count number
  ```

The values configured for **query-interval**, **query-response-interval**, and **robust-count** determine the multicast listener interval—the length of time the switch waits for a group membership report after a general query before removing a multicast group from its multicast forwarding table. The switch calculates the multicast listener interval by multiplying **query-interval** value by the **robust-count** value and then adding the **query-response-interval** to the product:

(**query-interval** x **robust-count**) + **query-response-interval** = multicast listener interval

For example, the multicast listener interval is 260 seconds when the default settings for **query-interval**, **query-response-interval**, and **robust-count** are used:

(125 x 2) + 10 = 260

To display the time remaining in the multicast listener interval before a group times out, use the **show mld-snooping membership** command.

Related Documentation

- Example: Configuring MLD Snooping on Switches with ELS Support on page 163
- Examples: Configuring MLD on page 50
- Verifying MLD Snooping on Switches on page 170

Example: Configuring MLD Snooping on EX Series Switches

You can enable MLD snooping on a VLAN to constrain the flooding of IPv6 multicast traffic on a VLAN. When MLD snooping is enabled, a switch examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the switch then forwards IPv6 multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

This example describes how to configure MLD snooping:

Requirements

This example uses the following software and hardware components:

- One EX Series switch
- Junos OS Release 12.1 or later

Before you configure MLD snooping, be sure you have:

- Configured the **vlan100** VLAN on the switch
- Assigned interfaces **ge-0/0/0**, **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/12** to **vlan100**
- Configured **ge-0/0/12** as a trunk interface.

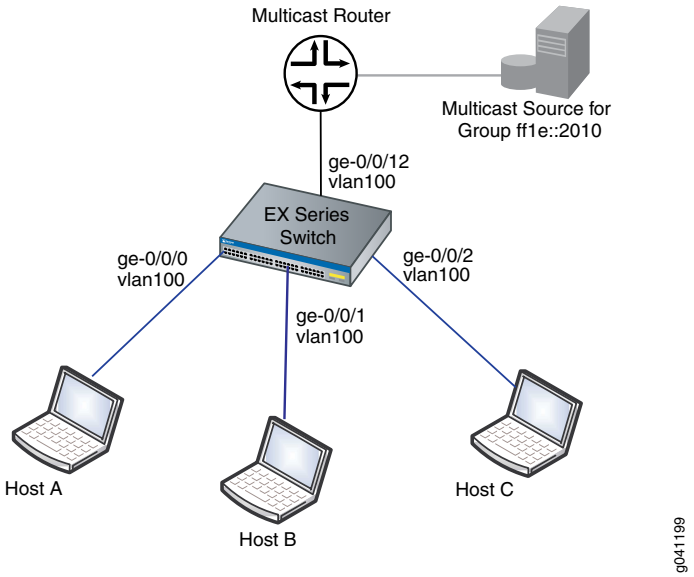
See *Configuring VLANs for EX Series Switches (CLI Procedure)*.

Overview and Topology

In this example, interfaces **ge-0/0/0**, **ge-0/0/1**, and **ge-0/0/2** on the switch are in **vlan100** and are connected to hosts that are potential multicast receivers. Interface **ge-0/0/12**, a trunk interface also in **vlan100**, is connected to a multicast router. The router acts as the MLD querier and forwards multicast traffic for group **ff1e::2010** to the switch from a multicast source.

The example topology is illustrated in Figure 19 on page 149.

Figure 19: Example MLD Snooping Topology



In this example topology, the multicast router forwards multicast traffic to the switch from the source when it receives a memberhsip report for group **ff1e::2010** from one of the hosts—for example, Host B. If MLD snooping is not enabled on **vlan100**, the switch floods the multicast traffic on all interfaces in **vlan100** (except for interface **ge-0/0/12**). If MLD snooping is enabled on **vlan100**, the switch monitors the MLD messages between the hosts and router, allowing it to determine that only Host B is interested in receiving the multicast traffic. The switch then forwards the multicast traffic only to interface **ge-0/0/1**.

This example shows how to enable MLD snooping on **vlan100**. It also shows how to perform the following optional configurations, which can reduce group join and leave latency:

- Configure immediate leave on the VLAN. When immediate leave is configured, the switch stops forwarding multicast traffic on an interface when it detects that the last member of the multicast group has left the group. If immediate leave is not configured, the switch waits until the group-specific membership queries time out before it stops forwarding traffic.
- Configure **ge-0/0/12** as a static multicast-router interface. In this topology, **ge-0/0/12** always leads to the multicast router. By statically configuring **ge-0/0/12** as a multicast-router interface, you avoid any delay imposed by the switch having to learn that **ge-0/0/12** is a multicast-router interface.

Configuration

To configure MLD snooping on a switch:

CLI Quick Configuration

To quickly configure MLD snooping, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols mld-snooping vlan vlan100
set protocols mld-snooping vlan vlan100 immediate-leave
set protocols mld-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
```

Step-by-Step Procedure

To configure MLD snooping:

1. Enable MLD snooping on VLAN **vlan100**:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100
   ```

2. Configure the switch to immediately remove a group membership from an interface when it receives a leave report from the last member of the group on the interface:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100 immediate-leave
   ```

3. Statically configure interface **ge-0/0/12** as a multicast-router interface:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
   ```

Results

Check the results of the configuration:

```
[edit protocols]
user@switch# show mld-snooping
vlan vlan100 {
    immediate-leave;
    interface ge-0/0/12.0 {
        multicast-router-interface;
    }
}
```

Verifying MLD Snooping Configuration

To verify that MLD snooping is enabled on the VLAN and the MLD snooping forwarding interfaces are correct, perform the following task:

- Verifying MLD Snooping Interface Membership on VLAN vlan100 on page 150

Verifying MLD Snooping Interface Membership on VLAN vlan100

Purpose

Verify that MLD snooping is enabled on **vlan100** and that the multicast-router interface is statically configured:

Action Show the group memberships maintained by MLD snooping for **vlan100**:

```
user@switch> show mld-snooping membership vlan vlan100 detail
VLAN: vlan100 Tag: 100 (Index: 8)
    Router interfaces:
        ge-0/0/12.0 static Uptime: 00:15:03
   Group: ff1e::2010
     ge-0/0/1.0 Timeout: 225 Flags: <V2-hosts>
     Last reporter: fe80::2020:1:1:3
```

Meaning MLD snooping is running on **vlan100**, and interface **ge-0/0/12.0** is a statically configured multicast-router interface. Because the multicast group **ff1e::2010** is listed, at least one host in the VLAN is a current member of the multicast group and that host is on interface **ge-0/0/1.0**.

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167
- Understanding MLD Snooping on page 125

Example: Configuring MLD Snooping on SRX Series Devices

You can enable MLD snooping on a VLAN to constrain the flooding of IPv6 multicast traffic on a VLAN. When MLD snooping is enabled, SRX Series device examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the device then forwards IPv6 multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

This example describes how to configure MLD snooping:

- Requirements on page 151
- Overview and Topology on page 152
- Configuration on page 153
- Verifying MLD Snooping Configuration on page 155

Requirements

This example uses the following software and hardware components:

- One SRX Series device
- Junos OS Release 18.1R1

Before you configure MLD snooping, be sure you have:

- Configured the **vlan100** VLAN on the device
- Assigned interfaces **ge-0/0/0**, **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/3** to **vlan100**

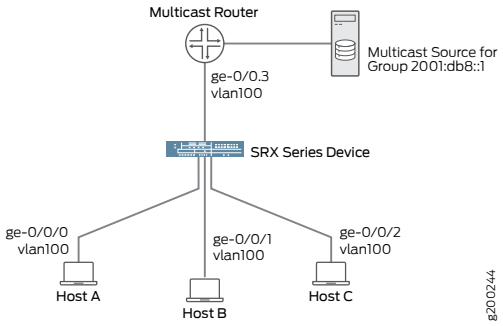
- Configured **ge-0/0/3** as a trunk interface.

Overview and Topology

In this example, interfaces **ge-0/0/0**, **ge-0/0/1**, and **ge-0/0/2** on the device are in **vlan100** and are connected to hosts that are potential multicast receivers. Interface **ge-0/0/3**, a trunk interface also in **vlan100**, is connected to a multicast router. The router acts as the MLD querier and forwards multicast traffic for group **2001:db8::1** to the device from a multicast source.

The example topology is illustrated in Figure 19 on page 149.

Figure 20: Example MLD Snooping Topology



In this example topology, the multicast router forwards multicast traffic to the device from the source when it receives a memberhsip report for group **2001:db8::1** from one of the hosts—for example, Host B. If MLD snooping is not enabled on **vlan100**, then the device floods the multicast traffic on all interfaces in **vlan100** (except for interface **ge-0/0/3**). If MLD snooping is enabled on **vlan100**, the device monitors the MLD messages between the hosts and router, allowing it to determine that only Host B is interested in receiving the multicast traffic. The device then forwards the multicast traffic only to interface **ge-0/0/1**.

This example shows how to enable MLD snooping on **vlan100**. It also shows how to perform the following optional configurations, which can reduce group join and leave latency:

- Configure immediate leave on the VLAN. When immediate leave is configured, the device stops forwarding multicast traffic on an interface when it detects that the last member of the multicast group has left the group. If immediate leave is not configured, the device waits until the group-specific membership queries time out before it stops forwarding traffic
- Configure **ge-0/0/3** as a static multicast-router interface. In this topology, **ge-0/0/3** always leads to the multicast router. By statically configuring **ge-0/0/3** as a multicast-router interface, you avoid any delay imposed by the device having to learn that **ge-0/0/3** is a multicast-router interface.

Configuration

To configure MLD snooping on a device:

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-0/0/0 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-0/0/1 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/1 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-0/0/2 unit 0 family ethernet-switching interface-mode access
set interfaces ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
set interfaces ge-0/0/3 unit 0 family ethernet-switching interface-mode trunk
set interfaces ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
set vlans vlan100 vlan-id 100
set routing-options nonstop-routing
set protocols mld-snooping vlan vlan100 query-interval 200
set protocols mld-snooping vlan vlan100 query-response-interval 0.4
set protocols mld-snooping vlan vlan100 query-last-member-interval 0.1
set protocols mld-snooping vlan vlan100 robust-count 4
set protocols mld-snooping vlan vlan100 immediate-leave
set protocols mld-snooping vlan vlan100 interface ge-0/0/1.0 host-only-interface
set protocols mld-snooping vlan vlan100 interface ge-0/0/0.0 group-limit 50
set protocols mld-snooping vlan vlan100 interface ge-0/0/2.0 static group 2001:db8::1
set protocols mld-snooping vlan vlan100 interface ge-0/0/3.0 multicast-router-interface
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see Using the CLI Editor in Configuration Mode in the CLI User Guide.

To configure MLD snooping:

1. Configure the access mode interfaces.

```
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/0 unit 0 family ethernet-switching vlan members vlan100
user@host# set ge-0/0/1 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/1 unit 0 family ethernet-switching vlan members vlan100
user@host# set ge-0/0/2 unit 0 family ethernet-switching interface-mode access
user@host# set ge-0/0/2 unit 0 family ethernet-switching vlan members vlan100
```

2. Configure the trunk mode interface.

```
[edit interfaces]
user@host# set ge-0/0/3 unit 0 family ethernet-switching interface-mode trunk
user@host# set ge-0/0/3 unit 0 family ethernet-switching vlan members vlan100
```

3. Configure the VLAN.

```
[edit vlans vlan100]
user@host# set vlans v100 vlan-id 100
```

4. Configure nonstop routing

```
[edit]
user@host# set routing-options nonstop-routing
```

5. Configure the limit for the number of multicast groups allowed on the ge-0/0/1.0 interface to 50.

```
[edit vlans vlan100]
user@host# set protocols mld-snooping vlan vlan100 interface ge-0/0/0.0
  group-limit 50
```

6. Configure the device to immediately remove a group membership from an interface when it receives a leave message from that interface without waiting for any other MLD messages to be exchanged.

```
[edit vlans vlan100]
user@host# set protocols mld-snooping vlan vlan100 immediate-leave
```

7. Statically configure interface ge-0/0/2.0 as a multicast-router interface.

```
[edit vlans vlan100]
user@host# set protocols mld-snooping vlan vlan100 interface ge-0/0/2.0 static
  group 2001:db8::1
```

8. Configure an interface to be an exclusively router-facing interface (to receive multicast traffic).

```
[edit vlans vlan100]
user@host# set protocols mld-snooping vlan vlan100 interface ge-0/0/3.0
  multicast-router-interface
```

9. Configure an interface to be an exclusively host-facing interface (to drop MLD query messages).

```
[edit vlans vlan100]
user@host# set protocols mld-snooping vlan vlan100 interface ge-0/0/1.0
  host-only-interface
```

10. Configure the IGMP message intervals and robustness count.

```
[edit vlans vlan100]
uer@host# set protocols mld-snooping vlan v100 query-interval 200
uer@host# set protocols mld-snooping vlan v100 query-response-interval 0.4
uer@host# set protocols mld-snooping vlan v100 query-last-member-interval 0.1
uer@host# set protocols mld-snooping vlan v1 robust-count 4
```

11. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results From configuration mode, confirm your configuration by entering the **show protocols mld-snooping** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
user@host# show protocols mld-snooping
  vlan vlan100 {
    query-interval 200;
    query-response-interval 0.4;
    query-last-member-interval 0.1;
    robust-count 4;
    immediate-leave;
    interface ge-0/0/1.0 {
      host-only-interface;
    }
    interface ge-0/0/0.0 {
      group-limit 50;
    }
    interface ge-0/0/2.0 {
      static {
        group 2001:db8::1;
      }
    }
    interface ge-0/0/3.0 {
      multicast-router-interface;
    }
  }
```

Verifying MLD Snooping Configuration

To verify that MLD snooping is enabled on the VLAN and the MLD snooping forwarding interfaces are correct, perform the following task:

- Verifying MLD Snooping Interface Membership on VLAN vlan100 on page 155

Verifying MLD Snooping Interface Membership on VLAN vlan100

Purpose Verify that MLD snooping is enabled on **vlan100** and that the multicast-router interface is statically configured:

Action From operational mode, enter the **show mld snooping membership** command.

```
user@host> show mld snooping membership
Instance: default-switch

Vlan: vlan100

Learning-Domain: default
Interface: ge-0/0/0.0, Groups: 0
Interface: ge-0/0/1.0, Groups: 0
Interface: ge-0/0/2.0, Groups: 1
    Group: 2001:db8::1
        Group mode: Exclude
        Source: ::
```

```
Last reported by: Local
Group timeout:      0 Type: Static
```

Meaning MLD snooping is running on **vlan100**, and interface **ge-0/0/3.0** is a statically configured multicast-router interface. Because the multicast group **2001:db8::1** is listed, at least one host in the VLAN is a current member of the multicast group and that host is on interface **ge-0/0/1.0**.

Related Documentation

- mld-snooping on page 1155
- Understanding MLD Snooping on page 125

Configuring MLD Snooping Tracing Operations on EX Series Switches (CLI Procedure)

By enabling tracing operations for MLD snooping, you can record detailed messages about the operation of the protocol, such as the various types of protocol packets sent and received. Table 8 on page 156 describes the tracing operations you can enable and the flags used to specify them in the tracing configuration.

Table 8: Supported Tracing Operations for MLD Snooping

Tracing Operation	Flag
Trace all (equivalent of including all flags).	**all**
Trace general MLD snooping protocol events.	**general**
Trace communication over routing socket events.	**krt**
Trace leave reports.	**leave**
Trace next-hop-related events.	**nexthop**
Trace normal MLD snooping protocol events. If you do not specify this flag, only unusual or abnormal operations are traced.	**normal**
Trace all MLD packets.	**packets**
Trace policy processing.	**policy**
Trace MLD membership query messages.	**query**
Trace membership reports	**report**
Trace routing information.	**route**
Trace state transitions.	**state**
Trace routing protocol task processing.	**task**

Table 8: Supported Tracing Operations for MLD Snooping (continued)

Tracing Operation	Flag
Trace timer processing.	**timer**
Trace VLAN-related events.	**vlan**

This topic covers:

- Configuring Tracing Operations on page 157
- Viewing, Stopping, and Restarting Tracing Operations on page 158

Configuring Tracing Operations

To configure tracing operations for MLD snooping:

1. Configure the filename for the trace file:

   ```
   [edit protocols mld-snooping ]
   user@switch# set traceoptions file filename
   ```

   For example:

   ```
   [edit protocols mld-snooping ]
   user@switch# set traceoptions file mld-snoop-trace
   ```

2. (Optional) Configure the maximum number of trace files and size of the trace files:

   ```
   [edit protocols mld-snooping ]
   user@switch # set file files number size size
   ```

   For example:

   ```
   [edit protocols mld-snooping ]
   user@switch # set traceoptions file files 5 size 1m
   ```

   causes the contents of the trace file to be emptied and archived in a .gz file when the file reaches 1 MB. Four archive files are maintained, the contents of which are rotated whenever the current active trace file is archived.

   If you omit this step, the maximum number of trace files defaults to 10, with the maximum file size defaulting to 128 K.

3. Specify one of the tracing flags shown in Table 8 on page 156:

   ```
   [edit protocols mld-snooping ]
   user@switch # set traceoptions flag flagname
   ```

   For example, to perform trace operations on VLAN-related events and MLD query messages:

   ```
   [edit protocols mld-snooping ]
   user@switch# set traceoptions flag vlan
   ```

   ```
   [edit protocols mld-snooping ]
   user@switch# set traceoptions flag query
   ```

Viewing, Stopping, and Restarting Tracing Operations

When you commit the configuration, tracing operations begin. You can view the trace file in the **/var/log** directory. For example:

```
user@switch> file show /var/log/mld-snoop-trace
```

You can stop and restart tracing operations by deactivating and reactivating the configuration:

```
[edit]
user@switch# deactivate protocols mld-snooping traceoptions

[edit]
user@switch# activate protocols mld-snooping traceoptions
```

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- *Tracing and Logging Junos OS Operations*

Configuring MLD Snooping Tracing Operations on EX Series Switch VLANs (CLI Procedure)

By enabling tracing operations for MLD snooping, you can record detailed messages about the operation of the protocol, such as the various types of protocol packets sent and received. Table 8 on page 156 describes the tracing operations you can enable and the flags used to specify them in the tracing configuration.

Table 9: Supported Tracing Operations for MLD Snooping

Tracing Operation	Flag
Trace all (equivalent of including all flags).	**all**
Trace client notifications.	**client-notification**
Trace general MLD snooping protocol events.	**general**
Trace group operations.	**group**
Trace host notifications.	**host-notification**
Trace leave reports.	**leave**
Trace normal MLD snooping protocol events. If you do not specify this flag, only unusual or abnormal operations are traced.	**normal**
Trace all MLD packets.	**packets**
Trace policy processing.	**policy**
Trace MLD membership query messages.	**query**

Table 9: Supported Tracing Operations for MLD Snooping (continued)

Tracing Operation	Flag
Trace membership reports.	**report**
Trace routing information.	**route**
Trace state transitions.	**state**
Trace routing protocol task processing.	**task**
Trace timer processing.	**timer**

This topic covers:

- Configuring Tracing Operations on page 159
- Viewing, Stopping, and Restarting Tracing Operations on page 160

Configuring Tracing Operations

To configure tracing operations for MLD snooping:

1. Configure the filename for the trace file:

   ```
   [edit protocols mld-snooping ]
   user@switch# set vlan vlan-name traceoptions file filename
   ```

   For example:

   ```
   [edit protocols mld-snooping ]
   user@switch# set vlan vlan100 traceoptions file mld-snoop-trace
   ```

2. (Optional) Configure the maximum number of trace files and size of the trace files:

   ```
   [edit protocols mld-snooping ]
   user@switch # set vlan vlan-name traceoptions file files number size size
   ```

   For example:

   ```
   [edit protocols mld-snooping ]
   user@switch # set vlan vlan100 traceoptions file files 5 size 1m
   ```

   causes the contents of the trace file to be emptied and archived in a .gz file when the file reaches 1 MB. Four archive files are maintained, the contents of which are rotated whenever the current active trace file is archived.

   If you omit this step, the maximum number of trace files defaults to 10, and the maximum file size to 128 KB.

3. Specify one of the tracing flags shown in Table 8 on page 156:

   ```
   [edit protocols mld-snooping ]
   user@switch # set vlan vlan-name traceoptions flag flagname
   ```

   For example, to perform trace operations on VLAN-related events and on MLD query messages:

```
[edit protocols mld-snooping ]
user@switch# set vlan vlan100 traceoptions flag vlan

[edit protocols mld-snooping ]
user@switch# set vlan vlan100 traceoptions flag query
```

Viewing, Stopping, and Restarting Tracing Operations

When you commit the configuration, tracing operations begin. You can view the trace file in the **/var/log** directory. For example:

```
user@switch> file show /var/log/mld-snoop-trace
```

You can stop and restart tracing operations by deactivating and reactivating the configuration:

```
[edit]
user@switch# deactivate protocols mld-snooping traceoptions

[edit]
user@switch# activate protocols mld-snooping traceoptions
```

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure) on page 142
- *Tracing and Logging Junos OS Operations*

Example: Configuring MLD Snooping on EX Series Switches

You can enable MLD snooping on a VLAN to constrain the flooding of IPv6 multicast traffic on a VLAN. When MLD snooping is enabled, a switch examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. Based on what it learns, the switch then forwards IPv6 multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

This example describes how to configure MLD snooping:

Requirements

This example uses the following software and hardware components:

- One EX Series switch
- Junos OS Release 12.1 or later

Before you configure MLD snooping, be sure you have:

- Configured the **vlan100** VLAN on the switch
- Assigned interfaces **ge-0/0/0**, **ge-0/0/1**, **ge-0/0/2**, and **ge-0/0/12** to **vlan100**
- Configured **ge-0/0/12** as a trunk interface.

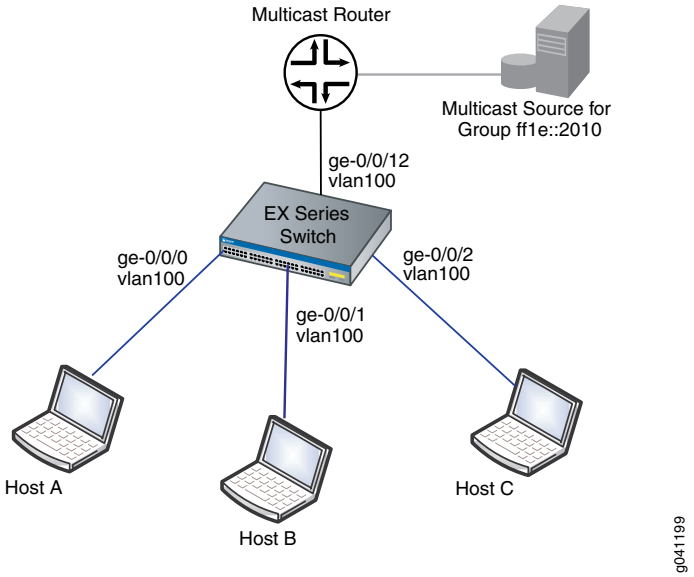
See *Configuring VLANs for EX Series Switches (CLI Procedure)*.

Overview and Topology

In this example, interfaces **ge-0/0/0**, **ge-0/0/1**, and **ge-0/0/2** on the switch are in **vlan100** and are connected to hosts that are potential multicast receivers. Interface **ge-0/0/12**, a trunk interface also in **vlan100**, is connected to a multicast router. The router acts as the MLD querier and forwards multicast traffic for group **ff1e::2010** to the switch from a multicast source.

The example topology is illustrated in Figure 19 on page 149.

Figure 21: Example MLD Snooping Topology



In this example topology, the multicast router forwards multicast traffic to the switch from the source when it receives a memberhsip report for group **ff1e::2010** from one of the hosts—for example, Host B. If MLD snooping is not enabled on **vlan100**, the switch floods the multicast traffic on all interfaces in **vlan100** (except for interface **ge-0/0/12**). If MLD snooping is enabled on **vlan100**, the switch monitors the MLD messages between the hosts and router, allowing it to determine that only Host B is interested in receiving the multicast traffic. The switch then forwards the multicast traffic only to interface **ge-0/0/1**.

This example shows how to enable MLD snooping on **vlan100**. It also shows how to perform the following optional configurations, which can reduce group join and leave latency:

- Configure immediate leave on the VLAN. When immediate leave is configured, the switch stops forwarding multicast traffic on an interface when it detects that the last member of the multicast group has left the group. If immediate leave is not configured, the switch waits until the group-specific membership queries time out before it stops forwarding traffic.
- Configure **ge-0/0/12** as a static multicast-router interface. In this topology, **ge-0/0/12** always leads to the multicast router. By statically configuring **ge-0/0/12** as a multicast-router interface, you avoid any delay imposed by the switch having to learn that **ge-0/0/12** is a multicast-router interface.

Configuration

To configure MLD snooping on a switch:

CLI Quick Configuration

To quickly configure MLD snooping, copy the following commands and paste them into the switch terminal window:

```
[edit]
set protocols mld-snooping vlan vlan100
set protocols mld-snooping vlan vlan100 immediate-leave
set protocols mld-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
```

Step-by-Step Procedure

To configure MLD snooping:

1. Enable MLD snooping on VLAN **vlan100**:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100
   ```

2. Configure the switch to immediately remove a group membership from an interface when it receives a leave report from the last member of the group on the interface:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100 immediate-leave
   ```

3. Statically configure interface **ge-0/0/12** as a multicast-router interface:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
   ```

Results

Check the results of the configuration:

```
[edit protocols]
user@switch# show mld-snooping
vlan vlan100 {
    immediate-leave;
    interface ge-0/0/12.0 {
        multicast-router-interface;
    }
}
```

Verifying MLD Snooping Configuration

To verify that MLD snooping is enabled on the VLAN and the MLD snooping forwarding interfaces are correct, perform the following task:

- Verifying MLD Snooping Interface Membership on VLAN vlan100 on page 163

Verifying MLD Snooping Interface Membership on VLAN vlan100

Purpose Verify that MLD snooping is enabled on **vlan100** and that the multicast-router interface is statically configured:

Action Show the group memberships maintained by MLD snooping for **vlan100**:

```
user@switch> show mld-snooping membership vlan vlan100 detail
VLAN: vlan100 Tag: 100 (Index: 8)
    Router interfaces:
        ge-0/0/12.0 static Uptime: 00:15:03
   Group: ff1e::2010
     ge-0/0/1.0 Timeout: 225 Flags: <V2-hosts>
     Last reporter: fe80::2020:1:1:3
```

Meaning MLD snooping is running on **vlan100**, and interface **ge-0/0/12.0** is a statically configured multicast-router interface. Because the multicast group **ff1e::2010** is listed, at least one host in the VLAN is a current member of the multicast group and that host is on interface **ge-0/0/1.0**.

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167
- Understanding MLD Snooping on page 125

Example: Configuring MLD Snooping on Switches with ELS Support

NOTE: This example uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. For ELS details, see *Using the Enhanced Layer 2 Software CLI*.

You can enable MLD snooping on a VLAN to constrain the flooding of IPv6 multicast traffic on a VLAN. When MLD snooping is enabled, a switch examines MLD messages between hosts and multicast routers and learns which hosts are interested in receiving multicast traffic for a multicast group. On the basis of what it learns, the switch then forwards IPv6 multicast traffic only to those interfaces connected to interested receivers instead of flooding the traffic to all interfaces.

This example describes how to configure MLD snooping:

Requirements

This example uses the following software and hardware components:

- One switch running Junos OS with ELS
- Junos OS Release 13.3 or later for EX Series switches or Junos OS Release 15.1X53-D10 or later for QFX10000 switches

Before you configure MLD snooping, be sure you have:

- Configured the vlan 100 VLAN on the switch.
- Assigned interfaces ge-0/0/0, ge-0/0/1, ge-0/0/2, and ge-0/0/12 to vlan100.
- Configured ge-0/0/12 as a trunk interface.

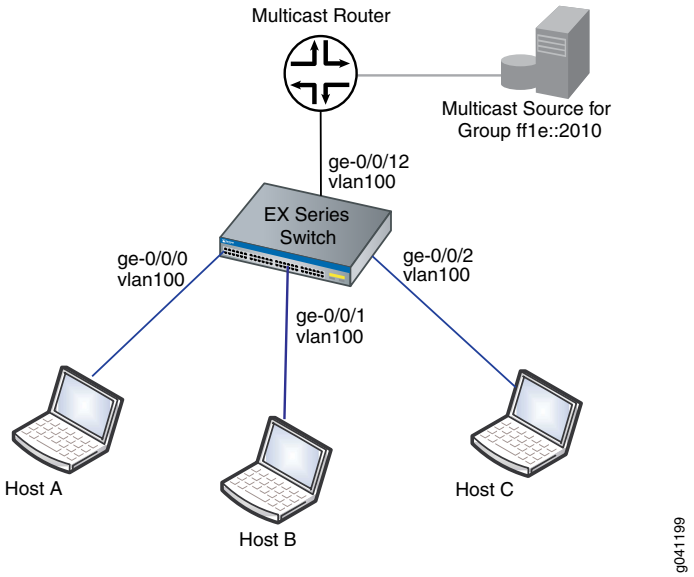
See *Configuring VLANs for EX Series Switches (CLI Procedure)* or *Configuring VLANs on Switches with Enhanced Layer 2 Support*.

Overview and Topology

In this example, interfaces ge-0/0/0, ge-0/0/1, and ge-0/0/2 on the switch are in vlan100 and are connected to hosts that are potential multicast receivers. Interface ge-0/0/12, a trunk interface also in vlan100, is connected to a multicast router. The router acts as the MLD querier and forwards multicast traffic for group ff1e::2010 to the switch from a multicast source.

The topology for this example is illustrated in Figure 19 on page 149.

Figure 22: MLD Snooping Topology Example



In this sample topology, the multicast router forwards multicast traffic to the switch from the source when it receives a membership report for group ff1e::2010 from one of the hosts—for example, Host B. If MLD snooping is not enabled on vlan100, the switch floods the multicast traffic on all interfaces in vlan100 (except for interface ge-0/0/12). If MLD snooping is enabled on vlan100, the switch monitors the MLD messages between the hosts and router, allowing it to determine that only Host B is interested in receiving the multicast traffic. The switch then forwards the multicast traffic only to interface ge-0/0/1.

This example shows how to enable MLD snooping on vlan100. It also shows how to perform the following optional configurations, which can reduce group join and leave latency:

- Configure immediate leave on the VLAN. When immediate leave is configured, the switch stops forwarding multicast traffic on an interface when it detects that the last member of the multicast group has left the group. If immediate leave is not configured, the switch waits until the group-specific membership queries time out before it stops forwarding traffic.
- Configure ge-0/0/12 as a static multicast-router interface. In this topology, ge-0/0/12 always leads to the multicast router. By statically configuring ge-0/0/12 as a multicast-router interface, you avoid any delay imposed by the switch having to learn that ge-0/0/12 is a multicast-router interface.

Configuration

To configure MLD snooping on a switch:

CLI Quick Configuration

To quickly configure MLD snooping, copy the following commands and paste them into the switch terminal window:

```
[edit]
```

```
set protocols mld-snooping vlan vlan100
set protocols mld-snooping vlan vlan100 immediate-leave
set protocols mld-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
```

Step-by-Step Procedure

To configure MLD snooping:

1. Enable MLD snooping on the VLAN vlan100:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100
   ```

2. Configure the switch to immediately remove a group membership from an interface when it receives a leave report from the last member of the group on the interface:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100 immediate-leave
   ```

3. Statically configure interface ge-0/0/12 as a multicast-router interface:

   ```
   [edit protocols]
   user@switch# set mld-snooping vlan vlan100 interface ge-0/0/12 multicast-router-interface
   ```

Results

Check the results of the configuration:

```
[edit protocols]
user@switch# show mld-snooping
vlan vlan100 {
    immediate-leave;
    interface ge-0/0/12.0 {
        multicast-router-interface;
    }
}
```

Verifying MLD Snooping Configuration

To verify that MLD snooping is enabled on the VLAN and the MLD snooping forwarding interfaces are correct, perform the following task:

- Verifying MLD Snooping Interface Membership on VLAN vlan100 on page 166

Verifying MLD Snooping Interface Membership on VLAN vlan100

Purpose

Verify that MLD snooping is enabled on the VLAN vlan 100 and that the multicast-router interface is statically configured:

Action

Show the MLD snooping information for ge-0/0/12.0:

```
user@switch> show mld snooping interface
Instance: default-switch

Vlan: vlan100
```

```
Learning-Domain: default
Interface: ge-0/0/12.0
    State:         Up Groups:      3
    Immediate leave: On
    Router interface: yes

Configured Parameters:
MLD Query Interval: 125.0
MLD Query Response Interval: 10.0
MLD Last Member Query Interval: 1.0
MLD Robustness Count: 2
```

Meaning MLD snooping is running on **vlan100**, and interface **ge-0/0/12.0** is a statically configured multicast-router interface. Immediate leave is enabled on the interface.

Related Documentation

- Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure) on page 142
- Verifying MLD Snooping on Switches on page 170
- Understanding MLD Snooping on page 125

Verifying MLD Snooping on EX Series Switches (CLI Procedure)

Multicast Listener Discovery (MLD) snooping constrains the flooding of IPv6 multicast traffic on VLANs on a switch. This topic describes how to verify MLD snooping operation on the switch.

It covers:

- Verifying MLD Snooping Memberships on page 167
- Verifying MLD Snooping VLANs on page 168
- Viewing MLD Snooping Statistics on page 169
- Viewing MLD Snooping Routing Information on page 169

Verifying MLD Snooping Memberships

Purpose Determine group memberships, multicast-router interfaces, host MLD versions, and the current values of timeout counters.

Action Enter the following command:

```
user@switch> show mld-snooping membership detail
VLAN: mld-vlan Tag: 100 (Index: 3)
    Router interfaces:
        ge-1/0/0.0 dynamic Uptime: 00:14:24 timeout: 253
   Group: ff1e::2010
     ge-1/0/30.0 Timeout: 180 Flags: <V2-hosts>
     Last reporter: fe80::2020:1:1:3
```

```
        Include source: 2020:1:1:1::2
        Include source: 2020:1:1:1::5
```

Meaning The switch has multicast membership information for one VLAN on the switch, **mld-vlan**. MLD snooping might be enabled on other VLANs, but the switch does not have any multicast membership information for them. The following information is provided:

- Information on the multicast-router interfaces for the VLAN—in this case, **ge-1/0/0.0**. The multicast-router interface has been learned by MLD snooping, as indicated by **dynamic**. The **timeout** value shows how many seconds from now the interface will be removed from the multicast forwarding table if the switch does not receive MLD queries or Protocol Independent Multicast (PIM) updates on the interface.
- Information about the group memberships for the VLAN:
  - Currently, the VLAN has membership in only one multicast group, **ff1e::2010**.
  - The host or hosts that have reported membership in the group are on interface **ge-1/0/30.0**. The interface group membership will time out in 180 seconds if no hosts respond to membership queries during this interval. The flags field shows the lowest version of MLD used by a host that is currently a member of the group, which in this case is MLD version 2 (MLDv2).
  - The last host that reported membership in the group has address **fe80::2020:1:1:3**.
  - Because interface has MLDv2 hosts on it, the source addresses from which the MLDv2 hosts want to receive group multicast traffic are shown (addresses **2020:1:1:1::2** and **2020:1:1:1::5**). The **timeout** value for the interface group membership is derived from the largest timeout value for all sources addresses for the group.

Verifying MLD Snooping VLANs

Purpose Verify that MLD snooping is enabled on a VLAN and display MLD snooping information for each VLAN on which MLD snooping is enabled.

Action Enter the following command:

```
user@switch> show mld-snooping vlans detail
VLAN: v10, Tag: 10
    Interface: ge-1/0/0.0, tagged, Groups: 0, Router
    Interface: ge-1/0/30.0, untagged, Groups: 1
    Interface: ge-12/0/30.0, untagged, Groups: 0
VLAN: v20, Tag: 20
    Interface: ge-1/0/0.0, tagged, Groups: 0, Router
    Interface: ge-1/0/31.0, untagged, Groups: 0
    Interface: ge-12/0/31.0, untagged, Groups: 1
```

Meaning MLD snooping is configured on two VLANs on the switch: **v10** and **v20**. Each interface in each VLAN is listed and the following information is provided:

- Whether the interface is a trunk (**tagged**) or access (**untagged**) interface.
- How many multicast groups the interface belongs to.
- Whether the interface is a multicast-router interface (**Router**).

Viewing MLD Snooping Statistics

Purpose Display MLD snooping statistics, such as number of MLD queries, reports, and leaves received and how many of these MLD messages contained errors.

Action Enter the following command:

```
user@switch> show mld-snooping statistics
Bad length: 0 Bad checksum: 0 Invalid interface: 0
Not local: 0 Receive unknown: 0 Timed out: 0

    MLD Type            Received      Transmitted    Recv Errors
    Queries:               74295                0              0
    Reports:            18148423                0       16333523
    Leaves:                    0                0              0
    Other:                     0                0              0
```

Meaning The output shows how many MLD messages of each type—**Queries**, **Reports**, **Leaves**—the switch received or transmitted on interfaces on which MLD snooping is enabled. For each message type, it also shows the number of MLD packets the switch received that had errors—for example, packets that do not conform to the MLDv1 or MLDv2 standards. If the **Recv Errors** count increases, verify that the hosts are compliant with MLDv1 or MLDv2 standards. If the switch is unable to recognize the MLD message type for a packet, it counts the packet under **Receive unknown**.

Viewing MLD Snooping Routing Information

Purpose Display the next-hop information maintained in the multicast forwarding table.

Action Enter the following command:

```
user@switch> show mld-snooping route detail
VLAN             Group          Next-hop
mld-vlan         ::0000:2010        1323
        Interfaces: ge-1/0/30.0, ge-1/0/33.0
VLAN             Group          Next-hop
mld-vlan         ff00::             1317
        Interfaces: ge-1/0/0.0, ge-1/0/33.0
VLAN             Group          Next-hop
mld-vlan         ::0000:0000        1317
        Interfaces: ge-1/0/0.0
VLAN             Group          Next-hop
mld-vlan1        ::0000:2010        1324
        Interfaces: ge-12/0/31.0
VLAN             Group          Next-hop
mld-vlan1        ff00::             1318
```

```
        Interfaces: ae200.0
VLAN             Group          Next-hop
mld-vlan1        ::0000:0000        1318
        Interfaces: ae200.0
```

Meaning The output shows the next-hop interfaces for a given multicast group on a VLAN. Only the last 32 bits of the group address are shown because the switch uses only these bits in determining multicast routes. For example, route **::0000:2010** on **mld-vlan** has next-hop interfaces **ge-1/0/30.0** and **ge-1/0/33.0**.

Related Documentation

- clear mld-snooping membership on page 1403
- clear mld-snooping statistics on page 1404
- Example: Configuring MLD Snooping on EX Series Switches on page 148
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

Verifying MLD Snooping on Switches

NOTE: This topic uses Junos OS with support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that does not support ELS, see "Verifying MLD Snooping on EX Series Switches (CLI Procedure)" on page 167. For ELS details, see *Using the Enhanced Layer 2 Software CLI*.

Multicast Listener Discovery (MLD) snooping constrains the flooding of IPv6 multicast traffic on VLANs. This topic describes how to verify MLD snooping operation on a VLAN.

It covers:

- Verifying MLD Snooping Memberships on page 170
- Verifying MLD Snooping Interfaces on page 171
- Viewing MLD Snooping Statistics on page 172
- Viewing MLD Snooping Routing Information on page 173

Verifying MLD Snooping Memberships

Purpose Verify that MLD snooping is enabled on a VLAN and determine group memberships.

Action Enter the following command:

```
user@switch> show mld snooping membership detail
Instance: default-switch

Vlan: v1
```

```
Learning-Domain: default
Interface: ge-0/0/1.0, Groups: 1
    Group: ff05::1
        Group mode: Exclude
        Source: ::
        Last reported by: fe80::
        Group timeout:     259 Type: Dynamic
Interface: ge-0/0/2.0, Groups: 0
```

Meaning The switch has multicast membership information for one VLAN on the switch, **v1**. MLD snooping might be enabled on other VLANs, but the switch does not have any multicast membership information for them.

- The following information is provided about the group memberships for the VLAN:
  - Currently, the VLAN has membership in only one multicast group, **ff05::1**.
  - The host or hosts that have reported membership in the group are on interface ge-0/0/1.0.
  - The last host that reported membership in the group has address **fe80::**.
  - The interface group membership will time out in **259** seconds if no hosts respond to membership queries during this interval.
  - The group membership has been learned by MLD snooping, as indicated by **Dynamic**.

Verifying MLD Snooping Interfaces

Purpose Display MLD snooping information for each interface on which MLD snooping is enabled.

Action Enter the following command:

```
user@switch>show mld snooping interface
Instance: default-switch

Vlan: v100

Learning-Domain: default
Interface: ge-0/0/1.0
    State:         Up Groups:      1
    Immediate leave: Off
    Router interface: no
Interface: ge-0/0/2.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no

Configured Parameters:
MLD Query Interval: 125.0
MLD Query Response Interval: 10.0
MLD Last Member Query Interval: 1.0
MLD Robustness Count: 2
```

Meaning MLD snooping is configured on one VLAN on the switch, **v100**. Each interface in each VLAN is listed and the following information is provided:

- How many multicast groups the interface belongs to.
- Whether immediate leave has been configured for the interface.
- Whether the interface is a multicast-router interface.

The output also shows the configured parameters for the MLD querier.

Viewing MLD Snooping Statistics

Purpose Display MLD snooping statistics, such as number of MLD queries, reports, and leaves received and how many of these MLD messages contained errors.

Action Enter the following command:

```
user@switch>show mld snooping statistics
Vlan: v1
MLD Message type            Received       Sent  Rx errors
Listener Query (v1/v2)             0          4          0
Listener Report (v1)             447          0          0
Listener Done (v1/v2)              0          0          0
Listener Report (v2)               0          0          0
Other Unknown types                                      0

Vlan: v2
MLD Message type            Received       Sent  Rx errors
Listener Query (v1/v2)             0          4          0
Listener Report (v1)             154          0          0
Listener Done (v1/v2)              0          0          0
Listener Report (v2)               0          0          0
Other Unknown types                                      0

Instance: default-switch
MLD Message type            Received       Sent  Rx errors
Listener Query (v1/v2)             0          8          0
Listener Report (v1)             601          0          0
Listener Done (v1/v2)              0          0          0
Listener Report (v2)               0          0          0
Other Unknown types                                      0

MLD Global Statistics
Bad Length                    0
Bad Checksum                  0
Bad Receive If                0
Rx non-local                  0
Timed out                     0
```

Meaning The output shows how many MLD messages of each type—**Queries**, **Done**, **Report**—the switch received or transmitted on interfaces on which MLD snooping is enabled. For each message type, it also shows the number of MLD packets the switch received that had errors—for example, packets that do not conform to the MLDv1 or MLDv2 standards. If

the **Rx errors** count increases, verify that the hosts are compliant with MLDv1 or MLDv2 standards. If the switch is unable to recognize the MLD message type for a packet, it counts the packet under **Other Unknown types**.

Viewing MLD Snooping Routing Information

Purpose Display the next-hop information maintained in the multicast snooping forwarding table.

Action Enter the following command:

```
user@switch>show multicast snooping route
Nexthop Bulking: OFF

             Family: INET6

Group: ff00::/8
    Source: ::/128
    Vlan: v1

Group: ff02::1/128
    Source: ::/128
    Vlan: v1
        Downstream interface list:
        ge-1/0/16.0

Group: ff05::1/128
    Source: ::/128
    Vlan: v1
        Downstream interface list:
        ge-1/0/16.0

Group: ff06::1/128
    Source: ::/128
    Vlan: v1
        Downstream interface list:
        ge-1/0/16.0
```

Meaning The output shows the next-hop interfaces for a given multicast group on a VLAN. For example, route **ff02::1/128** on VLAN **v1** has the next-hop interface **ge-1/0/16.0**.

Related Documentation

- *clear mld snooping membership*
- *clear mld snooping statistics*
- Example: Configuring MLD Snooping on Switches with ELS Support on page 163
- Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure) on page 142

CHAPTER 5

Configuring Multicast VLAN Registration

Understanding Multicast VLAN Registration

Multicast VLAN registration (MVR) enables you to efficiently distribute IPTV multicast streams across an Ethernet ring-based Layer 2 network and reduce the amount of bandwidth consumed by this multicast traffic.

In a standard Layer 2 network, a multicast stream received on one VLAN is never distributed to interfaces outside that VLAN. If hosts in multiple VLANs request the same multicast stream, a separate copy of that multicast stream is distributed to the requesting VLANs.

MVR introduces the concept of a *multicast source VLAN* (MVLAN), which is created by MVR and becomes the only VLAN over which IPTV multicast traffic flows throughout the Layer 2 network. A Juniper Networks EX Series switch or QFX Series switch that is enabled for MVR selectively forwards IPTV multicast traffic from interfaces on the MVLAN (source interfaces) to hosts that are connected to interfaces that are not part of the MVLAN. These interfaces are known as *MVR receiver ports*. The MVR receiver ports can receive traffic from a port on the MVLAN but cannot send traffic onto the MVLAN, and they remain in their own VLANs for bandwidth and security reasons.

This topic includes:

How MVR Works

In many ways, MVR is similar to IGMP snooping. Both MVR and IGMP snooping monitor IGMP join and leave messages and build forwarding tables based on the media access control (MAC) addresses of the hosts sending those IGMP messages. Whereas IGMP snooping operates within a given VLAN to regulate multicast traffic, MVR can operate with hosts on different VLANs in a Layer 2 network to selectively deliver IPTV multicast traffic to requesting hosts, thereby reducing the amount of bandwidth needed to forward multicast traffic.

When you configure an MVLAN, you assign a range of multicast group addresses to it. You then configure other VLANs to be MVR receiver VLANs, which receive multicast streams from the MVLAN. The MVR receiver ports comprise all the interfaces that exist on any of the MVR receiver VLANs. Interfaces that are on the MVLAN itself cannot be MVR receiver ports for that MVLAN.

NOTE: MVR is supported on VLANs running IGMP version 2 (IGMPv2) only.

MVR Modes

MVR operates in two modes: MVR transparent mode and MVR proxy mode. Both modes enable MVR to forward only one copy of a multicast stream to the Layer 2 network.

- MVR Transparent Mode on page 176
- MVR Proxy Mode on page 176

MVR Transparent Mode

In MVR transparent mode, the switch receives one copy of each IPTV multicast stream and then replicates the stream only to those hosts that want to receive it, while forwarding all other types of multicast traffic without modification. Transparent mode is the default mode.

The switch handles IGMP packets destined for both the multicast source VLAN and multicast receiver VLANs in the same way that it handles them when MVR is not being used. That is, when a host on a VLAN sends IGMP join and leave messages, the switch floods the messages to all router interfaces in the VLAN. Similarly, when a VLAN receives IGMP queries from its router interfaces, it floods the queries to all interfaces in the VLAN.

If a host on a multicast receiver port joins an MVR group on the multicast receiver VLAN, the appropriate bridging entry is added and the MVLAN forwards that group's IPTV multicast traffic on that port (even though that port is not in the MVLAN). Likewise, if a host on a multicast receiver port leaves an MVR group on the multicast receiver VLAN, the appropriate bridging entry is deleted, and the MVLAN stops forwarding that group's IPTV multicast traffic on that port. In addition, you can configure the switch to statically install the bridging entries on the multicast receiver VLAN.

MVR Proxy Mode

When you use MVR in proxy mode, the switch acts as a proxy for any MVR group in both the upstream and downstream directions. In the downstream direction, the switch acts as the querier for the groups in the MVR receiver VLANs. In the upstream direction, the switch originates the IGMP reports and leaves and answers IGMP queries from multicast routers. When the MVR receiver VLANs receive IGMP joins and leaves, the switch creates bridging entries on the MVLAN as needed, as it does in MVR transparent mode. In addition, the switch sends out IGMP joins and leaves on the MVLAN based on these bridging entries.

Configuring MVR proxy mode on the MVLAN automatically enables IGMP snooping proxy mode on all MVR receiver VLANs as well as on the MVLAN.

Related Documentation

- *Understanding FIP Snooping, FBF, and MVR Filter Scalability*
- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure)

Multicast VLAN registration (MVR) enables hosts that are not part of a multicast source VLAN (MVLAN) to still receive multicast streams from the MVLAN, allowing an MVLAN to be shared across a Layer 2 network. Hosts remain in their own VLANs for bandwidth and security reasons but are able to receive multicast streams from the MVLAN.

You can configure one or more VLANs on a switch to be MVLANs or MVR receiver VLANs. By default, MVR is not configured on EX Series switches and the QFX Series.

NOTE: MVR is supported on VLANs running IGMP version 2 (IGMPv2) only.

NOTE: When you configure MVR, the following restrictions apply:

- You cannot enable multicast protocols on VLAN interfaces that are members of MVLANs.
- If you configure an MVLAN in proxy mode, IGMP snooping proxy mode is automatically enabled on all MVR receiver VLANs of this MVLAN. If a VLAN is an MVR receiver VLAN for multiple MVLANs, all of the MVLANs must have proxy mode enabled or all must have proxy mode disabled. You can enable proxy mode only on VLANs that are configured as MVR source VLANs and that are not configured for Q-in-Q tunneling.
- After you configure a VLAN as an MVLAN, that VLAN is no longer available for other uses.

To configure MVR:

1. Configure the VLAN named mv0 to be an MVLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan mv0 data-forwarding source groups
  225.10.0.0/16
```

2. Configure the MVLAN mv0 to be a proxy VLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan mv0 proxy source-address 10.0.0.1
```

3. Configure the VLAN named v2 to be an MVR receiver VLAN with mv0 as its source:

```
[edit protocols]
user@switch# set igmp-snooping vlan v2 data-forwarding receiver source-vlans mv0
```

4. Install forwarding entries in the MVR receiver VLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan v2 data-forwarding receiver install
```

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Understanding Multicast VLAN Registration on page 175

Example: Configuring Multicast VLAN Registration on EX Series Switches

Multicast VLAN registration (MVR) enables hosts that are not part of a multicast VLAN (MVLAN) to receive multicast streams from the MVLAN, which enable the MVLAN to be shared across the Layer 2 network and eliminate the need to send duplicate multicast streams to each requesting VLAN in the network. Hosts remain in their own VLANs for bandwidth and security reasons.

This example describes how to configure MVR on EX Series switches and the QFX Series.

Requirements

This example uses the following hardware and software components:

- One EX Series or QFX Series switch
- Junos OS Release 9.6 or later for EX Series switches or Junos OS Release 12.3 or later for the QFX Series

Before you configure MVR, be sure you have:

- Configured two or more VLANs on the switch. See the task for your platform:
  - *Example: Setting Up Bridging with Multiple VLANs for EX Series Switches*
  - *Example: Setting Up Bridging with Multiple VLANs on Switches* for the QFX Series and EX4600 switch
- Connected the switch to a network that can transmit IPTV multicast streams from a video server.
- Connected a host that is capable of receiving IPTV multicast streams to an interface in one of the VLANs.

Overview and Topology

In a standard Layer 2 network, a multicast stream received on one VLAN is never distributed to interfaces outside that VLAN. If hosts in multiple VLANs request the same multicast stream, a separate copy of that multicast stream is distributed to the requesting VLANs.

MVR introduces the concept of a *multicast source VLAN* (MVLAN), which is created by MVR and becomes the only VLAN over which multicast traffic flows throughout the Layer 2 network. Multicast traffic can then be selectively forwarded from interfaces on the MVLAN (source ports) to hosts that are connected to interfaces (multicast receiver ports) that are not part of the multicast source VLAN. When you configure an MVLAN, you assign a range of multicast group addresses to it. You then configure other VLANs to be MVR receiver VLANs, which receive multicast streams from the MVLAN. The MVR receiver ports comprise all the interfaces that exist on any of the MVR receiver VLANs.

You can configure MVR to operate in one of two modes: transparent mode (the default mode) or proxy mode. Both modes enable MVR to forward only one copy of a multicast stream to the Layer 2 network.

In transparent mode, the switch receives one copy of each IPTV multicast stream and then replicates the stream only to those hosts that want to receive it, while forwarding all other types of multicast traffic without modification. Figure 23 on page 180 shows how MVR operates in transparent mode.

In proxy mode, the switch acts as a proxy for the IGMP multicast router in the MVLAN for MVR group memberships established in the MVR receiver VLANs and generates and sends IGMP packets into the MVLAN as needed. Figure 24 on page 181 shows how MVR operates in proxy mode.

This example shows how to configure MVR in both transparent mode and proxy mode on an EX Series switch or the QFX Series. The topology includes a video server that is connected to a multicast router, which in turn forwards the IPTV multicast traffic in the MVLAN to the Layer 2 network.

Figure 23 on page 180 shows the MVR topology in transparent mode. Interfaces P1 and P2 on Switch C belong to service VLAN s0 and MVLAN mv0. Interface P4 of Switch C also belongs to service VLAN s0. In the upstream direction of the network, only non-IPTV traffic is being carried in individual customer VLANs of service VLAN s0. VLAN c0 is an example of this type of customer VLAN. IPTV traffic is being carried on MVLAN mv0. If any host on any customer VLAN connected to port P4 requests an MVR stream, Switch C takes the stream from VLAN mv0 and replicates that stream onto port P4 with tag mv0. IPTV traffic, along with other network traffic, flows from port P4 out to the Digital Subscriber Line Access Multiplexer (DSLAM) D1.

Figure 23: MVR Topology in Transparent Mode

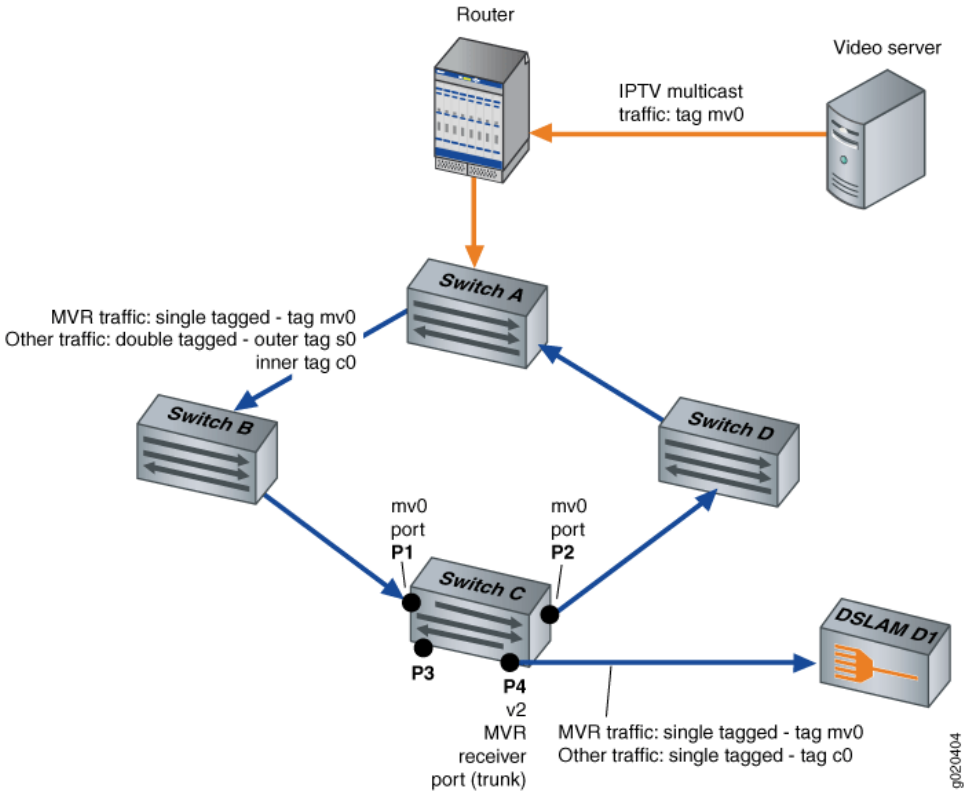
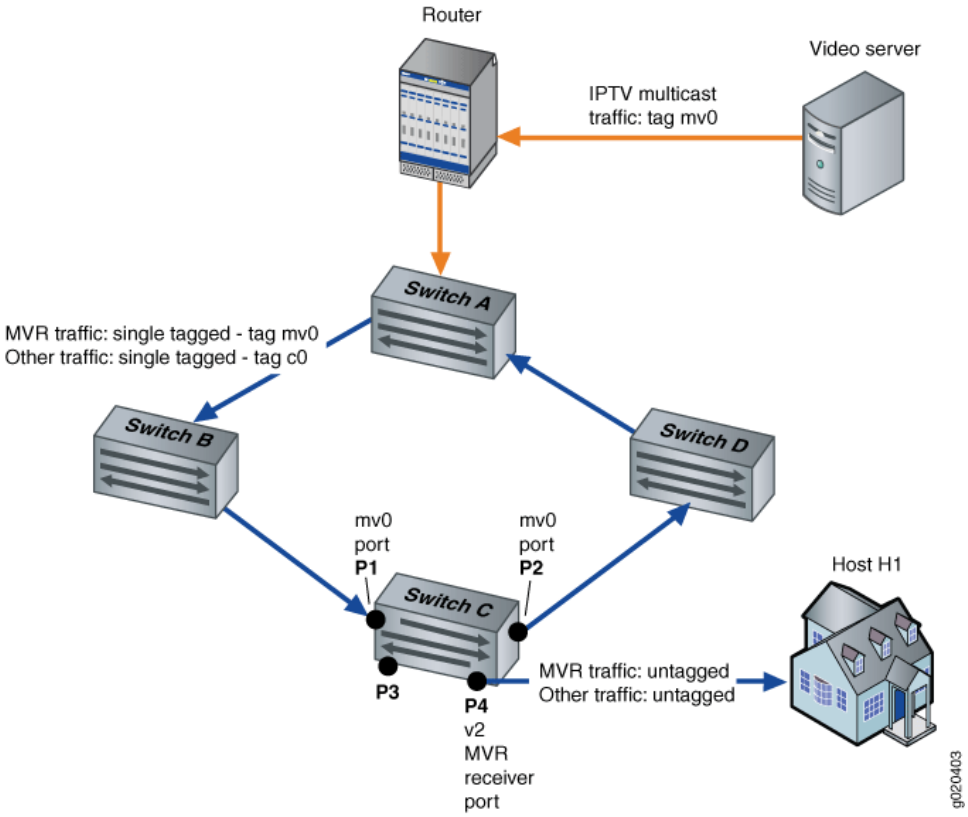


Figure 24 on page 181 shows the MVR topology in proxy mode. Interfaces P1 and P2 on Switch C belong to MVLAN mv0 and customer VLAN c0. Interface P4 on Switch C is an access port of customer VLAN c0. In the upstream direction of the network, only non-IPTV traffic is being carried on customer VLAN c0. Any IPTV traffic requested by hosts on VLAN c0 is replicated untagged to port P4 based on streams received in MVLAN mv0. IPTV traffic flows from port P4 out to an IPTV-enabled device in Host H1. Other traffic, such as data and voice traffic, also flows from port P4 to other network devices in Host H1.

Figure 24: MVR Topology in Proxy Mode



For information on VLAN tagging, see the topic for your platform:

- *Understanding Bridging and VLANs on Switches*

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit protocols igmp-snooping]** hierarchy level.

```
set vlan mv0 data-forwarding source groups 225.10.0.0/16
set vlan v2 data-forwarding receiver source-vlans mv0
set vlan v2 data-forwarding receiver install
set vlan mv0 proxy source-address 10.1.1.1
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure MVR:

1. Configure VLAN mv0 to be an MVLAN:

```
[edit protocols igmp-snooping]
user@switch# set vlan mv0 data-forwarding source groups 225.10.0.0/16
```

2. Configure VLAN v2 to be a multicast receiver VLAN with mv0 as its source:

```
[edit protocols igmp-snooping]
user@switch# set vlan v2 data-forwarding receiver source-vlans mv0
```

3. (Optional) Install forwarding entries in the multicast receiver VLAN v2:

```
[edit protocols igmp-snooping]
user@switch# set vlan v2 data-forwarding receiver install
```

4. (Optional) Configure MVR in proxy mode:

```
[edit protocols igmp-snooping]
user@switch# set vlan mv0 proxy source-address 10.1.1.1
```

Results From configuration mode, confirm your configuration by entering the **show** command at the **[edit protocols igmp-snooping]** hierarchy level. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
[edit protocols igmp-snooping]
user@switch# show
vlan mv0 {
  proxy {
    source-address 10.1.1.1;
  }
  data-forwarding {
    source {
      groups 225.10.0.0/16;
    }
  }
}
vlan v2 {
  data-forwarding {
    receiver {
      source-vlans mv0;
      install;
    }
  }
}
```

Related Documentation

- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177
- Understanding Multicast VLAN Registration on page 175

PART 3

Configuring Protocol Independent Multicast

CHAPTER 6

Understanding PIM

PIM Overview

The predominant multicast routing protocol in use on the Internet today is Protocol Independent Multicast, or PIM. The type of PIM used on the Internet is PIM sparse mode. PIM sparse mode is so accepted that when the simple term “PIM” is used in an Internet context, some form of sparse mode operation is assumed.

PIM emerged as an algorithm to overcome the limitations of dense-mode protocols such as the Distance Vector Multicast Routing Protocol (DVMRP), which was efficient for dense clusters of multicast receivers, but did not scale well for the larger, sparser, groups encountered on the Internet. The Core Based Trees (CBT) Protocol was intended to support sparse mode as well, but CBT, with its all-powerful core approach, made placement of the core critical, and large conference-type applications (many-to-many) resulted in bottlenecks in the core. PIM was designed to avoid the dense-mode scaling issues of DVMRP and the potential performance issues of CBT at the same time.

Starting in Junos OS Release 15.2, only PIM version 2 is supported. In the CLI, the command for specifying a version (1 or 2) is removed.

PIMv1 and PIMv2 can coexist on the same routing device and even on the same interface. The main difference between PIMv1 and PIMv2 is the packet format. PIMv1 messages use Internet Group Management Protocol (IGMP) packets, whereas PIMv2 has its own IP protocol number (103) and packet structure. All routing devices connecting to an IP subnet such as a LAN must use the same PIM version. Some PIM implementations can recognize PIMv1 packets and automatically switch the routing device interface to PIMv1. Because the difference between PIMv1 and PIMv2 involves the message format, but not the meaning of the message or how the routing device processes the PIM message, a routing device can easily mix PIMv1 and PIMv2 interfaces.

PIM is used for efficient routing to multicast groups that might span wide-area and interdomain internetworks. It is called “protocol independent” because it does not depend on a particular unicast routing protocol. Junos OS supports bidirectional mode, sparse mode, dense mode, and sparse-dense mode.

NOTE: ACX Series routers supports only sparse mode. Dense mode on ACX series is supported only for control multicast groups for auto-discovery of rendezvous point (auto-RP).

PIM operates in several modes: bidirectional mode, sparse mode, dense mode, and sparse-dense mode. In sparse-dense mode, some multicast groups are configured as dense mode (flood-and-prune, [S,G] state) and others are configured as sparse mode (explicit join to rendezvous point [RP], [*,G] state).

PIM drafts also establish a mode known as PIM source-specific mode, or PIM SSM. In PIM SSM there is only one specific source for the content of a multicast group within a given domain.

Because the PIM mode you choose determines the PIM configuration properties, you first must decide whether PIM operates in bidirectional, sparse, dense, or sparse-dense mode in your network. Each mode has distinct operating advantages in different network environments.

- In sparse mode, routing devices must join and leave multicast groups explicitly. Upstream routing devices do not forward multicast traffic to a downstream routing device unless the downstream routing device has sent an explicit request (by means of a join message) to the rendezvous point (RP) routing device to receive this traffic. The RP serves as the root of the shared multicast delivery tree and is responsible for forwarding multicast data from different sources to the receivers.

  Sparse mode is well suited to the Internet, where frequent interdomain join messages and prune messages are common.

  NOTE: On all the EX series switches (except EX4300 and EX9200), QFX5100 switches, and OCX series switches, the rate limit is set to 1pps per SG to avoid overwhelming the rendezvous point (RP), First hop router (FHR) with PIM-sparse mode (PIM-SM) register messages and cause CPU hogs. This rate limit helps in improving scaling and convergence times by avoiding duplicate packets being trapped, and tunneled to RP in software. (Platform support depends on the Junos OS release in your installation.)

- Bidirectional PIM is similar to sparse mode, and is especially suited to applications that must scale to support a large number of dispersed sources and receivers. In bidirectional PIM, routing devices build shared bidirectional trees and do not switch to a source-based tree. Bidirectional PIM scales well because it needs no source-specific (S,G) state. Instead, it builds only group-specific (*,G) state.
- Unlike sparse mode and bidirectional mode, in which data is forwarded only to routing devices sending an explicit PIM join request, dense mode implements a *flood-and-prune* mechanism, similar to the Distance Vector Multicast Routing Protocol (DVMRP). In dense mode, a routing device receives the multicast data on the incoming interface, then forwards the traffic to the outgoing interface list. Flooding occurs periodically and is used to refresh state information, such as the source IP address and multicast group

pair. If the routing device has no interested receivers for the data, and the outgoing interface list becomes empty, the routing device sends a PIM prune message upstream.

Dense mode works best in networks where few or no prunes occur. In such instances, dense mode is actually more efficient than sparse mode.

- Sparse-dense mode, as the name implies, allows the interface to operate on a per-group basis in either sparse or dense mode. A group specified as “dense” is not mapped to an RP. Instead, data packets destined for that group are forwarded by means of PIM dense mode rules. A group specified as “sparse” is mapped to an RP, and data packets are forwarded by means of PIM sparse-mode rules. Sparse-dense mode is useful in networks implementing auto-RP for PIM sparse mode.

NOTE: On SRX Series devices, PIM does not support upstream and downstream interfaces across different virtual routers in flow mode.

Basic PIM Network Components

PIM dense mode requires only a multicast source and series of multicast-enabled routing devices running PIM dense mode to allow receivers to obtain multicast content. Dense mode makes sure that all multicast traffic gets everywhere by periodically flooding the network with multicast traffic, and relies on prune messages to make sure that subnets where all receivers are uninterested in that particular multicast group stop receiving packets.

PIM sparse mode is more complicated and requires the establishment of special routing devices called *rendezvous points* (*RPs*) in the network core. These routing devices are where upstream join messages from interested receivers meet downstream traffic from the source of the multicast group content. A network can have many RPs, but PIM sparse mode allows only one RP to be active for any multicast group.

If there is only one RP in a routing domain, the RP and adjacent links might become congested and form a single point of failure for all multicast traffic. Thus, multiple RPs are the rule, but the issue then becomes how other multicast routing devices find the RP that is the source of the multicast group the receiver is trying to join. This RP-to-group mapping is controlled by a special *bootstrap router* (*BSR*) running the PIM BSR mechanism. There can be more than one bootstrap router as well, also for single-point-of-failure reasons.

The bootstrap router does not have to be an RP itself, although this is a common implementation. The bootstrap router's main function is to manage the collection of RPs and allow interested receivers to find the source of their group's multicast traffic. PIM bootstrap messages are sourced from the loopback address, which is always up. The loopback address must be routable. If it is not routable, then the bootstrap router is unable to send bootstrap messages to update the RP domain members. The **show pim bootstrap** command displays only those bootstrap routers that have routable loopback addresses.

PIM SSM can be seen as a subset of a special case of PIM sparse mode and requires no specialized equipment other than that used for PIM sparse mode (and IGMP version 3).

Bidirectional PIM RPs, unlike RPs for PIM sparse mode, do not need to perform PIM Register tunneling or other specific protocol action. Bidirectional PIM RPs implement no specific functionality. RP addresses are simply a location in the network to rendezvous toward. In fact, for bidirectional PIM, RP addresses need not be loopback interface addresses or even be addresses configured on any routing device, as long as they are covered by a subnet that is connected to a bidirectional PIM-capable routing device and advertised to the network.

Release History Table

Release	Description
15.2	Starting in Junos OS Release 15.2, only PIM version 2 is supported. In the CLI, the command for specifying a version (1 or 2) is removed.

Related Documentation

- Supported IP Multicast Protocol Standards on page 19

PIM on Aggregated Interfaces

You can configure several Protocol Independent Multicast (PIM) features on an interface regardless of its PIM mode (bidirectional, sparse, dense, or sparse-dense mode).

NOTE: ACX Series routers supports only sparse mode. Dense mode on ACX series is supported only for control multicast groups for auto-discovery of rendezvous point (auto-RP).

If you configure PIM on an aggregated (**ae-** or **as-**) interface, each of the interfaces in the aggregate is included in the multicast output interface list and carries the single stream of replicated packets in a load-sharing fashion. The multicast aggregate interface is “expanded” into its constituent interfaces in the next-hop database.

Related Documentation

- *Junos OS Network Interfaces Library for Routing Devices*

CHAPTER 7

Configuring PIM Basics

Configuring Multiple Instances of PIM

PIM instances are supported only for VRF instance types. You can configure multiple instances of PIM to support multicast over VPNs.

To configure multiple instances of PIM, include the following statements:

```
routing-instances {
  routing-instance-name {
    interface interface-name;
    instance-type vrf;
    protocols {
      pim {
        ... pim-configuration ...
      }
    }
  }
}
```

You can include the statements at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* protocols]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]**

Related Documentation

- *Multicast Protocols Feature Guide*

- *Junos OS VPNs Library for Routing Devices*

Changing the PIM Version

Starting in Junos OS Release 15.2, it is no longer necessary to configure the PIM version. Support for PIM version 1 has been removed and the remaining, default, version is PIM 2.

PIM version 2 is the default for both rendezvous point (RP) mode (at the **[edit protocols pim rp static address *address*]** hierarchy level) and for interface mode (at the **[edit protocols pim interface *interface-name*]** hierarchy level).

Release History Table

Release	Description
15.2	Starting in Junos OS Release 15.2, it is no longer necessary to configure the PIM version.

Optimizing the Number of Multicast Flows on QFabric Systems

Because of the distributed nature of QFabric systems, the default configuration does not allow the maximum number of supported Layer 3 multicast flows to be created. To allow a QFabric system to create the maximum number of supported flows, configure the following statement:

```
set fabric routing-options multicast fabric-optimized-distribution
```

After configuring this statement, you must reboot the QFabric Director group to make the change take effect.

Related Documentation

-

Modifying the PIM Hello Interval

Routing devices send hello messages at a fixed interval on all PIM-enabled interfaces. By using hello messages, routing devices advertise their existence as PIM routing devices on the subnet. With all PIM-enabled routing devices advertised, a single designated router for the subnet is established.

When a routing device is configured for PIM, it sends a hello message at a 30-second default interval. The interval range is from 0 through 255. When the interval counts down to 0, the routing device sends another hello message, and the timer is reset. A routing device that receives no response from a neighbor in 3.5 times the interval value drops the neighbor. In the case of a 30-second interval, the amount of time a routing device waits for a response is 105 seconds.

If a PIM hello message contains the hold-time option, the neighbor timeout is set to the hold-time sent in the message. If a PIM hello message does not contain the hold-time option, the neighbor timeout is set to the default hello hold time.

To modify how often the routing device sends hello messages out of an interface:

1. This example shows the configuration for the routing instance. Configure the interface globally or in the routing instance.

   [edit routing-instances PIM.master protocols pim interface fe-3/0/2.0]
   user@host# set hello-interval 255

2. Verify the configuration by checking the **Hello Option Holdtime** field in the output of the **show pim neighbors detail** command.

```
user@host> show pim neighbors detail
Instance: PIM.master
Interface: fe-3/0/2.0
Address: 192.168.195.37, IPv4, PIM v2, Mode: Sparse
Hello Option Holdtime: 255 seconds
Hello Option DR Priority: 1
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Join Suppression supported
Rx Join: Group Source Timeout
225.1.1.1 192.168.195.78 0
225.1.1.1 0

Interface: lo0.0
Address: 10.255.245.91, IPv4, PIM v2, Mode: Sparse
Hello Option Holdtime: 255 seconds
Hello Option DR Priority: 1
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Join Suppression supported

Interface: pd-6/0/0.32768
Address: 0.0.0.0, IPv4, PIM v2, Mode: Sparse
Hello Option Holdtime: 255 seconds
Hello Option DR Priority: 0
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Join Suppression supported
```

Related Documentation

- show pim neighbors on page 1657

Preserving Multicast Performance by Disabling Response to the ping Utility

The ping utility uses ICMP Echo messages to verify connectivity to any device with an IP address. However, in the case of multicast applications, a single ping sent to a multicast address can degrade the performance of routers because the stream of packets is replicated multiple times.

You can disable the router's response to ping (ICMP Echo) packets sent to multicast addresses. The system responds normally to unicast ping packets.

To disable the router's response to ping packets sent to multicast addresses:

1. Include the **no-multicast-echo** statement:

   [edit system]

```
user@host# set no-multicast-echo
```

2. Verify the configuration by checking the **echo drops with broadcast or multicast destination address** field in the output of the **show system statistics icmp** command.

```
user@host> show system statistics icmp
```

```
icmp:
0 drops due to rate limit
0 calls to icmp_error
0 errors not generated because old message was icmp
Output histogram:
echo reply: 21
0 messages with bad code fields
0 messages less than the minimum length
0 messages with bad checksum
0 messages with bad source address
0 messages with bad length
100 echo drops with broadcast or multicast destination address
0 timestamp drops with broadcast or multicast destination address
Input histogram:
echo: 21
21 message responses generated
```

Related Documentation

- *Configuring Junos OS to Disable the Routing Engine Response to Multicast Ping Packets*
- *show system statistics icmp*

Configuring PIM Trace Options

Tracing operations record detailed messages about the operation of routing protocols, such as the various types of routing protocol packets sent and received, and routing policy actions. You can specify which trace operations are logged by including specific tracing flags. The following table describes the flags that you can include.

Flag	Description
all	Trace all operations.
assert	Trace assert messages, which are used to resolve which of the parallel routers connected to a multiaccess LAN is responsible for forwarding packets to the LAN.
autorp	Trace bootstrap, RP, and auto-RP messages.
bidirectional-df-election	Trace bidirectional PIM designated-forwarder (DF) election events.
bootstrap	Trace bootstrap messages, which are sent periodically by the PIM domain's bootstrap router and are forwarded, hop by hop, to all routers in that domain.
general	Trace general events.

Flag	Description
graft	Trace graft and graft acknowledgment messages.
hello	Trace hello packets, which are sent so that neighboring routers can discover one another.
join	Trace join messages, which are sent to join a branch onto the multicast distribution tree.
mdt	Trace messages related to multicast data tunnels.
normal	Trace normal events.
nsr-synchronization	Trace nonstop routing synchronization events
packets	Trace all PIM packets.
policy	Trace poison-route-reverse packets.
prune	Trace prune messages, which are sent to prune a branch off the multicast distribution tree.
register	Trace register and register-stop messages. Register messages are sent to the RP when a multicast source first starts sending to a group.
route	Trace routing information.
rp	Trace candidate RP advertisements.
state	Trace state transitions.
task	Trace task processing.
timer	Trace timer processing.

In the following example, tracing is enabled for all routing protocol packets. Then tracing is narrowed to focus only on PIM packets of a particular type.

To configure tracing operations for PIM:

1. (Optional) Configure tracing at the [**routing-options** hierarchy level to trace all protocol packets.

   ```
   [edit routing-options traceoptions]
   user@host# set file all-packets-trace
   user@host# set flag all
   ```

2. Configure the filename for the PIM trace file.

   [edit protocols pim traceoptions]

```
user@host# set file pim-trace
```

3. (Optional) Configure the maximum number of trace files.

```
[edit protocols pim traceoptions]
user@host# set file files 5
```

4. (Optional) Configure the maximum size of each trace file.

```
[edit protocols pim traceoptions]
user@host# set file size 1m
```

5. (Optional) Enable unrestricted file access.

```
[edit protocols pim traceoptions]
user@host# set file world-readable
```

6. Configure tracing flags.
   Suppose you are troubleshooting issues with PIM version 1 control packets that are received on an interface configured for PIM version 2. The following example shows how to trace messages associated with this problem.

```
[edit protocols pim traceoptions]
user@host# set flag packets | match “Rx V1 Require V2”
```

7. View the trace file.

```
user@host> file list /var/log
user@host> file show /var/log/pim-trace
```

Related Documentation

- PIM Overview on page 185
- *Tracing and Logging Junos OS Operations*

Configuring BFD for PIM

The Bidirectional Forwarding Detection (BFD) Protocol is a simple hello mechanism that detects failures in a network. BFD works with a wide variety of network environments and topologies. A pair of routing devices exchanges BFD packets. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. The BFD failure detection timers have shorter time limits than the Protocol Independent Multicast (PIM) hello hold time, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases

the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

You must specify the minimum transmit and minimum receive intervals to enable BFD on PIM.

To enable failure detection:

1. Configure the interface globally or in a routing instance.

   This example shows the global configuration.

   ```
   [edit protocols pim]
   user@host# edit interface fe-1/0/0.0 family inet bfd-liveness-detection
   ```

2. Configure the minimum transmit interval.

   This is the minimum interval after which the routing device transmits hello packets to a neighbor with which it has established a BFD session. Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set transmit-interval 350
   ```

3. Configure the minimum interval after which the routing device expects to receive a reply from a neighbor with which it has established a BFD session.

   Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set minimum-receive-interval 350
   ```

4. (Optional) Configure other BFD settings.

   As an alternative to setting the receive and transmit intervals separately, configure one interval for both.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set minimum-interval 350
   ```

5. Configure the threshold for the adaptation of the BFD session detection time.

   When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set detection-time threshold 800
   ```

6. Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   ```

```
user@host# set multiplier 50
```

7. Configure the BFD version.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set version 1
```

8. Specify that BFD sessions should not adapt to changing network conditions.

   We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

```
[edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
user@host# set no-adaptation
```

9. Verify the configuration by checking the output of the **show bfd session** command.

Related Documentation

- *show bfd session*

Configuring BFD Authentication for PIM

1. Specify the BFD authentication algorithm for the PIM protocol.
2. Associate the authentication keychain with the PIM protocol.
3. Configure the related security authentication keychain.

Beginning with Junos OS Release 9.6, you can configure authentication for Bidirectional Forwarding Detection (BFD) sessions running over Protocol Independent Multicast (PIM). Routing instances are also supported.

The following sections provide instructions for configuring and viewing BFD authentication on PIM:

- Configuring BFD Authentication Parameters on page 196
- Viewing Authentication Information for BFD Sessions on page 198

Configuring BFD Authentication Parameters

BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

To configure BFD authentication:

1. Specify the algorithm (**keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**, or **simple-password**) to use for BFD authentication on a PIM route or routing instance.

```
[edit protocols pim]
user@host# set interface ge-0/1/5 family inet bfd-liveness-detection authentication
  algorithm keyed-sha-1
```

NOTE: Nonstop active routing (NSR) is not supported with the meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

2. Specify the keychain to be used to associate BFD sessions on the specified PIM route or routing instance with the unique security authentication keychain attributes.

   The keychain you specify must match the keychain name configured at the **[edit security authentication key-chains]** hierarchy level.

   ```
   [edit protocols pim]
   user@host# set interface ge-0/1/5 family inet bfd-liveness-detection authentication
     keychain bfd-pim
   ```

   NOTE: The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

3. Specify the unique security authentication information for BFD sessions:

   - The matching keychain name as specified in Step 2.
   - At least one key, a unique integer between **0** and **63**. Creating multiple keys allows multiple clients to use the BFD session.
   - The secret data used to allow access to the session.
   - The time at which the authentication key becomes active, in the format *yyyy-mm-dd.hh:mm:ss*.

   ```
   [edit security]
   user@host# set authentication-key-chains key-chain bfd-pim key 53 secret $ABC123$/
     start-time 2009-06-14.10:00:00
   ```

   NOTE:
   Security Authentication Keychain is not supported on SRX Series devices.

4. (Optional) Specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

   ```
   [edit protocols pim]
   user@host# set interface ge-0/1/5 family inet bfd-liveness-detection authentication
     loose-check
   ```

5. (Optional) View your configuration by using the **show bfd session detail** or **show bfd session extensive** command.

6. Repeat these steps to configure the other end of the BFD session.

Viewing Authentication Information for BFD Sessions

You can view the existing BFD authentication configuration by using the **show bfd session detail** and **show bfd session extensive** commands.

The following example shows BFD authentication configured for the **ge-0/1/5** interface. It specifies the keyed SHA-1 authentication algorithm and a keychain name of **bfd-pim**. The authentication keychain is configured with two keys. Key **1** contains the secret data "**$ABC123/**" and a start time of June 1, 2009, at 9:46:02 AM PST. Key **2** contains the secret data "**$ABC123/**" and a start time of June 1, 2009, at 3:29:20 PM PST.

```
[edit protocols pim]
interface ge-0/1/5 {
   family inet {
      bfd-liveness-detection {
         authentication {
            key-chain bfd-pim;
            algorithm keyed-sha-1;
         }
      }
   }
}
[edit security]
authentication key-chains {
   key-chain bfd-pim {
      key 1 {
         secret "$ABC123/";
         start-time "2009-6-1.09:46:02 -0700";
      }
      key 2 {
         secret "$ABC123/";
         start-time "2009-6-1.15:29:20 -0700";
      }
   }
}
```

If you commit these updates to your configuration, you see output similar to the following example. In the output for the **show bfd session detail** command, **Authenticate** is displayed to indicate that BFD authentication is configured. For more information about the configuration, use the **show bfd session extensive** command. The output for this command provides the keychain name, the authentication algorithm and mode for each client in the session, and the overall BFD authentication configuration status, keychain name, and authentication algorithm and mode.

show bfd session detail

```
user@host# show bfd session detail

                                                  Detect   Transmit
```

```
Address                 State     Interface      Time     Interval  Multiplier
192.0.2.2                Up        ge-0/1/5.0     0.900     0.300        3
 Client PIM, TX interval 0.300, RX interval 0.300, Authenticate
 Session up time 3d 00:34
 Local diagnostic None, remote diagnostic NbrSignal
 Remote state Up, version 1
 Replicated
```

show bfd session extensive

```
user@host# show bfd session extensive
                                                   Detect   Transmit
Address                 State     Interface      Time     Interval  Multiplier
192.0.2.2                Up        ge-0/1/5.0     0.900     0.300        3
 Client PIM, TX interval 0.300, RX interval 0.300, Authenticate
        keychain bfd-pim, algo keyed-sha-1, mode strict
  Session up time 00:04:42
 Local diagnostic None, remote diagnostic NbrSignal
 Remote state Up, version 1
 Replicated
 Min async interval 0.300, min slow interval 1.000
 Adaptive async TX interval 0.300, RX interval 0.300
 Local min TX interval 0.300, minimum RX interval 0.300, multiplier 3
 Remote min TX interval 0.300, min RX interval 0.300, multiplier 3
 Local discriminator 2, remote discriminator 2
 Echo mode disabled/inactive
 Authentication enabled/active, keychain bfd-pim, algo keyed-sha-1, mode strict
```

Release History Table

Release	Description
9.6	Beginning with Junos OS Release 9.6, you can configure authentication for Bidirectional Forwarding Detection (BFD) sessions running over Protocol Independent Multicast (PIM). Routing instances are also supported.

Related Documentation

- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 357
- Configuring BFD for PIM on page 194
- *authentication-key-chains*
- bfd-liveness-detection on page 984
- *show bfd session*

CHAPTER 8

Routing Content to Densely Clustered Receivers with PIM Dense Mode

Understanding PIM Dense Mode

PIM dense mode is less sophisticated than PIM sparse mode. PIM dense mode is useful for multicast LAN applications, the main environment for all dense mode protocols.

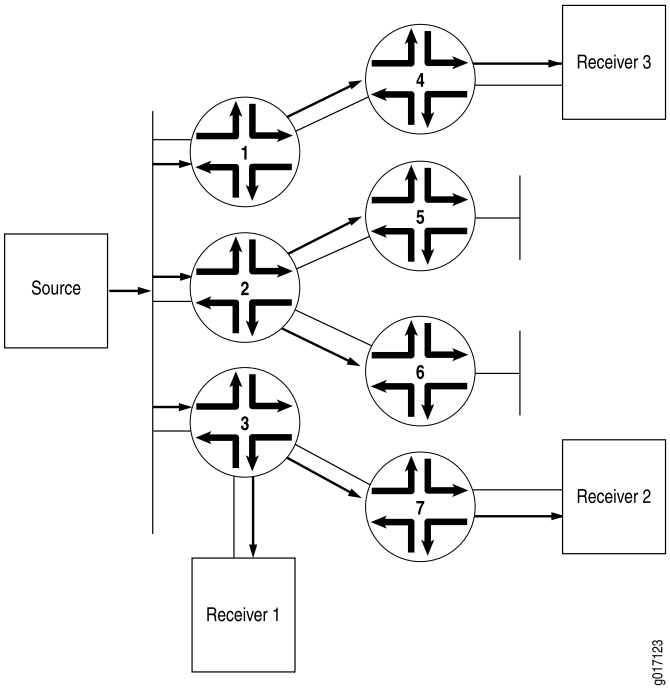
PIM dense mode implements the same flood-and-prune mechanism that DVMRP and other dense mode routing protocols employ. The main difference between DVMRP and PIM dense mode is that PIM dense mode introduces the concept of protocol independence. PIM dense mode can use the routing table populated by any underlying unicast routing protocol to perform reverse-path-forwarding (RPF) checks.

Internet service providers (ISPs) typically appreciate the ability to use any underlying unicast routing protocol with PIM dense mode because they do not need to introduce and manage a separate routing protocol just for RPF checks. While unicast routing protocols extended as multiprotocol BGP (MBGP) and Multitopology Routing in IS-IS (M-IS-IS) were later employed to build special tables to perform RPF checks, PIM dense mode does not require them.

PIM dense mode can use the unicast routing table populated by OSPF, IS-IS, BGP, and so on, or PIM dense mode can be configured to use a special multicast RPF table populated by MBGP or M-IS-IS when performing RPF checks.

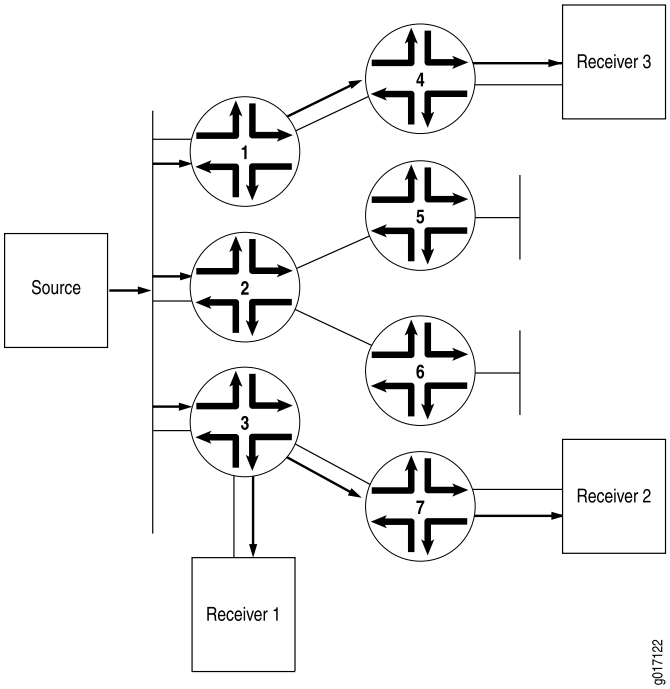
Unlike sparse mode, in which data is forwarded only to routing devices sending an explicit request, dense mode implements a *flood-and-prune* mechanism, similar to DVMRP. In PIM dense mode, there is no RP. A routing device receives the multicast data on the interface closest to the source, then forwards the traffic to all other interfaces (see Figure 25 on page 202).

Figure 25: Multicast Traffic Flooded from the Source Using PIM Dense Mode



Flooding occurs periodically. It is used to refresh state information, such as the source IP address and multicast group pair. If the routing device has no interested receivers for the data, and the OIL becomes empty, the routing device sends a prune message upstream to stop delivery of multicast traffic (see Figure 26 on page 202).

Figure 26: Prune Messages Sent Back to the Source to Stop Unwanted Multicast Traffic



Understanding PIM Sparse-Dense Mode

Sparse-dense mode, as the name implies, allows the interface to operate on a per-group basis in either sparse or dense mode. A group specified as dense is not mapped to an RP. Instead, data packets destined for that group are forwarded by means of PIM dense-mode rules. A group specified as sparse is mapped to an RP, and data packets are forwarded by means of PIM sparse-mode rules.

For information about PIM sparse-mode and PIM dense-mode rules, see "Understanding PIM Sparse Mode" on page 209 and "Understanding PIM Dense Mode" on page 201.

Related Documentation

- Understanding PIM Sparse Mode on page 209
- Understanding PIM Dense Mode on page 201

Mixing PIM Sparse and Dense Modes

It is possible to mix PIM dense mode, PIM sparse mode, and PIM source-specific multicast (SSM) on the same network, the same routing device, and even the same interface. This is because modes are effectively tied to multicast groups, an IP multicast group address must be unique for a particular group's traffic, and scoping limits enforce the division between potential or actual overlaps.

NOTE: **PIM sparse mode was capable of forming shortest-path trees (SPTs) already. Changes to PIM sparse mode to support PIM SSM mainly involved defining behavior in the SSM address range, because shared-tree behavior is prohibited for groups in the SSM address range.**

A multicast routing device employing sparse-dense mode is a good example of mixing PIM modes on the same network or routing device or interface. Dense modes are easy to support because of the flooding, but scaling issues make dense modes inappropriate for Internet use beyond very restricted uses.

Configuring PIM Dense Mode

- Understanding PIM Dense Mode on page 203
- Configuring PIM Dense Mode Properties on page 205

Understanding PIM Dense Mode

PIM dense mode is less sophisticated than PIM sparse mode. PIM dense mode is useful for multicast LAN applications, the main environment for all dense mode protocols.

PIM dense mode implements the same flood-and-prune mechanism that DVMRP and other dense mode routing protocols employ. The main difference between DVMRP and PIM dense mode is that PIM dense mode introduces the concept of protocol

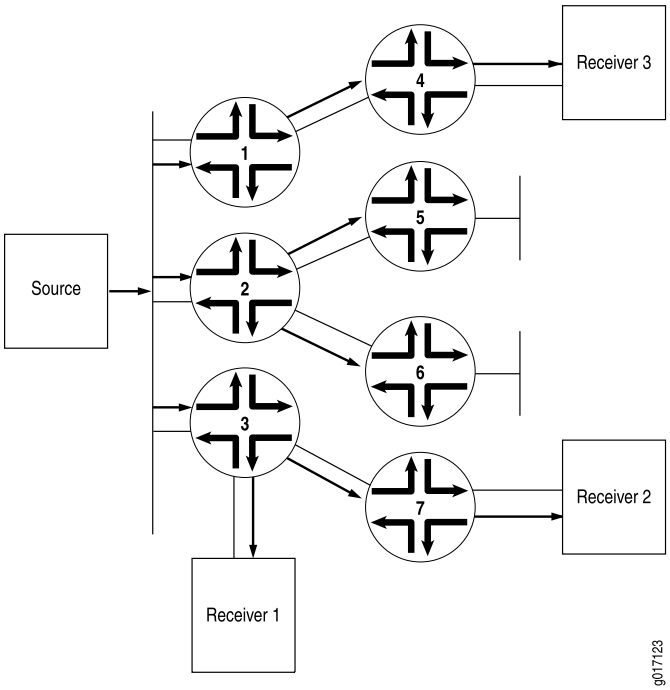
independence. PIM dense mode can use the routing table populated by any underlying unicast routing protocol to perform reverse-path-forwarding (RPF) checks.

Internet service providers (ISPs) typically appreciate the ability to use any underlying unicast routing protocol with PIM dense mode because they do not need to introduce and manage a separate routing protocol just for RPF checks. While unicast routing protocols extended as multiprotocol BGP (MBGP) and Multitopology Routing in IS-IS (M-IS-IS) were later employed to build special tables to perform RPF checks, PIM dense mode does not require them.

PIM dense mode can use the unicast routing table populated by OSPF, IS-IS, BGP, and so on, or PIM dense mode can be configured to use a special multicast RPF table populated by MBGP or M-IS-IS when performing RPF checks.

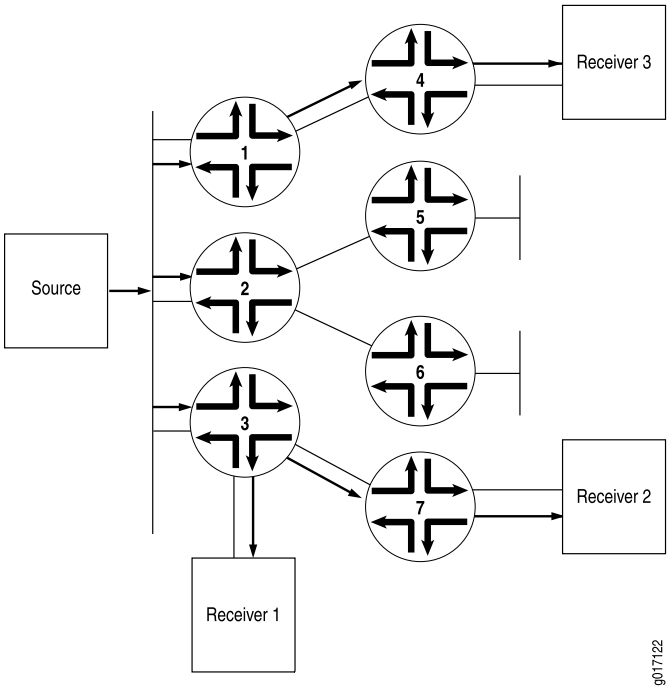
Unlike sparse mode, in which data is forwarded only to routing devices sending an explicit request, dense mode implements a *flood-and-prune* mechanism, similar to DVMRP. In PIM dense mode, there is no RP. A routing device receives the multicast data on the interface closest to the source, then forwards the traffic to all other interfaces (see Figure 25 on page 202).

Figure 27: Multicast Traffic Flooded from the Source Using PIM Dense Mode



Flooding occurs periodically. It is used to refresh state information, such as the source IP address and multicast group pair. If the routing device has no interested receivers for the data, and the OIL becomes empty, the routing device sends a prune message upstream to stop delivery of multicast traffic (see Figure 26 on page 202).

Figure 28: Prune Messages Sent Back to the Source to Stop Unwanted Multicast Traffic



Configuring PIM Dense Mode Properties

In PIM dense mode (PIM-DM), the assumption is that almost all possible subnets have at least one receiver wanting to receive the multicast traffic from a source, so the network is flooded with traffic on all possible branches, then pruned back when branches do not express an interest in receiving the packets, explicitly (by message) or implicitly (time-out silence). LANs are appropriate networks for dense-mode operation.

By default, PIM is disabled. When you enable PIM, it operates in sparse mode by default.

You can configure PIM dense mode globally or for a routing instance. This example shows how to configure the routing instance and how to specify that PIM dense mode use **inet.2** as its RPF routing table instead of **inet.0**.

To configure the router properties for PIM dense mode:

1. (Optional) Create an IPv4 routing table group so that interface routes are installed into two routing tables, **inet.0** and **inet.2**.

   ```
   [edit routing-options rib-groups]
   user@host# set pim-rg export-rib inet.0
   user@host# set pim-rg import-rib [ inet.0 inet.2 ]
   ```

2. (Optional) Associate the routing table group with a PIM routing instance.

   ```
   [edit routing-instances PIM.dense protocols pim]
   user@host# set rib-group inet pim-rg
   ```

3. Configure the PIM interface. If you do not specify any interfaces, PIM is enabled on all router interfaces. Generally, you specify interface names only if you are disabling PIM on certain interfaces.

   ```
   [edit routing-instances PIM.dense protocols pim]
   user@host# set interface fe-0/0/1.0 mode dense
   ```

   NOTE: You cannot configure both PIM and Distance Vector Multicast Routing Protocol (DVMRP) in forwarding mode on the same interface. You can configure PIM on the same interface only if you configured DVMRP in unicast-routing mode.

4. Monitor the operation of PIM dense mode by running the **show pim interfaces**, **show pim join**, **show pim neighbors**, and **show pim statistics** commands.

See Also
- Understanding PIM Dense Mode on page 201
- Example: Configuring a Dedicated PIM RPF Routing Table on page 806

Related Documentation
- Configuring PIM Sparse-Dense Mode on page 206
- *Configuring Basic PIM Settings*

Configuring PIM Sparse-Dense Mode

Understanding PIM Sparse-Dense Mode

Sparse-dense mode, as the name implies, allows the interface to operate on a per-group basis in either sparse or dense mode. A group specified as dense is not mapped to an RP. Instead, data packets destined for that group are forwarded by means of PIM dense-mode rules. A group specified as sparse is mapped to an RP, and data packets are forwarded by means of PIM sparse-mode rules.

For information about PIM sparse-mode and PIM dense-mode rules, see "Understanding PIM Sparse Mode" on page 209 and "Understanding PIM Dense Mode" on page 201.

See Also
- Understanding PIM Sparse Mode on page 209
- Understanding PIM Dense Mode on page 201

Mixing PIM Sparse and Dense Modes

It is possible to mix PIM dense mode, PIM sparse mode, and PIM source-specific multicast (SSM) on the same network, the same routing device, and even the same interface. This is because modes are effectively tied to multicast groups, an IP multicast group address must be unique for a particular group's traffic, and scoping limits enforce the division between potential or actual overlaps.

NOTE: PIM sparse mode was capable of forming shortest-path trees (SPTs) already. Changes to PIM sparse mode to support PIM SSM mainly involved defining behavior in the SSM address range, because shared-tree behavior is prohibited for groups in the SSM address range.

A multicast routing device employing sparse-dense mode is a good example of mixing PIM modes on the same network or routing device or interface. Dense modes are easy to support because of the flooding, but scaling issues make dense modes inappropriate for Internet use beyond very restricted uses.

Configuring PIM Sparse-Dense Mode Properties

Sparse-dense mode allows the interface to operate on a per-group basis in either sparse or dense mode. A group specified as “dense” is not mapped to an RP. Instead, data packets destined for that group are forwarded by means of PIM dense mode rules. A group specified as “sparse” is mapped to an RP, and data packets are forwarded by means of PIM sparse-mode rules. Sparse-dense mode is useful in networks implementing auto-RP for PIM sparse mode.

By default, PIM is disabled. When you enable PIM, it operates in sparse mode by default.

You can configure PIM sparse-dense mode globally or for a routing instance. This example shows how to configure PIM sparse-dense mode globally on all interfaces, specifying that the groups 224.0.1.39 and 224.0.1.40 are using dense mode.

To configure the router properties for PIM sparse-dense mode:

1. Configure the dense-mode groups.

   ```
   [protocols pim]
   user@host# set dense-groups 224.0.1.39
   user@host# set dense-groups 224.0.1.40
   ```

2. Configure all interfaces on the routing device to use sparse-dense mode. When configuring all interfaces, exclude the **fxp0.0** management interface by adding the disable statement for that interface.

   ```
   [edit protocols pim]
   user@host# set interface all mode sparse-dense
   user@host# set interface fxp0.0 disable
   ```

3. Monitor the operation of PIM sparse-dense mode by running the **show pim interfaces**, **show pim join**, **show pim neighbors**, and **show pim statistics** commands.

See Also

- Understanding PIM Sparse-Dense Mode on page 203

Related Documentation

- Configuring PIM Dense Mode on page 203
- *Configuring Basic PIM Settings*

CHAPTER 9

Routing Content to Larger, Sparser Groups with PIM Sparse Mode

Understanding PIM Sparse Mode

A Protocol Independent Multicast (PIM) sparse-mode domain uses reverse-path forwarding (RPF) to create a path from a data source to the receiver requesting the data. When a receiver issues an explicit join request, an RPF check is triggered. A (*,G) PIM join message is sent toward the RP from the receiver's designated router (DR). (By definition, this message is actually called a join/prune message, but for clarity in this description, it is called either join or prune, depending on its context.) The join message is multicast hop by hop upstream to the ALL-PIM-ROUTERS group (224.0.0.13) by means of each router's RPF interface until it reaches the RP. The RP router receives the (*,G) PIM join message and adds the interface on which it was received to the outgoing interface list (OIL) of the rendezvous-point tree (RPT) forwarding state entry. This builds the RPT connecting the receiver with the RP. The RPT remains in effect, even if no active sources generate traffic.

NOTE: State—the (*,G) or (S,G) entries—is the information used for forwarding unicast or multicast packets. S is the source IP address, G is the multicast group address, and * represents any source sending to group G. Routers keep track of the multicast forwarding state for the incoming and outgoing interfaces for each group.

When a source becomes active, the source DR encapsulates multicast data packets into a PIM register message and sends them by means of unicast to the RP router.

If the RP router has interested receivers in the PIM sparse-mode domain, it sends a PIM join message toward the source to build a shortest-path tree (SPT) back to the source. The source sends multicast packets out on the LAN, and the source DR encapsulates the packets in a PIM register message and forwards the message toward the RP router by means of unicast. The RP router receives PIM register messages back from the source, and thus adds a new source to the distribution tree, keeping track of sources in a PIM table. Once an RP router receives packets natively (with S,G), it sends a register stop message to stop receiving the register messages by means of unicast.

In actual application, many receivers with multiple SPTs are involved in a multicast traffic flow. To illustrate the process, we track the multicast traffic from the RP router to one receiver. In such a case, the RP router begins sending multicast packets down the RPT toward the receiver's DR for delivery to the interested receivers. When the receiver's DR receives the first packet from the RPT, the DR sends a PIM join message toward the source DR to start building an SPT back to the source. When the source DR receives the PIM join message from the receiver's DR, it starts sending traffic down all SPTs. When the first multicast packet is received by the receiver's DR, the receiver's DR sends a PIM prune message to the RP router to stop duplicate packets from being sent through the RPT. In turn, the RP router stops sending multicast packets to the receiver's DR, and sends a PIM prune message for this source over the RPT toward the source DR to halt multicast packet delivery to the RP router from that particular source.

If the RP router receives a PIM register message from an active source but has no interested receivers in the PIM sparse-mode domain, it still adds the active source into the PIM table. However, after adding the active source into the PIM table, the RP router sends a register stop message. The RP router discovers the active source's existence and no longer needs to receive advertisement of the source (which utilizes resources).

NOTE: If the number of PIM join messages exceeds the configured MTU, the messages are fragmented in IPv6 PIM sparse mode. To avoid the fragmentation of PIM join messages, the multicast traffic receives the interface MTU instead of the path MTU.

The major characteristics of PIM sparse mode are as follows:

- Routers with downstream receivers join a PIM sparse-mode tree through an explicit join message.
- PIM sparse-mode RPs are the routers where receivers meet sources.

- Senders announce their existence to one or more RPs, and receivers query RPs to find multicast sessions.
- Once receivers get content from sources through the RP, the last-hop router (the router closest to the receiver) can optionally remove the RP from the shared distribution tree (*,G) if the new source-based tree (S,G) is shorter. Receivers can then get content directly from the source.

  The transitional aspect of PIM sparse mode from shared to source-based tree is one of the major features of PIM, because it prevents overloading the RP or surrounding core links.

There are related issues regarding source, RPs, and receivers when sparse mode multicast is used:

- Sources must be able to send to all RPs.
- RPs must all know one another.
- Receivers must send explicit join messages to a known RP.
- Receivers initially need to know only one RP (they later learn about others).
- Receivers can explicitly prune themselves from a tree.
- Receivers that never transition to a source-based tree are effectively running Core Based Trees (CBT).

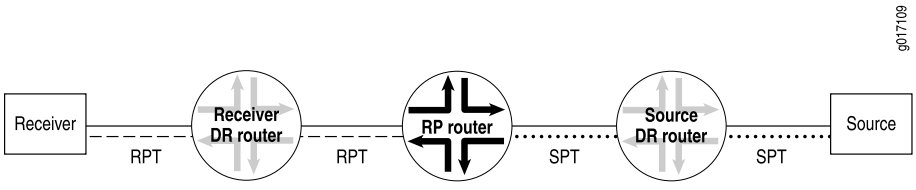
PIM sparse mode has standard features for all of these issues.

Rendezvous Point

The RP router serves as the information exchange point for the other routers. All routers in a PIM domain must provide mapping to an RP router. It is the only router that needs to know the active sources for a domain—the other routers just need to know how to reach the RP. In this way, the RP matches receivers with sources.

The RP router is downstream from the source and forms one end of the shortest-path tree. As shown in Figure 29 on page 211, the RP router is upstream from the receiver and thus forms one end of the rendezvous-point tree.

Figure 29: Rendezvous Point As Part of the RPT and SPT



The benefit of using the RP as the information exchange point is that it reduces the amount of state in non-RP routers. No network flooding is required to provide non-RP routers information about active sources.

RP Mapping Options

RPs can be learned by one of the following mechanisms:

- Static configuration
- Anycast RP
- Auto-RP
- Bootstrap router

We recommend a static RP mapping with anycast RP and a bootstrap router (BSR) with auto-RP configuration, because static mapping provides all the benefits of a bootstrap router and auto-RP without the complexity of the full BSR and auto-RP mechanisms.

Related Documentation

- Understanding Static RP on page 237
- Understanding RP Mapping with Anycast RP on page 244
- Understanding the PIM Bootstrap Router on page 253
- Understanding PIM Auto-RP on page 258

Mixing PIM Sparse and Dense Modes

It is possible to mix PIM dense mode, PIM sparse mode, and PIM source-specific multicast (SSM) on the same network, the same routing device, and even the same interface. This is because modes are effectively tied to multicast groups, an IP multicast group address must be unique for a particular group's traffic, and scoping limits enforce the division between potential or actual overlaps.

NOTE: PIM sparse mode was capable of forming shortest-path trees (SPTs) already. Changes to PIM sparse mode to support PIM SSM mainly involved defining behavior in the SSM address range, because shared-tree behavior is prohibited for groups in the SSM address range.

A multicast routing device employing sparse-dense mode is a good example of mixing PIM modes on the same network or routing device or interface. Dense modes are easy to support because of the flooding, but scaling issues make dense modes inappropriate for Internet use beyond very restricted uses.

Examples: Configuring PIM Sparse Mode

- Understanding PIM Sparse Mode on page 213
- Understanding Designated Routers on page 215
- Tunnel Services PICs and Multicast on page 216
- Enabling PIM Sparse Mode on page 217
- Configuring PIM Join Load Balancing on page 218

Understanding PIM Sparse Mode

A Protocol Independent Multicast (PIM) sparse-mode domain uses reverse-path forwarding (RPF) to create a path from a data source to the receiver requesting the data. When a receiver issues an explicit join request, an RPF check is triggered. A (*,G) PIM join message is sent toward the RP from the receiver's designated router (DR). (By definition, this message is actually called a join/prune message, but for clarity in this description, it is called either join or prune, depending on its context.) The join message is multicast hop by hop upstream to the ALL-PIM-ROUTERS group (224.0.0.13) by means of each router's RPF interface until it reaches the RP. The RP router receives the (*,G) PIM join message and adds the interface on which it was received to the outgoing interface list (OIL) of the rendezvous-point tree (RPT) forwarding state entry. This builds the RPT connecting the receiver with the RP. The RPT remains in effect, even if no active sources generate traffic.

NOTE: State—the (*,G) or (S,G) entries—is the information used for forwarding unicast or multicast packets. S is the source IP address, G is the multicast group address, and * represents any source sending to group G. Routers keep track of the multicast forwarding state for the incoming and outgoing interfaces for each group.

When a source becomes active, the source DR encapsulates multicast data packets into a PIM register message and sends them by means of unicast to the RP router.

If the RP router has interested receivers in the PIM sparse-mode domain, it sends a PIM join message toward the source to build a shortest-path tree (SPT) back to the source. The source sends multicast packets out on the LAN, and the source DR encapsulates the packets in a PIM register message and forwards the message toward the RP router by means of unicast. The RP router receives PIM register messages back from the source, and thus adds a new source to the distribution tree, keeping track of sources in a PIM table. Once an RP router receives packets natively (with S,G), it sends a register stop message to stop receiving the register messages by means of unicast.

In actual application, many receivers with multiple SPTs are involved in a multicast traffic flow. To illustrate the process, we track the multicast traffic from the RP router to one receiver. In such a case, the RP router begins sending multicast packets down the RPT toward the receiver's DR for delivery to the interested receivers. When the receiver's DR receives the first packet from the RPT, the DR sends a PIM join message toward the source DR to start building an SPT back to the source. When the source DR receives the PIM join message from the receiver's DR, it starts sending traffic down all SPTs. When the first multicast packet is received by the receiver's DR, the receiver's DR sends a PIM prune message to the RP router to stop duplicate packets from being sent through the RPT. In turn, the RP router stops sending multicast packets to the receiver's DR, and

sends a PIM prune message for this source over the RPT toward the source DR to halt multicast packet delivery to the RP router from that particular source.

If the RP router receives a PIM register message from an active source but has no interested receivers in the PIM sparse-mode domain, it still adds the active source into the PIM table. However, after adding the active source into the PIM table, the RP router sends a register stop message. The RP router discovers the active source's existence and no longer needs to receive advertisement of the source (which utilizes resources).

NOTE: If the number of PIM join messages exceeds the configured MTU, the messages are fragmented in IPv6 PIM sparse mode. To avoid the fragmentation of PIM join messages, the multicast traffic receives the interface MTU instead of the path MTU.

The major characteristics of PIM sparse mode are as follows:

- Routers with downstream receivers join a PIM sparse-mode tree through an explicit join message.
- PIM sparse-mode RPs are the routers where receivers meet sources.
- Senders announce their existence to one or more RPs, and receivers query RPs to find multicast sessions.
- Once receivers get content from sources through the RP, the last-hop router (the router closest to the receiver) can optionally remove the RP from the shared distribution tree (*,G) if the new source-based tree (S,G) is shorter. Receivers can then get content directly from the source.

  The transitional aspect of PIM sparse mode from shared to source-based tree is one of the major features of PIM, because it prevents overloading the RP or surrounding core links.

There are related issues regarding source, RPs, and receivers when sparse mode multicast is used:

- Sources must be able to send to all RPs.
- RPs must all know one another.
- Receivers must send explicit join messages to a known RP.
- Receivers initially need to know only one RP (they later learn about others).
- Receivers can explicitly prune themselves from a tree.
- Receivers that never transition to a source-based tree are effectively running Core Based Trees (CBT).

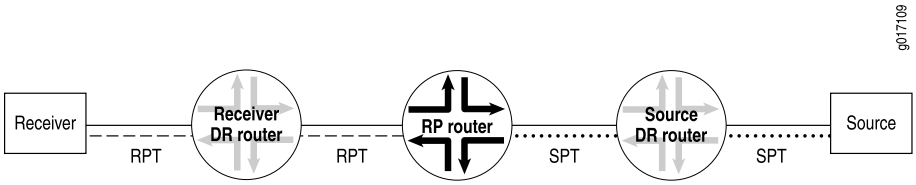
PIM sparse mode has standard features for all of these issues.

Rendezvous Point

The RP router serves as the information exchange point for the other routers. All routers in a PIM domain must provide mapping to an RP router. It is the only router that needs to know the active sources for a domain—the other routers just need to know how to reach the RP. In this way, the RP matches receivers with sources.

The RP router is downstream from the source and forms one end of the shortest-path tree. As shown in Figure 29 on page 211, the RP router is upstream from the receiver and thus forms one end of the rendezvous-point tree.

Figure 30: Rendezvous Point As Part of the RPT and SPT



The benefit of using the RP as the information exchange point is that it reduces the amount of state in non-RP routers. No network flooding is required to provide non-RP routers information about active sources.

RP Mapping Options

RPs can be learned by one of the following mechanisms:

- Static configuration
- Anycast RP
- Auto-RP
- Bootstrap router

We recommend a static RP mapping with anycast RP and a bootstrap router (BSR) with auto-RP configuration, because static mapping provides all the benefits of a bootstrap router and auto-RP without the complexity of the full BSR and auto-RP mechanisms.

See Also

- Understanding Static RP on page 237
- Understanding RP Mapping with Anycast RP on page 244
- Understanding the PIM Bootstrap Router on page 253
- Understanding PIM Auto-RP on page 258

Understanding Designated Routers

In a PIM sparse mode (PIM-SM) domain, there are two types of designated routers to consider:

- The receiver DR sends PIM join and PIM prune messages from the receiver network toward the RP.

- The source DR sends PIM register messages from the source network to the RP.

Neighboring PIM routers multicast periodic PIM hello messages to each other every 30 seconds (the default). The PIM hello message usually includes a holdtime value for the neighbor to use, but this is not a requirement. If the PIM hello message does not include a holdtime value, a default timeout value (in Junos OS, 105 seconds) is used. On receipt of a PIM hello message, a router stores the IP address and priority for that neighbor. If the DR priorities match, the router with the highest IP address is selected as the DR.

If a DR fails, a new one is selected using the same process of comparing IP addresses.

NOTE: In PIM dense mode (PIM-DM), a DR is elected by the same process that PIM-SM uses. However, the only time that a DR has any effect in PIM-DM is when IGMPv1 is used on the interface. (IGMPv2 is the default.) In this case, the DR also functions as the IGMP Query Router because IGMPv1 does not have a Query Router election mechanism.

Tunnel Services PICs and Multicast

On Juniper Networks routers, data packets are encapsulated and de-encapsulated into tunnels by means of hardware and not the software running on the router processor. The hardware used to create tunnel interfaces on M Series and T Series routers is a Tunnel Services PIC. If Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers are configured as rendezvous points or IP version 4 (IPv4) PIM sparse-mode DRs connected to a source, a Tunnel Services PIC is required. Juniper Networks MX Series Ethernet Services Routers do not require Tunnel Services PICs. However, on MX Series routers, you must enable tunnel services with the **tunnel-services** statement on one or more online FPC and PIC combinations at the **[edit chassis fpc *number* pic *number*]** hierarchy level.

CAUTION: For redundancy, we strongly recommend that each routing device has multiple Tunnel Services PICs. In the case of MX Series routers, the recommendation is to configure multiple **tunnel-services** statements.

We also recommend that the Tunnel PICs be installed (or configured) on different FPCs. If you have only one Tunnel PIC or if you have multiple Tunnel PICs installed on a single FPC and then that FPC is removed, the multicast session will not come up. Having redundant Tunnel PICs on separate FPCs can help ensure that at least one Tunnel PIC is available and that multicast will continue working.

On MX Series routers, the redundant configuration looks like the following example:

```
[edit chassis]
user@mx-host# set fpc 1 pic 0 tunnel-services bandwidth 1g
user@mx-host# set fpc 2 pic 0 tunnel-services bandwidth 1g
```

In PIM sparse mode, the source DR takes the initial multicast packets and encapsulates them in PIM register messages. The source DR then unicasts the packets to the PIM sparse-mode RP router, where the PIM register message is de-encapsulated.

When a router is configured as a PIM sparse-mode RP router (by specifying an address using the **address** statement at the **[edit protocols pim rp local]** hierarchy level) and a Tunnel PIC is present on the router, a PIM register de-encapsulation interface, or **pd** interface, is automatically created. The **pd** interface receives PIM register messages and de-encapsulates them by means of the hardware.

If PIM sparse mode is enabled and a Tunnel Services PIC is present on the router, a PIM register encapsulation interface (**pe** interface) is automatically created for each RP address. The **pe** interface is used to encapsulate source data packets and send the packets to RP addresses on the PIM DR and the PIM RP. The **pe** interface receives PIM register messages and encapsulates the packets by means of the hardware.

Do not confuse the configurable **pe** and **pd** hardware interfaces with the nonconfigurable **pime** and **pimd** software interfaces. Both pairs encapsulate and de-encapsulate multicast packets, and are created automatically. However, the **pe** and **pd** interfaces appear only if a Tunnel Services PIC is present. The **pime** and **pimd** interfaces are not useful in situations requiring the **pe** and **pd** interfaces.

If the source DR is the RP, then there is no need for PIM register messages and consequently no need for a Tunnel Services PIC.

When PIM sparse mode is used with IP version 6 (IPv6), a Tunnel PIC is required on the RP, but not on the IPv6 PIM DR. The lack of a Tunnel PIC requirement on the IPv6 DR applies only to IPv6 PIM sparse mode and is not to be confused with IPv4 PIM sparse-mode requirements.

Table 10 on page 217 shows the complete matrix of IPv4 and IPv6 PIM Tunnel PIC requirements.

Table 10: Tunnel PIC Requirements for IPv4 and IPv6 Multicast

IP Version	Tunnel PIC on RP	Tunnel PIC on DR
IPv4	Yes	Yes
IPv6	Yes	No

Enabling PIM Sparse Mode

In PIM sparse mode (PIM-SM), the assumption is that very few of the possible receivers want packets from a source, so the network establishes and sends packets only on branches that have at least one leaf indicating (by message) a desire for the traffic. WANs are appropriate networks for sparse-mode operation.

Starting in Junos OS Release 16.1, PIM is disabled by default. When you enable PIM, it operates in sparse mode by default. You do not need to configure Internet Group

Management Protocol (IGMP) version 2 for a sparse mode configuration. After you enable PIM, by default, IGMP version 2 is also enabled.

Junos OS uses PIM version 2 for both rendezvous point (RP) mode (at the **[edit protocols pim rp static address *address*]** hierarchy level) and interface mode (at the **[edit protocols pim interface *interface-name*]** hierarchy level).

All systems on a subnet must run the same version of PIM.

You can configure PIM sparse mode globally or for a routing instance. This example shows how to configure PIM sparse mode globally on all interfaces. It also shows how to configure a static RP router and how to configure the non-RP routers.

To configure the router properties for PIM sparse mode:

1. Configure the static RP router.

   ```
   [edit protocols pim]
   user@host# set rp local family inet address 192.168.3.253
   ```

2. Configure the RP router interfaces. When configuring all interfaces, exclude the **fxp0.0** management interface by including the **disable** statement for that interface.

   ```
   [edit protocols pim]
   user@host# set interface all mode sparse
   user@host# set interface fxp0.0 disable
   ```

3. Configure the non-RP routers. Include the following configuration on all of the non-RP routers.

   ```
   [edit protocols pim]
   user@host# set rp static address 192.168.3.253
   user@host# set interface all mode sparse
   user@host# set interface fxp0.0 disable
   ```

4. Monitor the operation of PIM sparse mode.

   - show pim interfaces
   - show pim join
   - show pim neighbors
   - show pim rps

See Also

- Understanding PIM Sparse Mode on page 209

Configuring PIM Join Load Balancing

By default, PIM join messages are sent toward a source based on the RPF routing table check. If there is more than one equal-cost path toward the source, then one upstream interface is chosen to send the join message. This interface is also used for all downstream

traffic, so even though there are alternative interfaces available, the multicast load is concentrated on one upstream interface and routing device.

For PIM sparse mode, you can configure PIM join load balancing to spread join messages and traffic across equal-cost upstream paths (interfaces and routing devices) provided by unicast routing toward a source. PIM join load balancing is only supported for PIM sparse mode configurations.

PIM join load balancing is supported on draft-rosen multicast VPNs (also referred to as dual PIM multicast VPNs). PIM join load balancing is not supported on multiprotocol BGP-based multicast VPNs (also referred to as next-generation Layer 3 VPN multicast). When PIM join load balancing is enabled in a draft-rosen Layer 3 VPN scenario, the load balancing is achieved based on the join counts for the far-end PE routing devices, not for any intermediate P routing devices.

If an internal BGP (IBGP) multipath forwarding VPN route is available, the Junos OS uses the multipath forwarding VPN route to send join messages to the remote PE routers to achieve load balancing over the VPN.

By default, when multiple PIM joins are received for different groups, all joins are sent to the same upstream gateway chosen by the unicast routing protocol. Even if there are multiple equal-cost paths available, these alternative paths are not utilized to distribute multicast traffic from the source to the various groups.

When PIM join load balancing is configured, the PIM joins are distributed equally among all equal-cost upstream interfaces and neighbors. Every new join triggers the selection of the least-loaded upstream interface and neighbor. If there are multiple neighbors on the same interface (for example, on a LAN), join load balancing maintains a value for each of the neighbors and distributes multicast joins (and downstream traffic) among these as well.

Join counts for interfaces and neighbors are maintained globally, not on a per-source basis. Therefore, there is no guarantee that joins for a particular source are load-balanced. However, the joins for all sources and all groups known to the routing device are load-balanced. There is also no way to administratively give preference to one neighbor over another: all equal-cost paths are treated the same way.

You can configure message filtering globally or for a routing instance. This example shows the global configuration.

You configure PIM join load balancing on the non-RP routers in the PIM domain.

1. Determine if there are multiple paths available for a source (for example, an RP) with the output of the **show pim join extensive** or **show pim source** commands.

```
user@host> show pim join extensive
Instance: PIM.master Family: INET

Group: 224.1.1.1
    Source: *
    RP: 10.255.245.6
    Flags: sparse,rptree,wildcard
    Upstream interface: t1-0/2/3.0
    Upstream neighbor: 192.168.38.57
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: t1-0/2/1.0
            192.168.38.16 State: JOIN Flags; SRW Timeout: 164
Group: 224.2.127.254
    Source: *
    RP: 10.255.245.6
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/3/0.0
    Upstream neighbor: 192.168.38.47
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: t1-0/2/3.0
            192.168.38.16 State: JOIN Flags; SRW Timeout: 164
```

   Note that for this router, the RP at IP address 10.255.245.6 is the source for two multicast groups: 224.1.1.1 and 224.2.127.254. This router has two equal-cost paths through two different upstream interfaces (**t1-0/2/3.0** and **so-0/3/0.0**) with two different neighbors (192.168.38.57 and 192.168.38.47). This router is a good candidate for PIM join load balancing.

2. On the non-RP router, configure PIM sparse mode and join load balancing.

```
[edit protocols pim ]
user@host# set interface all mode sparse version 2
user@host# set join-load-balance
```

3. Then configure the static address of the RP.

```
[edit protocols pim rp]
user@host# set static address 10.10.10.1
```

4. Monitor the operation.

   If load balancing is enabled for this router, the number of PIM joins sent on each interface is shown in the output for the **show pim interfaces** command.

```
user@host> show pim interfaces
Instance: PIM.master

Name             Stat Mode        IP V State NbrCnt JoinCnt   DR address
lo0.0            Up   Sparse       4 2 DR         0       0   10.255.168.58
```

```
pe-1/2/0.32769   Up   Sparse       4 2 P2P        0       0
so-0/3/0.0       Up   Sparse       4 2 P2P        1       1
t1-0/2/1.0       Up   Sparse       4 2 P2P        1       0
t1-0/2/3.0       Up   Sparse       4 2 P2P        1       1
lo0.0            Up   Sparse       6 2 DR         0       0   fe80::2a0:a5ff:4b7
```

Note that the two equal-cost paths shown by the **show pim interfaces** command now have nonzero join counts. If the counts differ by more than one and were zero (0) when load balancing commenced, an error occurs (joins before load balancing are not redistributed). The join count also appears in the **show pim neighbors detail** output:

```
user@host> show pim neighbors detail
Interface: so-0/3/0.0

    Address: 192.168.38.46, IPv4, PIM v2, Mode: Sparse, Join Count: 0
        Hello Option Holdtime: 65535 seconds
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1689116164
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms

    Address: 192.168.38.47, IPv4, PIM v2, Join Count: 1
        BFD: Disabled
        Hello Option Holdtime: 105 seconds 102 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 792890329
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms

Interface: t1-0/2/3.0

    Address: 192.168.38.56, IPv4, PIM v2, Mode: Sparse, Join Count: 0
        Hello Option Holdtime: 65535 seconds
        Hello Option DR Priority: 1
        Hello Option Generation ID: 678582286
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms

    Address: 192.168.38.57, IPv4, PIM v2, Join Count: 1
        BFD: Disabled
        Hello Option Holdtime: 105 seconds 97 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1854475503
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
```

Note that the join count is nonzero on the two load-balanced interfaces toward the upstream neighbors.

PIM join load balancing only takes effect when the feature is configured. Prior joins are not redistributed to achieve perfect load balancing. In addition, if an interface or neighbor fails, the new joins are redistributed among remaining active interfaces and neighbors. However, when the interface or neighbor is restored, prior joins are not redistributed. The **clear pim join-distribution** command redistributes the existing flows to new or restored upstream neighbors. Redistributing the existing flows causes traffic to be disrupted, so we recommend that you perform PIM join redistribution during a maintenance window.

See Also

- clear pim join-distribution on page 1418
- show pim interfaces on page 1639

- show pim neighbors on page 1657
- show pim source on page 1686

Modifying the Join State Timeout

This section describes how to configure the join state timeout.

A downstream router periodically sends join messages to refresh the join state on the upstream router. If the join state is not refreshed before the timeout expires, the join state is removed.

By default, the join state timeout is 210 seconds. You can change this timeout to allow additional time to receive the join messages. Because the messages are called join-prune messages, the name used is the **join-prune-timeout** statement.

To modify the timeout, include the **join-prune-timeout** statement:

```
user@host# set protocols pim join-prune-timeout 230
```

The join timeout value can be from 210 through 420 seconds.

See Also
- join-prune-timeout on page 1117

Example: Enabling Join Suppression

This example describes how to enable PIM join suppression.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 217.

Overview

PIM join suppression enables a router on a multiaccess network to defer sending join messages to an upstream router when it sees identical join messages on the same network. Eventually, only one router sends these join messages, and the other routers suppress identical messages. Limiting the number of join messages improves scalability and efficiency by reducing the number of messages sent to the same router.

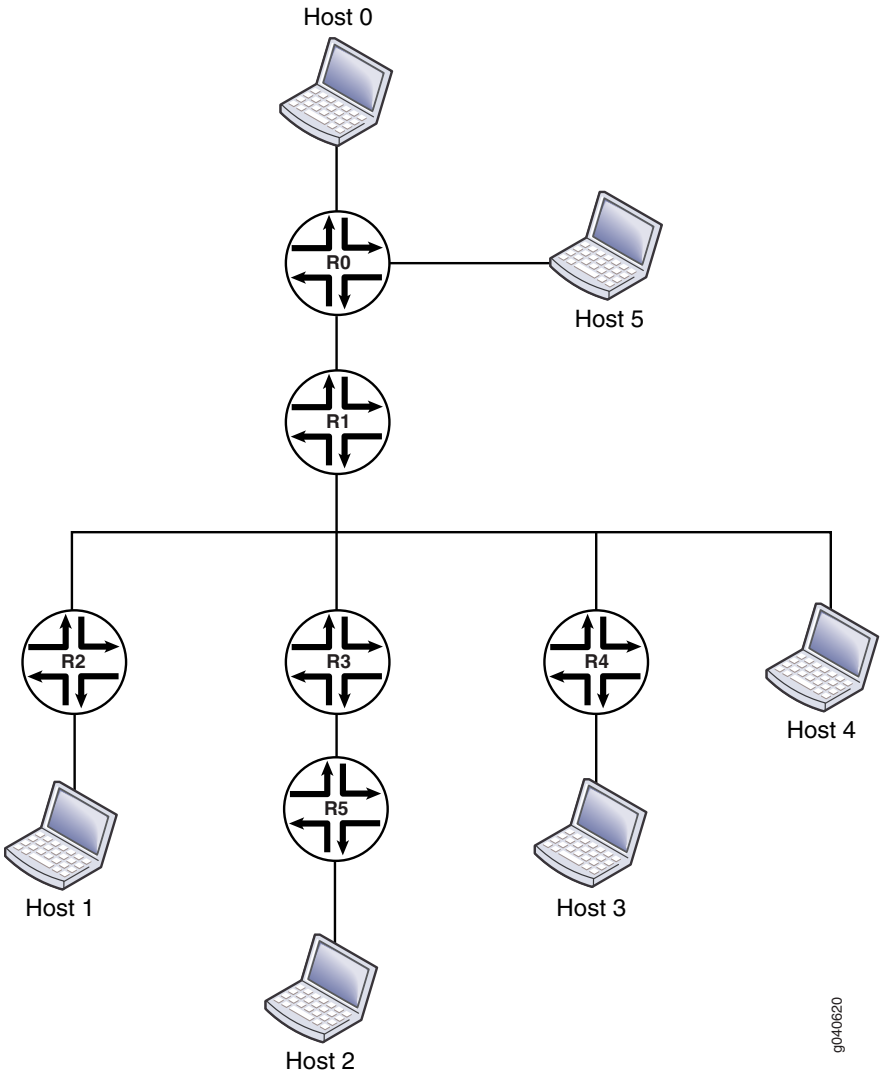
This example includes the following statements:

- **override-interval**—Sets the maximum time in milliseconds to delay sending override join messages. When a router sees a prune message for a join it is currently suppressing, it waits before it sends an override join message. Waiting helps avoid multiple downstream routers sending override join messages at the same time. The override interval is a random timer with a value of 0 through the maximum override value.
- **propagation-delay**—Sets a value in milliseconds for a prune pending timer, which specifies how long to wait before executing a prune on an upstream router. During this period, the router waits for any prune override join messages that might be currently suppressed. The period for the prune pending timer is the sum of the **override-interval** value and the value specified for **propagation-delay**.
- **reset-tracking-bit**—Enables PIM join suppression on each multiaccess downstream interface. This statement resets a tracking bit field (T-bit) on the LAN prune delay hello option from the default of 1 (join suppression disabled) to 0 (join suppression enabled).

  When multiple identical join messages are received, a random join suppression timer is activated, with a range of 66 through 84 milliseconds. The timer is reset each time join suppression is triggered.

Figure 31 on page 224 shows the topology used in this example.

Figure 31: Join Suppression



The items in the figure represent the following functions:

- Host 0 is the multicast source.
- Host 1, Host 2, Host 3, and Host 4 are receivers.
- Router R0 is the first-hop router and the RP.
- Router R1 is an upstream router.
- Routers R2, R3, R4, and R5 are downstream routers in the multicast LAN.

This example shows the configuration of the downstream devices: Routers R2, R3, R4, and R5.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set protocols pim traceoptions file pim.log
set protocols pim traceoptions file size 5m
set protocols pim traceoptions file world-readable
set protocols pim traceoptions flag join detail
set protocols pim traceoptions flag prune detail
set protocols pim traceoptions flag normal detail
set protocols pim traceoptions flag register detail
set protocols pim rp static address 10.255.112.160
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim interface fxp0.0 disable
set protocols pim reset-tracking-bit
set protocols pim propagation-delay 500
set protocols pim override-interval 4000
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure PIM join suppression on a non-RP downstream router in the multicast LAN:

1. Configure PIM sparse mode on the interfaces.

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp static address 10.255.112.160
[edit protocols pim]
user@host# set interface all mode sparse version 2
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
```

2. Enable the join suppression timer.

```
[edit protocols pim]
user@host# set reset-tracking-bit
```

3. Configure the prune override interval value.

```
[edit protocols pim]
user@host# set override-interval 4000
```

4. Configure the propagation delay of the link.

```
[edit protocols pim]
user@host# set propagation-delay 500
```

5. (Optional) Configure PIM tracing operations.

```
[edit protocols pim]
user@host# set traceoptions file pim.log size 5m world-readable
[edit protocols pim]
user@host# set traceoptions flag join detail
[edit protocols pim]
user@host# set traceoptions flag normal detail
[edit protocols pim]
user@host# set traceoptions flag register detail
```

6. If you are done configuring the device, commit the configuration.

```
[edit protocols pim]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show protocols
pim {
    traceoptions {
        file pim.log size 5m world-readable;
        flag join detail;
        flag prune detail;
        flag normal detail;
        flag register detail;
    }
    rp {
        static {
            address 10.255.112.160;
        }
    }
    interface all {
        mode sparse;
        version 2;
    }
    interface fxp0.0 {
        disable;
    }
    reset-tracking-bit;
    propagation-delay 500;
    override-interval 4000;
}
```

Verification

To verify the configuration, run the following commands on the upstream and downstream routers:

- show pim join **extensive**
- show multicast route **extensive**

See Also

- Example: Configuring the PIM Assert Timeout on page 291
- Example: Configuring PIM RPF Selection on page 816
- Example: Configuring the PIM SPT Threshold Policy on page 293
- Enabling PIM Sparse Mode on page 217
- PIM Overview on page 185

Example: Configuring PIM Sparse Mode over an IPsec VPN

IPsec VPNs create secure point-to-point connections between sites over the Internet. The Junos OS implementation of IPsec VPNs supports multicast and unicast traffic. The following example shows how to configure PIM sparse mode for the multicast solution and how to configure IPsec to secure your traffic.

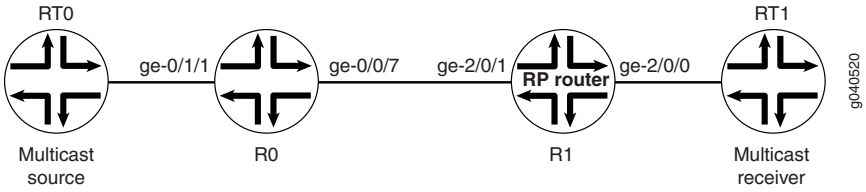
The configuration shown in this example works on the following platforms:

- M Series and T Series routers with one of the following PICs:
  - Adaptive Services (AS) PIC
  - Multiservices (MS) PIC
- JCS1200 platform with a Multiservices PIC (MS-500)

The tunnel endpoints do not need to be the same platform type. For example, the device on one end of the tunnel can be a JCS1200 router, while the device on the other end can be a standalone T Series router. The two routers that are the tunnel endpoints can be in the same autonomous system or in different autonomous systems.

In the configuration shown in this example, OSPF is configured between the tunnel endpoints. In Figure 32 on page 227, the tunnel endpoints are R0 and R1. The network that contains the multicast source is connected to R0. The network that contains the multicast receivers is connected to R1. R1 serves as the statically configured rendezvous point (RP).

Figure 32: PIM Sparse Mode over an IPsec VPN



To configure PIM sparse mode with IPsec:

1. On R0, configure the incoming Gigabit Ethernet interface.

   ```
   [edit interfaces]
   user@host# set ge-0/1/1 description "incoming interface"
   user@host# set ge-0/1/1 unit 0 family inet address 10.20.0.1/30
   ```

2. On R0, configure the outgoing Gigabit Ethernet interface.

   ```
   [edit interfaces]
   user@host# set ge-0/0/7 description "outgoing interface"
   user@host# set ge-0/0/7 unit 0 family inet address 10.10.1.1/30
   ```

3. On R0, configure unit 0 on the **sp-** interface. The Junos OS uses unit 0 for service logging and other communication from the services PIC.

   ```
   [edit interfaces]
   user@host# set sp-0/2/0 unit 0 family inet
   ```

4. On R0, configure the logical interfaces that participate in the IPsec services. In this example, unit 1 is the inward-facing interface. Unit 1001 is the interface that faces the remote IPsec site.

   ```
   [edit interfaces]
   user@host# set sp-0/2/0 unit 1 family inet
   user@host# set sp-0/2/0 unit 1 service-domain inside
   user@host# set sp-0/2/0 unit 1001 family inet
   user@host# set sp-0/2/0 unit 1001 service-domain outside
   ```

5. On R0, direct OSPF traffic into the IPsec tunnel.

   ```
   [edit protocols ospf]
   user@host# set area 0.0.0.0 interface sp-0/2/0.1
   user@host# set parea 0.0.0.0 interface ge-0/1/1.0 passive
   user@host# set area 0.0.0.0 interface lo0.0
   ```

6. On R0, configure PIM sparse mode. This example uses static RP configuration. Because R0 is a non-RP router, configure the address of the RP router, which is the routable address assigned to the loopback interface on R1.

   ```
   [edit protocols pim]
   user@host# set rp static address 10.255.0.156
   user@host# set interfaces sp-0/2/0.1
   user@host# set interfaces ge-0/1/1.0
   user@host# set interfaces lo0.0
   ```

7. On R0, create a rule for a bidirectional dynamic IKE security association (SA) that references the IKE policy and the IPsec policy.

   ```
   [edit services ipsec-vpn rule ipsec_rule]
   user@host# set term ipsec_dynamic then remote-gateway 10.10.1.2
   user@host# set term ipsec_dynamic then dynamic ike-policy ike_policy
   user@host# set term ipsec_dynamic then dynamic ipsec-policy ipsec_policy
   ```



```
user@host# set match-direction input
```

8. On R0, configure the IPsec proposal. This example uses the Authentication Header (AH) Protocol.

```
[edit services ipsec-vpn ipsec proposal ipsec_prop]
user@host# set protocol ah
user@host# set authentication-algorithm hmac-md5-96
```

9. On R0, define the IPsec policy.

```
[edit services ipsec-vpn ipsec policy ipsec_policy]
user@host# set perfect-forward-secrecy keys group1
user@host# set proposal ipsec_prop
```

10. On R0, configure IKE authentication and encryption details.

```
[edit services ipsec-vpn ike proposal ike_prop]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group1
user@host# set authentication-algorithm md5
user@host# set authentication-algorithm 3des-cbc
```

11. On R0, define the IKE policy.

```
[edit services ipsec-vpn ike policy ike_policy]
user@host# set proposals ike_prop
user@host# set pre-shared-key ascii-text "$ABC123"
```

12. On R0, create a service set that defines IPsec-specific information. The first command associates the IKE SA rule with IPsec. The second command defines the address of the local end of the IPsec security tunnel. The last two commands configure the logical interfaces that participate in the IPsec services. Unit 1 is for the IPsec inward-facing traffic. Unit 1001 is for the IPsec outward-facing traffic.

```
[edit services service-set ipsec_svc]
user@host# set ipsec-vpn-rules ipsec_rule
user@host# set ipsec-vpn-options local-gateway 10.10.1.1
user@host# set next-hop-service inside-service-interface sp-0/2/0.1
user@host# set next-hop-service outside-service-interface sp-0/2/0.1001
```

13. On R1, configure the incoming Gigabit Ethernet interface.

```
[edit interfaces]
user@host# set ge-2/0/1 description "incoming interface"
user@host# set ge-2/0/1 unit 0 family inet address 10.10.1.2/30
```

14. On R1, configure the outgoing Gigabit Ethernet interface.

```
[edit interfaces]
user@host# set ge-2/0/0 description "outgoing interface"
user@host# set ge-2/0/0 unit 0 family inet address 10.20.0.5/30
```

15. On R1, configure the loopback interface.

    ```
    [edit interfaces]
    user@host# set lo0.0 family inet address 10.255.0.156
    ```

16. On R1, configure unit 0 on the **sp-** interface. The Junos OS uses unit 0 for service logging and other communication from the services PIC.

    ```
    [edit interfacesinterfaces]
    user@host# set sp-2/1/0 unit 0 family inet
    ```

17. On R1, configure the logical interfaces that participate in the IPsec services. In this example, unit 1 is the inward-facing interface. Unit 1001 is the interface that faces the remote IPsec site.

    ```
    [edit interfaces]
    user@host# set sp-2/1/0 unit 1 family inet
    user@host# set sp-2/1/0 unit 1 service-domain inside
    user@host# set sp-2/1/0 unit 1001 family inet
    user@host# set sp-2/1/0 unit 1001 service-domain outside
    ```

18. On R1, direct OSPF traffic into the IPsec tunnel.

    ```
    [edit protocols ospf]
    user@host# set area 0.0.0.0 interface sp-2/1/0.1
    user@host# set area 0.0.0.0 interface ge-2/0/0.0 passive
    user@host# set area 0.0.0.0 interface lo0.0
    ```

19. On R1, configure PIM sparse mode. R1 is an RP router. When you configure the local RP address, use the shared address, which is the address of R1's loopback interface.

    ```
    [edit protocols pim]
    user@host# set rp local address 10.255.0.156
    user@host# set interface sp-2/1/0.1
    user@host# set interface ge-2/0/0.0
    user@host# set interface lo0.0 family inet
    ```

20. On R1, create a rule for a bidirectional dynamic Internet Key Exchange (IKE) security association (SA) that references the IKE policy and the IPsec policy.

    ```
    [edit services ipsec-vpn rule ipsec_rule]
    user@host# set term ipsec_dynamic from source-address 192.168.195.34/32
    user@host# set term ipsec_dynamic then remote-gateway 10.10.1.1
    user@host# set term ipsec_dynamic then dynamic ike-policy ike_policy
    user@host# set term ipsec_dynamic then dynamic ipsec-policy ipsec_policy
    user@host# set match-direction input
    ```

21. On R1, define the IPsec proposal for the dynamic SA.

    ```
    [edit services ipsec-vpn ipsec proposal ipsec_prop]
    user@host# set protocol ah
    user@host# set authentication-algorithm hmac-md5-96
    ```

22. On R1, define the IPsec policy.

```
[edit services ipsec-vpn ipsec policy ipsec_policy]
user@host# set perfect-forward-secrecy keys group1
user@host# set proposal ipsec_prop
```

23. On R1, configure IKE authentication and encryption details.

```
[edit services ipsec-vpn ike proposal ike_prop]
user@host# set authentication-method pre-shared-keys
user@host# set dh-group group1
user@host# set authentication-algorithm md5
user@host# set authentication-algorithm 3des-cbc
```

24. On R0, define the IKE policy.

```
[edit services ipsec-vpn ike policy ike_policy]
user@host# set proposal ike_prop
user@host# set pre-shared-key ascii-text "$ABC123"
```

25. On R1, create a service set that defines IPsec-specific information. The first command associates the IKE SA rule with IPsec. The second command defines the address of the local end of the IPsec security tunnel. The last two commands configure the logical interfaces that participate in the IPsec services. Unit 1 is for the IPsec inward-facing traffic. Unit 1001 is for the IPsec outward-facing traffic.

```
[edit services service-set ipsec_svc]
user@host# set ipsec-vpn-rules ipsec_rule
user@host# set ipsec-vpn-options local-gateway 10.10.1.2
user@host# set next-hop-service inside-service-interface sp-2/1/0.1
user@host# set next-hop-service outside-service-interface sp-2/1/0.1001
```

To verify the configuration, run the following commands:

Check which RPs the various routers have learned about.

```
user@host> show pim rps extensive inet
```

Check that the IPsec SA negotiation is successful.

```
user@host> show services ipsec-vpn ipsec security-associations
```

Check that the IKE SA negotiation is successful.

```
user@host> show services ipsec-vpn ike security-associations
```

Check that traffic is traveling over the IPsec tunnel.

```
user@host> show services ipsec-vpn ipsec statistics
```

See Also

- Understanding PIM Sparse Mode on page 209
- *Security Services Administration Guide*
- show pim rps on page 1678 in the CLI Explorer

- *show services ipsec-vpn ipsec statistics* in the CLI Explorer
- *show services ipsec-vpn ike security-associations* in the CLI Explorer
- *show services ipsec-vpn ipsec security-associations* in the CLI Explorer

Example: Configuring Multicast for Virtual Routers with IPv6 Interfaces

A virtual router is a type of simplified routing instance that has a single routing table. This example shows how to configure PIM in a virtual router.

Requirements

Before you begin, configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.

Overview

You can configure PIM for the **virtual-router** instance type as well as for the **vrf** instance type. The **virtual-router** instance type is similar to the **vrf** instance type used with Layer 3 VPNs, except that it is used for non-VPN-related applications.

The **virtual-router** instance type has no VPN routing and forwarding (VRF) import, VRF export, VRF target, or route distinguisher requirements. The **virtual-router** instance type is used for non-Layer 3 VPN situations.

When PIM is configured under the **virtual-router** instance type, the VPN configuration is not based on RFC 2547, *BGP/MPLS VPNs*, so PIM operation does not comply with the Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP VPNs*. In the **virtual-router** instance type, PIM operates in a routing instance by itself, forming adjacencies with PIM neighbors over the routing instance interfaces as the other routing protocols do with neighbors in the routing instance.

This example includes the following general steps:

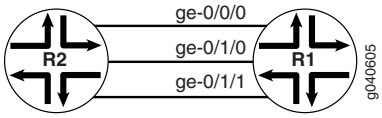
1. On R1, configure a virtual router instance with three interfaces (**ge-0/0/0.0**, **ge-0/1/0.0**, and **ge-0/1/1.0**).
2. Configure PIM and the RP.
3. Configure an MLD static group containing interfaces **ge-0/1/0.0** and **ge-0/1/1.0**.

After you configure this example, you should be able to send multicast traffic from R2 through **ge-0/0/0** on R1 to the static group and verify that the traffic egresses from **ge-0/1/0.0** and **ge-0/1/1.0**.

NOTE: Do not include the **group-address** statement for the **virtual-router** instance type.

Figure 33 on page 233 shows the topology for this example.

Figure 33: Virtual Router Instance with Three Interfaces



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set interfaces ge-0/0/0 unit 0 family inet6 address 2001:4:4:4::1/64
set interfaces ge-0/1/0 unit 0 family inet6 address 2001:24:24:24::1/64
set interfaces ge-0/1/1 unit 0 family inet6 address 2001:7:7:7::1/64
set protocols mld interface ge-0/1/0.0 static group ff0e::10
set protocols mld interface ge-0/1/1.0 static group ff0e::10
set routing-instances mvrf1 instance-type virtual-router
set routing-instances mvrf1 interface ge-0/0/0.0
set routing-instances mvrf1 interface ge-0/1/0.0
set routing-instances mvrf1 interface ge-0/1/1.0
set routing-instances mvrf1 protocols pim rp local family inet6 address 2001:1:1:1::1
set routing-instances mvrf1 protocols pim interface ge-0/0/0.0
set routing-instances mvrf1 protocols pim interface ge-0/1/0.0
set routing-instances mvrf1 protocols pim interface ge-0/1/1.0
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure multicast for virtual routers:

1. Configure the interfaces.

```
[edit]
user@host# edit interfaces
[edit interfaces]
user@host# set ge-0/0/0 unit 0 family inet6 address 2001:4:4:4::1/64
[edit interfaces]
user@host# set ge-0/1/0 unit 0 family inet6 address 2001:24:24:24::1/64
[edit interfaces]
user@host# set ge-0/1/1 unit 0 family inet6 address 2001:7:7:7::1/64
[edit interfaces]
user@host# exit
```

2. Configure the routing instance type.

```
[edit]
user@host# edit routing-instances
[edit routing-instances]
user@host# set mvrf1 instance-type virtual-router
```

3. Configure the interfaces in the routing instance.

```
[edit routing-instances]
user@host# set mvrf1 interface ge-0/0/0
[edit routing-instances]
user@host# set mvrf1 interface ge-0/1/0
[edit routing-instances]
user@host# set mvrf1 interface ge-0/1/1
```

4. Configure PIM and the RP in the routing instance.

```
[edit routing-instances]
user@host# set mvrf1 protocols pim rp local family inet6 address 2001:1:1:1::1
```

5. Configure PIM on the interfaces.

```
[edit routing-instances]
user@host# set mvrf1 protocols pim interface ge-0/0/0
[edit routing-instances]
user@host# set mvrf1 protocols pim interface ge-0/1/0
[edit routing-instances]
user@host# set mvrf1 protocols pim interface ge-0/1/1
[edit routing-instances]
user@host# exit
```

6. Configure the MLD group.

```
[edit]
user@host# edit protocols mld
[edit protocols mld]
user@host# set interface ge-0/1/0.0 static group ff0e::10
[edit protocols mld]
user@host# set interface ge-0/1/1.0 static group ff0e::10
```

7. If you are done configuring the device, commit the configuration.

```
[edit routing-instances]
user@host# commit
```

Results

Confirm your configuration by entering the **show interfaces**, **show routing-instances**, and **show protocols** commands.

```
user@host# show interfaces
```

```
ge-0/0/0 {
    unit 0 {
        family inet6 {
            address 2001:4:4:4::1/64;
        }
    }
}
    ge-0/1/0 {
        unit 0 {
            family inet6 {
                address 2001:24:24:24::1/64;
            }
        }
    }
    ge-0/1/1 {
        unit 0 {
            family inet6 {
                address 2001:7:7:7::1/64;
            }
        }
    }
```

```
user@host# show routing-instances
mvrf1 {
    instance-type virtual-router;
    interface ge-0/0/0.0;
    interface ge-0/1/0.0;
    interface ge-0/1/1.0;
    protocols {
        pim {
            rp {
                local {
                    family inet6 {
                        address 2001:1:1:1::1;
                    }
                }
            }
            interface ge-0/0/0.0;
            interface ge-0/1/0.0;
            interface ge-0/1/1.0;
        }
    }
}
```

```
user@host# show protocols
mld {
    interface ge-0/1/0.0 {
        static {
            group ff0e::10;
        }
    }
    interface ge-0/1/1.0 {
        static {
            group ff0e::10;
        }
    }
}
```

Verification

To verify the configuration, run the following commands:

- show mld group
- show mld interface
- show mld statistics
- show multicast interface
- show multicast route
- show multicast rpf
- show pim interfaces
- show pim join
- show pim neighbors
- show route forwarding-table
- show route instance
- show route table

See Also

- *Configuring Virtual-Router Routing Instances in VPNs* in the *Junos OS VPNs Library for Routing Devices*
- *Types of VPNs* in the *Junos OS VPNs Library for Routing Devices*

Release History Table

Release	Description
16.1	Starting in Junos OS Release 16.1, PIM is disabled by default. When you enable PIM, it operates in sparse mode by default.

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring PIM Dense Mode on page 203
- Configuring a Designated Router for PIM on page 301
- Configuring PIM Filtering on page 267
- Example: Configuring Nonstop Active Routing for PIM on page 371
- Examples: Configuring PIM RPT and SPT Cutover on page 282
- Configuring PIM Sparse-Dense Mode on page 206
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 357
- *Configuring Basic PIM Settings*

Configuring Static RP

Understanding Static RP

Protocol Independent Multicast (PIM) sparse mode is the most common multicast protocol used on the Internet. PIM sparse mode is the default mode whenever PIM is configured on any interface of the device. However, because PIM must not be configured on the network management interface, you must disable it on that interface.

Each any-source multicast (ASM) group has a shared tree through which receivers learn about new multicast sources and new receivers learn about all multicast sources. The rendezvous point (RP) router is the root of this shared tree and receives the multicast traffic from the source. To receive multicast traffic from the groups served by the RP, the device must determine the IP address of the RP for the source.

You can configure a static rendezvous point (RP) configuration that is similar to static routes. A static configuration has the benefit of operating in PIM version 1 or version 2. When you configure the static RP, the RP address that you select for a particular group must be consistent across all routers in a multicast domain.

Starting in Junos OS Release 15.2, the static configuration uses PIM version 2 by default, which is the only version supported in that release and beyond..

One common way for the device to locate RPs is by static configuration of the IP address of the RP. A static configuration is simple and convenient. However, if the statically defined RP router becomes unreachable, there is no automatic failover to another RP router. To remedy this problem, you can use anycast RP.

See Also

- Configuring Local PIM RPs on page 237
- Configuring the Static PIM RP Address on the Non-RP Routing Device on page 242

Configuring Local PIM RPs

Local RP configuration makes the routing device a statically defined RP. Consider statically defining an RP if the network does not have many different RPs defined or if the RP assignment does not change very often. The Junos IPv6 PIM implementation supports only static RP configuration. Automatic RP announcement and bootstrap routers are not available with IPv6.

You can configure a local RP globally or for a routing instance. This example shows how to configure a local RP in a routing instance for IPv4 or IPv6.

To configure the routing device's RP properties:

1. Configure the routing instance as the local RP.

   ```
   [routing-instances VPN-A protocols pim]
   user@host# set rp local
   ```

2. Configure the IP protocol family and IP address.

   IPv6 PIM hello messages are sent to every interface on which you configure **family inet6**, whether at the PIM level of the hierarchy or not. As a result, if you configure an interface with both **family inet** at the **[edit interface *interface-name*]** hierarchy level and **family inet6** at the **[edit protocols pim interface *interface-name*]** hierarchy level, PIM sends both IPv4 and IPv6 hellos to that interface.

   By default, PIM operates in sparse mode on an interface. If you explicitly configure sparse mode, PIM uses this setting for all IPv6 multicast groups. However, if you configure sparse-dense mode, PIM does not accept IPv6 multicast groups as dense groups and operates in sparse mode over them.

   ```
   [edit routing-instances VPN-A protocols pim rp local]
   user@host# set family inet6 address 2001:db8:85a3::8a2e:370:7334
   user@host# set family inet address 10.1.2.254
   ```

3. (IPv4 only) Configure the routing device's RP priority.

   NOTE: The priority statement is not supported for IPv6, but is included here for informational purposes. The routing device's priority value for becoming the RP is included in the bootstrap messages that the routing device sends. Use a smaller number to increase the likelihood that the routing device becomes the RP for local multicast groups. Each PIM routing device uses the priority value and other factors to determine the candidate RPs for a particular group range. After the set of candidate RPs is distributed, each routing device determines algorithmically the RP from the candidate RP set using a hash function. By default, the priority value is set to 1. If this value is set to 0, the bootstrap router can override the group range being advertised by the candidate RP.

   ```
   [edit routing-instances VPN-A protocols pim rp local]
   user@host# set priority 5
   ```

4. Configure the groups for which the routing device is the RP.

   By default, a routing device running PIM is eligible to be the RP for all IPv4 or IPv6 groups (224.0.0.0/4 or FF70::/12 to FFF0::/12). The following example limits the groups for which this routing device can be the RP.

   ```
   [edit routing-instances VPN-A protocols pim rp local]
   ```

```
user@host# set group-ranges fec0::/10
user@host# set group-ranges 10.1.2.0/24
```

5. (IPv4 only) Modify the local RP hold time.

   If the local routing device is configured as an RP, it is considered a candidate RP for its local multicast groups. For candidate RPs, the hold time is used by the bootstrap router to time out RPs, and applies to the bootstrap RP-set mechanism. The RP hold time is part of the candidate RP advertisement message sent by the local routing device to the bootstrap router. If the bootstrap router does not receive a candidate RP advertisement from an RP within the hold time, it removes that routing device from its list of candidate RPs. The default hold time is 150 seconds.

   ```
   [edit routing-instances VPN-A protocols pim rp local]
   user@host# set hold-time 200
   ```

6. (Optional) Override dynamic RP for the specified group address range.

   If you configure both static RP mapping and dynamic RP mapping (such as auto-RP) in a single routing instance, allow the static mapping to take precedence for the given static RP group range, and allow dynamic RP mapping for all other groups.

   If you exclude this statement from the configuration and you use both static and dynamic RP mechanisms for different group ranges within the same routing instance, the dynamic RP mapping takes precedence over the static RP mapping, even if static RP is defined for a specific group range.

   ```
   [edit routing-instances VPN-A protocols pim rp local]
   user@host# set override
   ```

7. Monitor the operation of PIM by running the **show pim** commands. Run **show pim ?** to display the supported commands.

See Also

- PIM Overview on page 185
- Understanding MLD on page 51

Example: Configuring PIM Sparse Mode and RP Static IP Addresses

This example shows how to configure PIM sparse mode and RP static IP addresses.

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements.
8. Configure IGMP.

Overview

In this example, you set the interface value to **all** and disable the **ge-0/0/0** interface. Then you configure the IP address of the RP as **192.168.14.27**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level and then enter **commit** from configuration mode.

```
set protocols pim interface all
set protocols pim interface ge-0/0/0 disable
set protocols pim rp static address 192.168.14.27
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure PIM sparse mode and the RP static IP address:

1. Configure PIM.

   ```
   [edit]
   user@host# edit protocols pim
   ```

2. Set the interface value.

   ```
   [edit protocols pim]
   ```

```
user@host# set pim interface all
```

3. Disable PIM on the network management interface.

```
[edit protocols pim interface]
user@host# set pim interface ge-0/0/0 unit 0 disable
```

4. Configure RP.

```
[edit]
user@host# edit protocols pim rp
```

5. Configure the IP address of the RP.

```
[edit]
user@host# set static address 192.168.14.27
```

Results From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show protocols
  pim {
  rp {
  static {
  address 192.168.14.27;
  }
}
interface all;
  interface ge-0/0/0.0 {
  disable;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 241
- Verifying the IGMP Version on page 242
- Verifying the PIM Mode and Interface Configuration on page 242

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

See Also
- *PIM Configuration Statements* in the *Multicast Protocols Feature Guide*
- Configuring the Static PIM RP Address on the Non-RP Routing Device on page 242 in the *Multicast Protocols Feature Guide*
- Multicast Configuration Overview on page 16
- *Verifying a Multicast Configuration*

Configuring the Static PIM RP Address on the Non-RP Routing Device

Consider statically defining an RP if the network does not have many different RPs defined or if the RP assignment does not change very often. The Junos IPv6 PIM implementation supports only static RP configuration. Automatic RP announcement and bootstrap routers are not available with IPv6.

You configure a static RP address on the non-RP routing device. This enables the non-RP routing device to recognize the local statically defined RP. For example, if R0 is a non-RP router and R1 is the local RP router, you configure R0 with the static RP address of R1. The static IP address is the routable address assigned to the loopback interface on R1. In the following example, the loopback address of the RP is 2001:db8:85a3::8a2e:370:7334.

Starting in Junos OS Release 15.2, the default PIM version is version 2, and version 1 is not supported.

For Junsos OS Release 15.1 and earlier, the default PIM version can be version 1 or version 2, depending on the mode you are configuring. PIM version 1 is the default for RP mode (**[edit pim rp static address *address*]**). PIM version 2 is the default for interface mode (**[edit pim interface *interface-name*]**). An explicitly configured PIM version will override the default setting.

You can configure a static RP address globally or for a routing instance. This example shows how to configure a static RP address in a routing instance for IPv6.

To configure the static RP address:

1. On a non-RP routing device, configure the routing instance to point to the routable address assigned to the loopback interface of the RP.

   ```
   [routing-instances VPN-A protocols pim rp]
   user@host# set static address 2001:db8:85a3::8a2e:370:7334
   ```

   NOTE: Logical systems are also supported. You can configure a static RP address in a logical system only if the logical system is not directly connected to a source.

2. (Optional) Set the PIM sparse mode version.

   For each static RP address, you can optionally specify the PIM version. For Junos OS Release 15.1 and earlier, the default PIM version is version 1.

   ```
   [edit routing-instances VPN-A protocols pim rp]
   user@host# set static address 2001:db8:85a3::8a2e:370:7334 version 2
   ```

3. (Optional) Set the group address range.

   By default, a routing device running PIM is eligible to be the RP for all IPv4 or IPv6 groups (224.0.0.0/4 or FF70::/12 to FFF0::/12). The following example limits the groups for which the 2001:db8:85a3::8a2e:370:7334 address can be the RP.

   ```
   [edit routing-instances VPN-A protocols pim rp]
   user@host# set static address 2001:db8:85a3::8a2e:370:7334 group-ranges fec0::/10
   ```

   The RP that you select for a particular group must be consistent across all routers in a multicast domain.

4. (Optional) Override dynamic RP for the specified group address range.

   If you configure both static RP mapping and dynamic RP mapping (such as auto-RP) in a single routing instance, allow the static mapping to take precedence for the given static RP group range, and allow dynamic RP mapping for all other groups.

   If you exclude this statement from the configuration and you use both static and dynamic RP mechanisms for different group ranges within the same routing instance, the dynamic RP mapping takes precedence over the static RP mapping, even if static RP is defined for a specific group range.

   ```
   [edit routing-instances VPN-A protocols pim rp static address
     2001:db8:85a3::8a2e:370:7334]
   user@host# set override
   ```

5. Monitor the operation of PIM by running the **show pim** commands. Run **show pim ?** to display the supported commands.

See Also

- PIM Overview on page 185
- Understanding MLD on page 51

Release History Table

Release	Description
15.2	Starting in Junos OS Release 15.2, the static configuration uses PIM version 2 by default, which is the only version supported in that release and beyond.
15.2	Starting in Junos OS Release 15.2, the default PIM version is version 2, and version 1 is not supported.
15.1	For Junsos OS Release 15.1 and earlier, the default PIM version can be version 1 or version 2, depending on the mode you are configuring. PIM version 1 is the default for RP mode (**[edit pim rp static address *address*]**). PIM version 2 is the default for interface mode (**[edit pim interface *interface-name*]**). An explicitly configured PIM version will override the default setting.

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring a Designated Router for PIM on page 301
- Examples: Configuring PIM Sparse Mode on page 212
- *Configuring Basic PIM Settings*

Example: Configuring Anycast RP

- Understanding RP Mapping with Anycast RP on page 244
- Example: Configuring Multiple RPs in a Domain with Anycast RP on page 245
- Example: Configuring PIM Anycast With or Without MSDP on page 248
- Configuring a PIM Anycast RP Router Using Only PIM on page 251

Understanding RP Mapping with Anycast RP

Having a single active rendezvous point (RP) per multicast group is much the same as having a single server providing any service. All traffic converges on this single point, although other servers are sitting idle, and convergence is slow when the resource fails. In multicast specifically, there might be closer RPs on the shared tree, so the use of a single RP is suboptimal.

For the purposes of load balancing and redundancy, you can configure anycast RP. You can use anycast RP within a domain to provide redundancy and RP load sharing. When an RP fails, sources and receivers are taken to a new RP by means of unicast routing. When you configure anycast RP, you bypass the restriction of having one active RP per multicast group, and instead deploy multiple RPs for the same group range. The RP routers share one unicast IP address. Sources from one RP are known to other RPs that

use the Multicast Source Discovery Protocol (MSDP). Sources and receivers use the closest RP, as determined by the interior gateway protocol (IGP).

Anycast means that multiple RP routers share the same unicast IP address. Anycast addresses are advertised by the routing protocols. Packets sent to the anycast address are sent to the nearest RP with this address. Anycast addressing is a generic concept and is used in PIM sparse mode to add load balancing and service reliability to RPs.

Anycast RP is defined in RFC3446 , *Anycast RP Mechanism Using PIM and MSDP*, and can be found here: https://www.ietf.org/rfc/rfc3446.txt .

See Also

- Configuring the Static PIM RP Address on the Non-RP Routing Device on page 242
- Example: Configuring Multiple RPs in a Domain with Anycast RP on page 245
- Example: Configuring PIM Anycast With or Without MSDP on page 248

Example: Configuring Multiple RPs in a Domain with Anycast RP

This example shows how to configure anycast RP on each RP router in the PIM-SM domain. With this configuration you can deploy more than one RP for a single group range. This enables load balancing and redundancy.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 217.

Overview

When you configure anycast RP, the RP routers in the PIM-SM domain use a shared address. In this example, the shared address is 10.1.1.2/32. Anycast RP uses Multicast Source Discovery Protocol (MSDP) to discover and maintain a consistent view of the active sources. Anycast RP also requires an RP selection method, such as static, auto-RP, or bootstrap RP. This example uses static RP and shows only one RP router configuration.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network

configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

RP Routers

```
set interfaces lo0 unit 0 family inet address 192.168.132.1/32 primary
set interfaces lo0 unit 0 family inet address 10.1.1.2/32
set protocols msdp local-address 192.168.132.1
set protocols msdp peer 192.168.12.1
set protocols pim rp local address 10.1.1.2
set routing-options router-id 192.168.132.1
```

Non-RP Routers

```
set protocols pim rp static address 10.1.1.2
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure anycast RP:

1. On each RP router in the domain, configure the shared anycast address on the router's loopback address.

   ```
   [edit interfaces]
   user@host# set lo0 unit 0 family inet address 10.1.1.2/32
   ```

2. On each RP router in the domain, make sure that the router's regular loopback address is the primary address for the interface, and set the router ID.

   ```
   [edit interfaces]
   user@host# set lo0 unit 0 family inet address 192.168.132.1/32 primary

   [edit routing-options]
   user@host# set router-id 192.168.132.1
   ```

3. On each RP router in the domain, configure the local RP address, using the shared address.

   ```
   [edit protocols pim]
   user@host# set rp local address 10.1.1.2
   ```

4. On each RP router in the domain, create MSDP sessions to the other RPs in the domain.

   ```
   [edit protocols msdp]
   user@host# set local-address 192.168.132.1
   user@host# set peer 192.168.12.1
   ```

5. On each non-RP router in the domain, configure a static RP address using the shared address.

   ```
   [edit protocols pim]
   user@host# set rp static address 10.1.1.2
   ```

6. If you are done configuring the devices, commit the configuration.

```
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
lo0 {
    unit 0 {
        family inet {
            address 192.168.132.1/32 {
                primary;
            }
            address 10.1.1.2/32;
        }
    }
}
```

On the RP routers:

```
user@host# show protocols
msdp {
    local-address 192.168.132.1;
    peer 192.168.12.1;
}
pim {
    rp {
        local {
            address 10.1.1.2;
        }
    }
}
```

On the non-RP routers:

```
user@host# show protocols
pim {
    rp {
        static {
            address 10.1.1.2;
        }
    }
}
```

```
user@host# show routing-options
router-id 192.168.132.1;
```

Verification

To verify the configuration, run the show pim rps **extensive inet** command.

See Also

- Example: Configuring PIM Anycast With or Without MSDP on page 248
- Understanding PIM Sparse Mode on page 209
- Understanding RP Mapping with Anycast RP on page 244

Example: Configuring PIM Anycast With or Without MSDP

When you configure anycast RP, you bypass the restriction of having one active rendezvous point (RP) per multicast group, and instead deploy multiple RPs for the same group range. The RP routers share one unicast IP address. Sources from one RP are known to other RPs that use the Multicast Source Discovery Protocol (MSDP). Sources and receivers use the closest RP, as determined by the interior gateway protocol (IGP).

You can use anycast RP within a domain to provide redundancy and RP load sharing. When an RP stops operating, sources and receivers are taken to a new RP by means of unicast routing.

You can configure anycast RP to use PIM and MSDP for IPv4, or PIM alone for both IPv4 and IPv6 scenarios. Both are discussed in this section.

We recommend a static RP mapping with anycast RP over a bootstrap router and auto-RP configuration because it provides all the benefits of a bootstrap router and auto-RP without the complexity of the BSR and auto-RP mechanisms.

Starting in Junos OS Release 16.1, all systems on a subnet must run the same version of PIM.

The default PIM version can be version 1 or version 2, depending on the mode you are configuring. PIMv1 is the default RP mode (at the **[edit protocols pim rp static address *address*]** hierarchy level). However, PIMv2 is the default for interface mode (at the **[edit protocols pim interface *interface-name*]** hierarchy level). Explicitly configured versions override the defaults. This example explicitly configures PIMv2 on the interfaces.

The following example shows an anycast RP configuration for the RP routers, first with MSDP and then using PIM alone, and for non-RP routers.

1. For a network using an RP with MSDP, configure the RP using the **lo0** loopback interface, which is always up. Include the **address** statement and specify the unique and routable router ID and the RP address at the **[edit interfaces lo0 unit 0 family inet]** hierarchy level. In this example, the router ID is **198.51.100.254** and the shared RP address is **198.51.100.253**. Include the **primary** statement for the first address. Including the **primary** statement selects the router's primary address from all the preferred addresses on all interfaces.

```
interfaces {
  lo0 {
    description "PIM RP";
    unit 0 {
      family inet {
        address 198.51.100.254/32;
        primary;
        address 198.51.100.253/32;
```

```
        }
      }
    }
  }
```

2. Specify the RP address. Include the **address** statement at the **[edit protocols pim rp local]** hierarchy level (the same address as the secondary **lo0** interface).

   For all interfaces, include the **mode** statement to set the mode to **sparse** and the **version** statement to specify PIM version 2 at the **[edit protocols pim rp local interface all]** hierarchy level. When configuring all interfaces, exclude the **fxp0.0** management interface by including the **disable** statement for that interface.

```
protocols {
  pim {
    rp {
      local {
        family inet;
        address 198.51.100.253;
      }
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }
  }
}
```

3. Configure MSDP peering. Include the **peer** statement to configure the address of the MSDP peer at the **[edit protocols msdp]** hierarchy level. For MSDP peering, use the unique, primary addresses instead of the anycast address. To specify the local address for MSDP peering, include the **local-address** statement at the **[edit protocols msdp peer]** hierarchy level.

```
protocols {
  msdp {
    peer 198.51.100.250 {
      local-address address 198.51.100.254;
    }
  }
}
```

NOTE: If you need to configure a PIM RP for both IPv4 and IPv6 scenarios, perform Step 4 and Step 5. Otherwise, go to Step 6.

4. Configure an RP using the **lo0** loopback interface, which is always up. Include the **address** statement to specify the unique and routable router address and the RP address at the **[edit interfaces lo0 unit 0 family inet]** hierarchy level. In this example,

the router ID is **198.51.100.254** and the shared RP address is **198.51.100.253**. Include the **primary** statement on the first address. Including the **primary** statement selects the router's primary address from all the preferred addresses on all interfaces.

```
interfaces {
  lo0 {
    description "PIM RP";
    unit 0 {
      family inet {
        address 198.51.100.254/32 {
          primary;
        }
        address 198.51.100.253/32;
      }
    }
  }
}
```

5. Include the **address** statement at the **[edit protocols pim rp local]** hierarchy level to specify the RP address (the same address as the secondary **lo0** interface).

   For all interfaces, include the **mode** statement to set the mode to **sparse**, and the **version** statement to specify PIM version 2 at the **[edit protocols pim rp local interface all]** hierarchy level. When configuring all interfaces, exclude the **fxp0.0** management interface by Including the **disable** statement for that interface.

   Include the **anycast-pim** statement to configure anycast RP without MSDP (for example, if IPv6 is used for multicasting). The other RP routers that share the same IP address are configured using the **rp-set** statement. There is one entry for each RP, and the maximum that can be configured is 15. For each RP, specify the routable IP address of the router and whether MSDP source active (SA) messages are forwarded to the RP.

   MSDP configuration is not necessary for this type of IPv4 anycast RP configuration.

```
protocols {
  pim {
    rp {
      local {
        family inet {
          address 198.51.100.253;
          anycast-pim {
            rp-set {
              address 198.51.100.240;
              address 198.51.100.241 forward-msdp-sa;
            }
            local-address 198.51.100.254; #If not configured, use lo0 primary
          }
        }
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
```

```
        interface fxp0.0 {
            disable;
        }
    }
}
```

6. Configure the non-RP routers. The anycast RP configuration for a non-RP router is the same whether MSDP is used or not. Specify a static RP by adding the address at the **[edit protocols pim rp static]** hierarchy level. Include the **version** statement at the **[edit protocols pim rp static address]** hierarchy level to specify PIM version 2.

```
protocols {
    pim {
        rp {
            static {
                address 198.51.100.253 {
                    version 2;
                }
            }
        }
    }
}
```

7. Include the **mode** statement at the **[edit protocols pim interface all]** hierarchy level to specify sparse mode on all interfaces. Then include the **version** statement at the **[edit protocols pim rp interface all mode]** to configure all interfaces for PIM version 2. When configuring all interfaces, exclude the **fxp0.0** management interface by including the **disable** statement for that interface.

```
protocols {
    pim {
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring a PIM Anycast RP Router Using Only PIM

In this example, configure an RP using the **lo0** loopback interface, which is always up. Use the **address** statement to specify the unique and routable router address and the RP address at the **[edit interfaces lo0 unit 0 family inet]** hierarchy level. In this case, the router ID is 198.51.100.254/32 and the shared RP address is 198.51.100/32. Add the flag statement **primary** to the first address. Using this flag selects the router's primary address from all the preferred addresses on all interfaces.

```
interfaces {
    lo0 {
        description "PIM RP";
```

```
        unit 0 {
          family inet {
            address 198.51.100.254/32 {
              primary;
            }
            address 198.51.100.253/32;
          }
        }
      }
    }
```

Add the **address** statement at the **[edit protocols pim rp local]** hierarchy level to specify the RP address (the same address as the secondary **lo0** interface).

For all interfaces, use the **mode** statement to set the mode to **sparse**, and include the **version** statement to specify PIM version 2 at the **[edit protocols pim rp local interface all]** hierarchy level. When configuring all interfaces, exclude the **fxp0.0** management interface by adding the **disable** statement for that interface.

Use the **anycast-pim** statement to configure anycast RP without MSDP (for example, if IPv6 is used for multicasting). The other RP routers that share the same IP address are configured using the **rp-set** statement. There is one entry for each RP, and the maximum that can be configured is 15. For each RP, specify the routable IP address of the router and whether MSDP source active (SA) messages are forwarded to the RP.

```
protocols {
  pim {
    rp {
      local {
        family inet {
          address 198.51.100.253;
          anycast-pim {
            rp-set {
              address 198.51.100.240;
              address 198.51.100.241 forward-msdp-sa;
            }
            local-address 198.51.100.254; #If not configured, use lo0 primary
          }
        }
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

MSDP configuration is not necessary for this type of IPv4 anycast RP configuration.

See Also

- JTAC Certified Step-by-Step Troubleshooting: Junos OS Multicast

Release History Table

Release	Description
16.1	Starting in Junos OS Release 16.1, all systems on a subnet must run the same version of PIM.

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring a Designated Router for PIM on page 301
- Examples: Configuring PIM Sparse Mode on page 212
- *Configuring Basic PIM Settings*

Configuring PIM Bootstrap Router

Understanding the PIM Bootstrap Router

To determine which router is the rendezvous point (RP), all routers within a PIM sparse-mode domain collect bootstrap messages. A PIM sparse-mode domain is a group of routers that all share the same RP router. The domain bootstrap router initiates bootstrap messages, which are sent hop by hop within the domain. The routers use bootstrap messages to distribute RP information dynamically and to elect a bootstrap router when necessary.

See Also

- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255

Configuring PIM Bootstrap Properties for IPv4

For correct operation, every multicast router within a PIM domain must be able to map a particular multicast group address to the same Rendezvous Point (RP). The bootstrap router mechanism is one way that a multicast router can learn the set of group-to-RP mappings. Bootstrap routers are supported in IPv4 and IPv6.

NOTE: For legacy configuration purposes, there are two sections that describe the configuration of bootstrap routers: one section for both IPv4 and IPv6, and this section, which is for IPv4 only. The method described in "Configuring PIM Bootstrap Properties for IPv4 or IPv6" on page 255 is recommended. A commit error occurs if the same IPv4 bootstrap statements are included in both the IPv4-only and the IPv4-and-IPv6 sections of the hierarchy. The error message is "duplicate IPv4 bootstrap configuration."

To determine which routing device is the RP, all routing devices within a PIM domain collect bootstrap messages. A PIM domain is a contiguous set of routing devices that implement PIM. All are configured to operate within a common boundary. The domain's bootstrap router initiates bootstrap messages, which are sent hop by hop within the domain. The routing devices use bootstrap messages to distribute RP information dynamically and to elect a bootstrap router when necessary.

You can configure bootstrap properties globally or for a routing instance. This example shows the global configuration.

To configure the bootstrap router properties:

1. Configure the bootstrap priority.

   By default, each routing device has a bootstrap priority of 0, which means the routing device can never be the bootstrap router. A priority of 0 disables the function for IPv4 and does not cause the routing device to send bootstrap router packets with a 0 in the priority field. The routing device with the highest priority value is elected to be the bootstrap router. In the case of a tie, the routing device with the highest IP address is elected to be the bootstrap router. A simple bootstrap configuration assigns a bootstrap priority value to a routing device.

   ```
   [edit protocols pim rp]
   user@host# set bootstrap-priority 3
   ```

2. (Optional) Create import and export policies to control the flow of IPv4 bootstrap messages to and from the RP, and apply the policies to PIM. Import and export policies are useful when some of the routing devices in your PIM domain have interfaces that connect to other PIM domains. Configuring a policy prevents bootstrap messages from crossing domain boundaries. The **bootstrap-import** statement prevents messages from being imported into the RP. The **bootstrap-export** statement prevents messages from being exported from the RP.

   ```
   [edit protocols pim rp]
   user@host# set bootstrap-import pim-bootstrap-import
   user@host# set bootstrap-export pim-bootstrap-export
   ```

3. Configure the policies.

   ```
   [edit policy-options policy-statement pim-bootstrap-import]
   user@host# set from interface se-0/0/0
   user@host# set then reject
   ```

```
[edit policy-options policy-statement pim-bootstrap-export]
user@host# set from interface se-0/0/0
user@host# set then reject
```

4. Monitor the operation of PIM bootstrap routing devices by running the **show pim bootstrap** command.

See Also

- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255
- Understanding PIM Sparse Mode on page 209
- Example: Rejecting PIM Bootstrap Messages at the Boundary of a PIM Domain on page 257
- **show pim bootstrap on page 1637** in the CLI Explorer

Configuring PIM Bootstrap Properties for IPv4 or IPv6

For correct operation, every multicast router within a PIM domain must be able to map a particular multicast group address to the same Rendezvous Point (RP). The bootstrap router mechanism is one way that a multicast router can learn the set of group-to-RP mappings. Bootstrap routers are supported in IPv4 and IPv6.

NOTE: For legacy configuration purposes, there are two sections that describe the configuration of bootstrap routers: one section for IPv4 only, and this section, which is for both IPv4 and IPv6. The method described in this section is recommended. A commit error occurs if the same IPv4 bootstrap statements are included in both the IPv4-only and the IPv4-and-IPv6 sections of the hierarchy. The error message is “duplicate IPv4 bootstrap configuration.”

To determine which routing device is the RP, all routing devices within a PIM domain collect bootstrap messages. A PIM domain is a contiguous set of routing devices that implement PIM. All devices are configured to operate within a common boundary. The domain's bootstrap router initiates bootstrap messages, which are sent hop by hop within the domain. The routing devices use bootstrap messages to distribute RP information dynamically and to elect a bootstrap router when necessary.

You can configure bootstrap properties globally or for a routing instance. This example shows the global configuration.

To configure the bootstrap router properties:

1. Configure the bootstrap priority.

   By default, each routing device has a bootstrap priority of 0, which means the routing device can never be the bootstrap router. The routing device with the highest priority value is elected to be the bootstrap router. In the case of a tie, the routing device with the highest IP address is elected to be the bootstrap router. A simple bootstrap configuration assigns a bootstrap priority value to a routing device.

NOTE: In the IPv4-only configuration, specifying a bootstrap priority of 0 disables the bootstrap function and does not cause the routing device to send BSR packets with a 0 in the priority field. In the configuration shown here, specifying a bootstrap priority of 0 does not disable the function, but causes the routing device to send BSR packets with a 0 in the priority field. To disable the bootstrap function in the IPv4 and IPv6 configuration, delete the **bootstrap** statement.

```
user@host# edit protocols pim rp
user@host# set bootstrap family inet priority 3
```

2. (Optional) Create import and export policies to control the flow of bootstrap messages to and from the RP, and apply the policies to PIM. Import and export policies are useful when some of the routing devices in your PIM domain have interfaces that connect to other PIM domains. Configuring a policy prevents bootstrap messages from crossing domain boundaries. The **import** statement prevents messages from being imported into the RP. The **export** statement prevents messages from being exported from the RP.

   ```
   [edit protocols pim rp]
   user@host# set bootstrap family inet import pim-bootstrap-import
   user@host# set bootstrap family inet export pim-bootstrap-export
   ```

3. Configure the policies.

   ```
   [edit policy-options policy-statement pim-bootstrap-import]
   user@host# set from interface se-0/0/0
   user@host# set then reject
   user@host# exit
   user@host# edit policy-options policy-statement pim-bootstrap-export
   user@host# set from interface se-0/0/0
   user@host# set then reject
   ```

4. Monitor the operation of PIM bootstrap routing devices by running the **show pim bootstrap** command.

See Also

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Understanding PIM Sparse Mode on page 209
- Example: Rejecting PIM Bootstrap Messages at the Boundary of a PIM Domain on page 257
- **show pim bootstrap on page 1637** in the CLI Explorer

Example: Rejecting PIM Bootstrap Messages at the Boundary of a PIM Domain

In this example, the **from interface so-0-1/0 then reject** policy statement rejects bootstrap messages from the specified interface (the example is configured for both IPv4 and IPv6 operation):

```
protocols {
   pim {
      rp {
         bootstrap {
            family inet {
               priority 1;
               import pim-import;
               export pim-export;
            }
            family inet6 {
               priority 1;
               import pim-import;
               export pim-export;
            }
         }
      }
   }
}
policy-options {
   policy-statement pim-import {
      from interface so-0/1/0;
      then reject;
   }
   policy-statement pim-export {
      to interface so-0/1/0;
      then reject;
   }
}
```

Example: Configuring PIM BSR Filters

Configure a filter to prevent BSR messages from entering or leaving your network. Add this configuration to all routers:

```
protocols {
   pim {
      rp {
         bootstrap-import no-bsr;
         bootstrap-export no-bsr;
      }
   }
}
policy-options {
   policy-statement no-bsr {
      then reject;
   }
}
```

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring a Designated Router for PIM on page 301
- Examples: Configuring PIM Sparse Mode on page 212
- *Configuring Basic PIM Settings*

Configuring PIM Auto-RP

Understanding PIM Auto-RP

You can configure a more dynamic way of assigning rendezvous points (RPs) in a multicast network by means of auto-RP. When you configure auto-RP for a router, the router learns the address of the RP in the network automatically and has the added advantage of operating in PIM version 1 and version 2.

Although auto-RP is a nonstandard (non-RFC-based) function that typically uses dense mode PIM to advertise control traffic, it provides an important failover advantage that simple static RP assignment does not. You can configure multiple routers as RP candidates. If the elected RP fails, one of the other preconfigured routers takes over the RP functions. This capability is controlled by the auto-RP mapping agent.

See Also

- Configuring PIM Auto-RP on page 258

Configuring PIM Auto-RP

For correct operation, every multicast router within a PIM domain must be able to map a particular multicast group address to the same rendezvous point (RP). The auto-RP mechanism is one way that a multicast router can learn the set of group-to-RP mappings. Auto-RP automatically distributes mapping information to routing devices. It simplifies use of multiple RPs for different multicast group ranges, thus allowing multiple RPs to act as backups for each other. Auto-RP relies on a router to act as the RP mapping agent. Potential RPs announce themselves to the mapping agent, and the mapping agent resolves any conflicts.

The mapping agent sends the multicast group-RP mapping information to the other routers using PIM dense mode. The specific groups used are 224.0.1.39 and .40. The first (.39) is used to advertise, the second (.40) is used for discovery. Because PIM dense mode is necessary to enable auto-RP to work, which in turns enables PIM sparse mode to work, you must configure PIM sparse-dense mode in the PIM domains that use auto-RP.

Although auto-RP is a nonstandard (non-RFC-based) function requiring dense mode PIM to advertise control traffic, it provides an important failover advantage that static RP assignment does not. That is, you can configure multiple routing devices as RP candidates. If the elected RP fails, one of the other preconfigured routing devices takes over the RP functions. This capability is controlled by the auto-RP mapping agent.

In most cases, how the routing device handles auto-RP discovery, announce, or mapping messages depends on whether the routing device is an RP (configured as local RP) or not. Table 11 on page 259 shows how the routing device behaves depending on the local RP configuration.

Table 11: Local RP and Auto-RP Message Types

Auto-RP Message Type	Local RP?	Routing Device Behavior
discovery	No	Listen for auto-RP mapping messages.
discovery	Yes	Listen for auto-RP mapping messages.
announce	No	Listen for auto-RP mapping messages.
announce	Yes	Listen for auto-RP mapping messages. Send auto-RP announce messages.
mapping	No	Listen for auto-RP mapping messages. Listen for auto-RP announce messages. If elected mapping agent, send auto-RP mapping messages.
mapping	Yes	Listen for auto-RP mapping messages. Send auto-RP announce messages. Listen for auto-RP announce messages. If elected mapping agent, send auto-RP mapping messages.

NOTE: If the routing device receives auto-RP announcements split across multiple messages, the routing device loses the information in the previous part of the message as soon as the next part of the message is received.

You can configure auto-RP properties globally or for a routing instance. This example shows the global configuration.

To configure auto-RP properties:

1. Configure PIM in sparse-dense mode on all routing devices in the PIM domain.

   ```
   [edit protocols pim]
   user@host# edit
   user@host# set interface all mode sparse-dense
   ```

   This configuration allows the routing device to operate in sparse mode for most groups and dense mode for others. The default is to operate in sparse mode unless the routing device is specifically informed of a dense mode group.

2. Configure a routable loopback interface address on all routing devices in the PIM domain.

The routing device joins the auto-RP groups on the configured interfaces and on the loopback interface **lo0.0**. For auto-RP to work correctly, configure a routable IP address on the loopback interface. You cannot use the loopback address 127.0.0.1. Also, you must enable PIM sparse-dense mode on the **lo0.0** interface if you do not specify **interface all**.

```
[edit interfaces lo0.0 unit 0 family inet]
user@host# set address 192.168.0.3 preferred
```

3. Configure the two multicast dense groups on all the routing devices.

   Auto-RP requires multicast flooding to announce potential RP candidates and to discover the elected RPs in the network. Multicast flooding occurs through a PIM dense mode model, where group 224.0.1.39 is used for **announce** messages and group 224.0.1.40 is used for **discovery** messages.

   ```
   [edit protocols pim]
   user@host# set dense-groups 224.0.1.39/32
   user@host# set dense-groups 224.0.1.40/32
   ```

   TIP: Step 3 is required. When auto-RP is enabled, the auto-RP announce group (224.0.1.39) and auto-RP-discovery group (224.0.1.40) must be configured explicitly as dense groups. When the auto-RP discovery group is not configured as a dense group, auto-RP is not enabled. When the auto-RP announce group is not configured as a dense group, auto-RP is enabled in the discovery mode only, and mapping and announce modes are disabled.

4. Configure the auto-RP **announce** option.

   At least one routing device in the PIM domain must announce auto-RP messages and at least one must map them, or you can configure a routing device to perform both functions.

   When a routing device sends announce messages in the network, it is advertising itself as a candidate RP. A routing device configured with this option must also be configured as an RP, or announce messages are not sent.

   ```
   [edit protocols pim rp]
   user@host# set local address 192.168.0.1
   user@host# set auto-rp announce
   ```

   NOTE: You cannot include the **auto-rp announce** option at the **[edit logical-systems** *logical-system-name* **routing-instances** *routing-instance-name* **protocols pim]** hierarchy level.

5. Configure the auto-RP mapping agent.

The mapping agent sends discovery messages to the network, informing all routing devices in a multicast group of which RP to use. If the mapping agent is also an RP, the **mapping** option also allows the routing device to send auto-RP announcements (mapping on an RP allows the routing device to perform both the announcement and mapping functions).

```
[edit protocols pim rp]
user@host# set auto-rp mapping
```

If the mapping agent is also an RP, configure the mapping agent as a local RP.

```
[edit protocols pim rp]
user@host# set local address 192.168.0.2
```

6. Configure mapping agent election.

   If you configure the **mapping** option on more than one routing device in the PIM domain, configure mapping agent election on each potential mapping agent.

   Auto-RP specifications state that mapping agents do not send mapping messages if they receive messages from a mapping agent with a higher IP address. However, some vendors' mapping agents continue to announce mappings, even in the presence of higher-addressed mapping agents. In other words, some mapping agents will always send mapping messages.

   The default auto-RP operation is to perform mapping agent election. To explicitly configure mapping agent election, you can include the **mapping-agent-election** statement. When this option is configured, the mapping agent will stop sending mapping messages if it receives messages from a mapping agent with a higher IP address.

   ```
   [edit protocols pim rp]
   user@host# set auto-rp mapping mapping-agent-election
   ```

   Mapping message suppression is disabled with the **no-mapping-agent-election** statement. When this option is configured, the mapping agent will always send mapping messages even in the presence of higher-addressed mapping agents.

   To disable mapping agent election for compatibility with other vendors' equipment, include the **no-mapping-agent-election** statement.

   ```
   [edit protocols pim rp]
   user@host# set auto-rp mapping no-mapping-agent-election
   ```

7. Configure the remaining routing devices in the PIM domain to discover the RP.

   Discovery enables the routing devices to receive and process discovery messages from the mapping agent. This is the most basic auto-RP option.

   ```
   [edit protocols pim rp]
   user@host# set auto-rp discovery
   ```

8. Monitor the operation of PIM auto-RP routers by running the following commands:

   - show pim interfaces

- show pim rps
- show pim rps

9. Issue the **show pim rps extensive** command to see information about how an RP is learned, what groups it handles, and the number of groups actively using the RP.

```
user@host> show pim rps extensive
RP: 192.168.5.1
Learned from 192.168.5.1 via: auto-rp
Time Active: 00:34:29
Holdtime: 150 with 108 remaining
Device Index: 6
Subunit: 32769
Interface: pd-0/0/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        224.2.2.100
        total 1 groups active
Register State for RP:
Group     Source FirstHop       RP Address      StateRP address Type Holdtime
 Timeout
```

In the example, the RP at 192.168.5.1 was learned through auto-RP. The RP is able to support all groups in the 224.0.0.0/4 range (all possible groups). The local router has sent PIM control traffic for the 224.2.2.100 group to the RP.

Additionally, the presence of a Tunnel Physical Interface Card (PIC) in an RP router creates a de-encapsulation interface, which allows the RP to receive multicast traffic from the source. This interface is indicated by **pd-0/0/0.32769**.

See Also

- Understanding PIM Sparse Mode on page 209
- **show pim interfaces on page 1639**
- **show pim rps on page 1678**

Related Documentation

- Configuring PIM Bootstrap Router on page 253
- Configuring a Designated Router for PIM on page 301
- Examples: Configuring PIM Sparse Mode on page 212
- *Configuring Basic PIM Settings*

Configuring All PIM Anycast Non-RP Routers

Use the **mode** statement at the **[edit protocols pim rp interface all]** hierarchy level to specify sparse mode on all interfaces. Then add the **version** statement at the **[edit protocols pim rp interface all mode]** to configure all interfaces for PIM version 2. When configuring all interfaces, exclude the **fxp0.0** management interface by adding the **disable** statement for that interface.

```
protocols {
  pim {
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Configuring a PIM Anycast RP Router with MSDP

Add the **address** statement at the **[edit protocols pim rp local]** hierarchy level to specify the RP address (the same address as the secondary **lo0** interface).

For all interfaces, use the **mode** statement to set the mode to **sparse** and the **version** statement to specify PIM version 2 at the **[edit protocols pim rp local interface all]** hierarchy level. When configuring all interfaces, exclude the **fxp0.0** management interface by adding the **disable** statement for that interface.

```
protocols {
  pim {
    rp {
      local {
        family inet;
        address 198.51.100.253;
      }
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }
  }
}
```

To configure MSDP peering, add the **peer** statement to configure the address of the MSDP peer at the **[edit protocols msdp]** hierarchy level. For MSDP peering, use the unique, primary addresses instead of the anycast address. To specify the local address for MSDP peering, add the **local-address** statement at the **[edit protocols msdp peer]** hierarchy level.

```
protocols {
  msdp {
    peer 198.51.100.250 {
      local-address 198.51.100.254;
    }
  }
}
```

Configuring Embedded RP

Understanding Embedded RP for IPv6 Multicast

Global IPv6 multicast between routing domains has been possible only with source-specific multicast (SSM) because there is no way to convey information about IPv6 multicast RPs between PIM sparse mode RPs. In IPv4 multicast networks, this information is conveyed between PIM RPs using MSDP, but there is no IPv6 support in current MSDP standards. IPv6 uses the concept of an embedded RP to resolve this issue without requiring SSM. This feature embeds the RP address in an IPv6 multicast address.

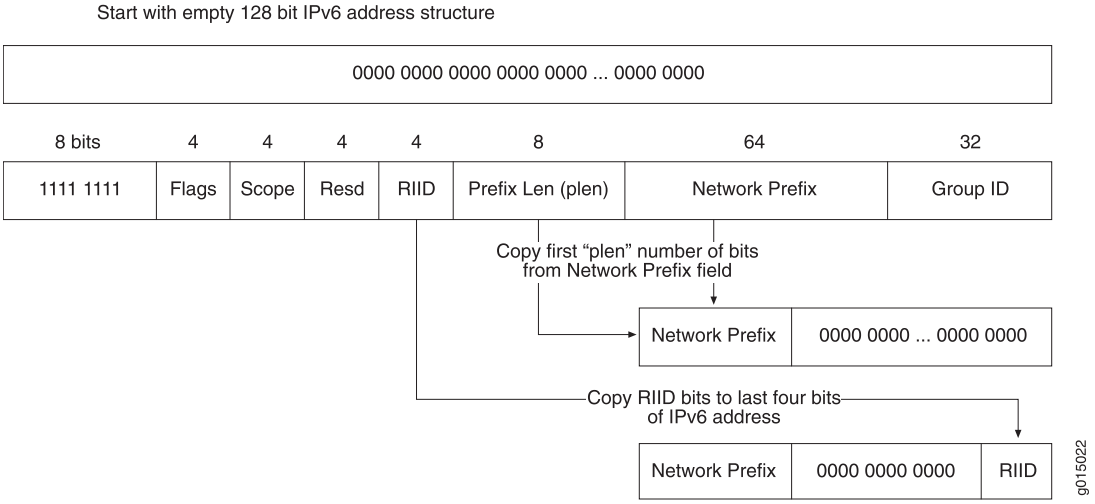
All IPv6 multicast addresses begin with 8 1-bits (1111 1111) followed by a 4-bit flag field normally set to 0011. The flag field is set to 0111 when embedded RP is used. Then the low-order bits of the normally reserved field in the IPv6 multicast address carry the 4-bit RP interface identifier (RIID).

When the IPv6 address of the RP is embedded in a unicast-prefix-based any-source multicast (ASM) address, all of the following conditions must be true:

- The address must be an IPv6 multicast address and have 0111 in the flags field (that is, the address is part of the prefix FF70::/12).
- The 8-bit prefix length (plen) field must not be all 0. An all 0 plen field implies that SSM is in use.
- The 8-bit prefix length field value must not be greater than 64, which is the length of the network prefix field in unicast-prefix-based ASM addresses.

The routing platform derives the value of the interdomain RP by copying the prefix length field number of bits from the 64-bit network prefix field in the received IPv6 multicast address to an empty 128-bit IPv6 address structure and copying the last bits from the 4-bit RIID. For example, if the prefix length field bits have the value 32, then the routing platform copies the first 32 bits of the IPv6 multicast address network prefix field to an all-0 IPv6 address and appends the last four bits determined by the RIID. See Figure 34 on page 265 for an illustration of this process.

Figure 34: Extracting the Embedded RP IPv6 Address



For example, the administrator of IPv6 network 2001:DB8::/32 sets up an RP for the 2001:DB8:BEEF:FEED::/96 subnet. In that case, the received embedded RP IPv6 ASM address has the form:

FF70:y40:2001:DB8:BEEF:FEED::/96

and the derived RP IPv6 address has the form:

2001:DB8:BEEF:FEED::y

where **y** is the RIID (**y** cannot be 0).

When configured, the routing platform checks for embedded RP information in every PIM join request received for IPv6. The use of embedded RP does not change the processing of IPv6 multicast and RPs in any way, except that the embedded RP address is used if available and selected for use. There is no need to specify the IPv6 address family for embedded RP configuration because the information can be used only if IPv6 multicast is properly configured on the routing platform.

The following receive events trigger extraction of an IPv6 embedded RP address on the routing platform:

- Multicast Listener Discovery (MLD) report for an embedded RP multicast group address
- PIM join message with an embedded RP multicast group address
- Static embedded RP multicast group address associated with an interface
- Packets sent to an embedded RP multicast group address received on the DR

The embedded RP node discovered through these events is added if it does not already exist on the routing platform. The routing platform chooses the embedded RP as the RP for a multicast group before choosing an RP learned through BSRs or a statically configured RP. The embedded RP is removed whenever all PIM join states using this RP are removed or the configuration changes to remove the embedded RP feature.

Configuring PIM Embedded RP for IPv6

You configure embedded RP to allow multidomain IPv6 multicast networks to find RPs in other routing domains. Embedded RP embeds an RP address inside PIM join messages and other types of messages sent between routing domains. Global IPv6 multicast between routing domains has been possible only with source-specific multicast (SSM) because there is no way to convey information about IPv6 multicast RPs between PIM sparse mode RPs. In IPv4 multicast networks, this information is conveyed between PIM RPs using MSDP, but there is no IPv6 support in current MSDP standards. IPv6 uses the concept of an embedded RP to resolve this issue without requiring SSM. Thus, embedded RP enables you can deploy IPv6 with any-source multicast (ASM).

Embedded RP is disabled by default.

When you configure embedded RP for IPv6, embedded RPs are preferred to RPs discovered by IPv6 any other way. You configure embedded RP independent of any other IPv6 multicast properties. This feature is applied only when IPv6 multicast is properly configured.

You can configure embedded RP globally or for a routing instance. This example shows the routing instance configuration.

To configure embedded RP for IPv6 PIM sparse mode:

1. Define which multicast addresses or prefixes can embed RP address information. If messages within a group range contain embedded RP information and the group range is not configured, the embedded RP in that group range is ignored. Any valid unicast-prefix-based ASM address can be used as a group range. The default group range is FF70::/12 to FFF0::/12. Messages with embedded RP information that do not match any configured group ranges are treated as normal multicast addresses.

   ```
   [edit routing-instances vpn-A protocols pim rp embedded-rp]
   user@host# set group-ranges fec0::/10
   ```

   If the derived RP address is not a valid IPv6 unicast address, it is treated as any other multicast group address and is not used for RP information. Verification fails if the extracted RP address is a local interface, unless the routing device is configured as an RP and the extracted RP address matches the configured RP address. Then the local RP determines whether it is configured to act as an RP for the embedded RP multicast address.

2. Limit the number of embedded RPs created in a specific routing instance. The range is from 1 through 500. The default is 100.

   ```
   [edit routing-instances vpn-A protocols pim rp]
   user@host# set maximum-rps 50
   ```

3. Monitor the operation by running the **show pim rps** and **show pim statistics** commands.

See Also

- Understanding Embedded RP for IPv6 Multicast on page 264

- **show pim rps on page 1678** in the CLI Explorer
- **show pim statistics on page 1689** in the CLI Explorer

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring a Designated Router for PIM on page 301
- Examples: Configuring PIM Sparse Mode on page 212
- *Configuring Basic PIM Settings*

Configuring PIM Filtering

Understanding Multicast Message Filters

Multicast sources and routers generate a considerable number of control messages, especially when using PIM sparse mode. These messages form distribution trees, locate rendezvous points (RPs) and designated routers (DRs), and transition from one type of tree to another. In most cases, this multicast messaging system operates transparently and efficiently. However, in some configurations, more control over the sending and receiving of multicast control messages is necessary.

You can configure multicast filtering to control the sending and receiving of multicast control messages.

To prevent unauthorized groups and sources from registering with an RP router, you can define a routing policy to reject PIM register messages from specific groups and sources and configure the policy on the designated router or the RP router.

- If you configure the reject policy on an RP router, it rejects incoming PIM register messages from the specified groups and sources. The RP router also sends a register stop message by means of unicast to the designated router. On receiving the register stop message, the designated router sends periodic null register messages for the specified groups and sources to the RP router.

- If you configure the reject policy on a designated router, it stops sending PIM register messages for the specified groups and sources to the RP router.

NOTE: If you have configured the reject policy on an RP router, we recommend that you configure the same policy on all the RP routers in your multicast network.

NOTE: If you delete a group and source address from the reject policy configured on an RP router and commit the configuration, the RP router will register the group and source only when the designated router sends a null register message.

See Also

- Filtering MAC Addresses on page 268
- Filtering RP and DR Register Messages on page 268
- Filtering MSDP SA Messages on page 269

Filtering MAC Addresses

When a router is exclusively configured with multicast protocols on an interface, multicast sets the interface media access control (MAC) filter to multicast promiscuous mode, and the number of multicast groups is unlimited. However, when the router is not exclusively used for multicasting and other protocols such as OSPF, Routing Information Protocol version 2 (RIPv2), or Network Time Protocol (NTP) are configured on an interface, each of these protocols individually requests that the interface program the MAC filter to pick up its respective multicast group only. In this case, without multicast configured on the interface, the maximum number of multicast MAC filters is limited to 20. For example, the maximum number of interface MAC filters for protocols such as OSPF (multicast group 224.0.0.5) is 20, unless a multicast protocol is also configured on the interface.

No configuration is necessary for MAC filters.

Filtering RP and DR Register Messages

You can filter Protocol Independent Multicast (PIM) register messages sent from the designated router (DR) or to the rendezvous point (RP). The PIM RP keeps track of all active sources in a single PIM sparse mode domain. In some cases, more control over which sources an RP discovers, or which sources a DR notifies other RPs about, is desired. A high degree of control over PIM register messages is provided by RP and DR register message filtering. Message filtering also prevents unauthorized groups and sources from registering with an RP router.

Register messages that are filtered at a DR are not sent to the RP, but the sources are available to local users. Register messages that are filtered at an RP arrive from source

DRs, but are ignored by the router. Sources on multicast group traffic can be limited or directed by using RP or DR register message filtering alone or together.

If the action of the register filter policy is to discard the register message, the router needs to send a register-stop message to the DR. Register-stop messages are throttled to prevent malicious users from triggering them on purpose to disrupt the routing process.

Multicast group and source information is encapsulated inside unicast IP packets. This feature allows the router to inspect the multicast group and source information before sending or accepting the PIM register message.

Incoming register messages to an RP are passed through the configured register message filtering policy before any further processing. If the register message is rejected, the RP router sends a register-stop message to the DR. When the DR receives the register-stop message, the DR stops sending register messages for the filtered groups and sources to the RP. Two fields are used for register message filtering:

- Group multicast address
- Source address

The syntax of the existing policy statements is used to configure the filtering on these two fields. The **route-filter** statement is useful for multicast group address filtering, and the **source-address-filter** statement is useful for source address filtering. In most cases, the action is to **reject** the register messages, but more complex filtering policies are possible.

Filtering cannot be performed on other header fields, such as DR address, protocol, or port. In some configurations, an RP might not send register-stop messages when the policy action is to discard the register messages. This has no effect on the operation of the feature, but the router will continue to receive register messages.

When anycast RP is configured, register messages can be sent or received by the RP. All the RPs in the anycast RP set need to be configured with the same RP register message filtering policies. Otherwise, it might be possible to circumvent the filtering policy.

See Also

- Understanding RP Mapping with Anycast RP on page 244
- Configuring Register Message Filters on a PIM RP and DR on page 279

Filtering MSDP SA Messages

Along with applying MSDP source active (SA) filters on all external MSDP sessions (in and out) to prevent SAs for groups and sources from leaking in and out of the network, you need to apply bootstrap router (BSR) filters. Applying a BSR filter to the boundary of a network prevents foreign BSR messages (which announce RP addresses) from leaking into your network. Since the routers in a PIM sparse-mode domain need to know the address of only one RP router, having more than one in the network can create issues.

If you did not use multicast scoping to create boundary filters for all customer-facing interfaces, you might want to use PIM join filters. Multicast scopes prevent the actual multicast data packets from flowing in or out of an interface. PIM join filters prevent PIM

sparse-mode state from being created in the first place. Since PIM join filters apply only to the PIM sparse-mode state, it might be more beneficial to use multicast scoping to filter the actual data.

NOTE: When you apply firewall filters, firewall action modifiers, such as log, sample, and count, work only when you apply the filter on an inbound interface. The modifiers do not work on an outbound interface.

See Also

- Filtering Incoming PIM Join Messages on page 275
- Example: Configuring PIM BSR Filters on page 257

Configuring Interface-Level PIM Neighbor Policies

You can configure a policy to filter unwanted PIM neighbors. In the following example, the PIM interface compares neighbor IP addresses with the IP address in the policy statement before any hello processing takes place. If any of the neighbor IP addresses (primary or secondary) match the IP address specified in the prefix list, PIM drops the hello packet and rejects the neighbor.

If you configure a PIM neighbor policy after PIM has already established a neighbor adjacency to an unwanted PIM neighbor, the adjacency remains intact until the neighbor hold time expires. When the unwanted neighbor sends another hello message to update its adjacency, the router recognizes the unwanted address and rejects the neighbor.

To configure a policy to filter unwanted PIM neighbors:

1. Configure the policy. The neighbor policy must be a properly structured policy statement that uses a prefix list (or a route filter) containing the neighbor primary address (or any secondary IP addresses) in a prefix list, and the **reject** option to reject the unwanted address.

   ```
   [edit policy-options]
   user@host# set prefix-list nbrGroup 1 20.20.20.1/32
   user@host# set policy-statement nbr-policy from prefix-list nbrGroup1
   user@host# set policy-statement nbr-policy then reject
   ```

2. Configure the interface globally or in the routing instance. This example shows the configuration for the routing instance.

   ```
   [edit routing-instances PIM.master protocols pim]
   user@host# set neighbor-policy nbr-policy
   ```

3. Verify the configuration by checking the **Hello dropped on neighbor policy** field in the output of the **show pim statistics** command.

See Also

- Understanding PIM Sparse Mode on page 209
- show pim statistics on page 1689

Filtering Outgoing PIM Join Messages

When the core of your network is using MPLS, PIM join and prune messages stop at the customer edge (CE) routers and are not forwarded toward the core, because these routers do not have PIM neighbors on the core-facing interfaces. When the core of your network is using IP, PIM join and prune messages are forwarded to the upstream PIM neighbors in the core of the network.

When the core of your network is using a mix of IP and MPLS, you might want to filter certain PIM join and prune messages at the upstream egress interface of the CE routers.

You can filter PIM sparse mode (PIM-SM) join and prune messages at the egress interfaces for IPv4 and IPv6 in the upstream direction. The messages can be filtered based on the group address, source address, outgoing interface, PIM neighbor, or a combination of these values. If the filter is removed, the join is sent after the PIM periodic join timer expires.

To filter PIM sparse mode join and prune messages at the egress interfaces, create a policy rejecting the group address, source address, outgoing interface, or PIM neighbor, and then apply the policy.

The following example filters PIM join and prune messages for group addresses 224.0.1.2 and 225.1.1.1.

1. In configuration mode, create the policy.

```
user@host# set policy-options policy-statement block-groups term t1 from route-filter
  224.0.1.2/32 exact
user@host# set policy-options policy-statement block-groups term t1 from route-filter
  225.1.1.1/32 exact
user@host# set policy-options policy-statement block-groups term t1 then reject
user@host# set policy-options policy-statement block-groups term last then accept
```

2. Verify the policy configuration by running the **show policy-options** command.

```
user@host# show policy-options
policy-statement block-groups {
  term t1 {
    from {
      route-filter 224.0.1.2/32 exact;
      route-filter 225.1.1.1/32 exact;
      then reject;
    }
    term last {
      then accept;
    }
  }
```

3. Apply the PIM join and prune message filter.

```
user@host> set protocols pim export block-groups
```

4. After the configuration is committed, use the **show pim statistics** command to verify that outgoing PIM join and prune messages are being filtered.

```
user@host> show pim statistics | grep filtered
RP Filtered Source                  0

Rx Joins/Prunes filtered            0

Tx Joins/Prunes filtered            254
```

The egress filter count is shown on the **Tx Joins/Prunes filtered** line.

See Also
- Filtering Incoming PIM Join Messages on page 275

Example: Stopping Outgoing PIM Register Messages on a Designated Router

This example shows how to stop outgoing PIM register messages on a designated router.

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements.
8. Configure IGMP.
9. Configure the PIM static RP.
10. Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 276.

Overview

In this example, you configure the group address as **224.2.2.2/32** and the source address in the group as **20.20.20.1/32**. You set the match action to not send PIM register messages

for the group and source address. Then you configure the policy on the designated router to **stop-pim-register-msg-dr**.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement stop-pim-register-msg-dr from route-filter
  224.2.2.2/32 exact
set policy-options policy-statement stop-pim-register-msg-dr from source-address-filter
  20.20.20.1/32 exact
set policy-options policy-statement stop-pim-register-msg-dr then reject
set protocols pim rp dr-register-policy stop-pim-register-msg-dr
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To stop outgoing PIM register messages on a designated router:

1. Configure the policy options.

   ```
   [edit]
   user@host# edit policy-options
   ```

2. Set the group address.

   ```
   [edit policy-options]
   user@host# set policy statement stop-pim-register-msg-dr from route-filter
     224.2.2.2/32 exact
   ```

3. Set the source address.

   ```
   [edit policy-options]
   user@host# set policy statement stop-pim-register-msg-dr from
     source-address-filter 20.20.20.1/32 exact
   ```

4. Set the match action.

   ```
   [edit policy-options]
   user@host# set policy statement stop-pim-register-msg-dr then reject
   ```

5. Assign the policy.

   ```
   [edit]
   user@host# set dr-register-policy stop-pim-register-msg-dr
   ```

Results From configuration mode, confirm your configuration by entering the **show policy-options** and **show protocols** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show policy-options
  policy-statement stop-pim-register-msg-dr {
  from {
  route-filter 224.2.2.2/32 exact;
  source-address-filter 20.20.20.1/32 exact;
  }
  then reject;
}
[edit]
  user@host# show protocols
  pim {
  rp {
  dr-register-policy stop-pim-register-msg-dr;
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Verifying the PIM RP Configuration

Purpose Verify that the PIM RP is statically configured with the correct IP address.

Action From operational mode, enter the **show pim rps** command.

See Also
- Configuring Register Message Filters on a PIM RP and DR on page 279
- Multicast Configuration Overview on page 16

Filtering Incoming PIM Join Messages

Multicast scoping controls the propagation of multicast messages. Whereas multicast scoping prevents the actual multicast data packets from flowing in or out of an interface, PIM join filters prevent a state from being created in a router. A state—the (*,G) or (S,G) entries—is the information used for forwarding unicast or multicast packets. Using PIM join filters prevents the transport of multicast traffic across a network and the dropping of packets at a scope at the edge of the network. Also, PIM join filters reduce the potential for denial-of-service (DoS) attacks and PIM state explosion—large numbers of PIM join messages forwarded to each router on the rendezvous-point tree (RPT), resulting in memory consumption.

To use PIM join filters to efficiently restrict multicast traffic from certain source addresses, create and apply the routing policy across all routers in the network.

See Table 12 on page 275 for a list of match conditions.

Table 12: PIM Join Filter Match Conditions

Match Condition	Matches On
interface	Router interface or interfaces specified by name or IP address
neighbor	Neighbor address (the source address in the IP header of the join and prune message)
route-filter	Multicast group address embedded in the join and prune message
source-address-filter	Multicast source address embedded in the join and prune message

The following example shows how to create a PIM join filter. The filter is composed of a route filter and a source address filter—**bad-groups** and **bad-sources**, respectively. the **bad-groups** filter prevents (*,G) or (S,G) join messages from being received for all groups listed. The **bad-sources** filter prevents (S,G) join messages from being received for all sources listed. The **bad-groups** filter and **bad-sources** filter are in two different terms. If route filters and source address filters are in the same term, they are logically ANDed.

To filter incoming PIM join messages:

1. Configure the policy.

   ```
   [edit policy-statement pim-join-filter term bad-groups]
   user@host# set from route-filter 224.0.1.2/32 exact
   user@host# set from route-filter 239.0.0.0/8 orlonger
   user@host# set then reject

   [edit policy-statement pim-join-filter term bad-sources]
   user@host# set from source-address-filter 10.0.0.0/8 orlonger
   user@host# set from source-address-filter 127.0.0.0/8 orlonger
   user@host# set then reject

   [edit policy-statement pim-join-filter term last]
   user@host# set then accept
   ```

2. Apply one or more policies to routes being imported into the routing table from PIM.

   ```
   [edit protocols pim]
   user@host# set import pim-join-filter
   ```

3. Verify the configuration by checking the output of the **show pim join** and **show policy** commands.

See Also

- Understanding Multicast Administrative Scoping on page 899
- Filtering Outgoing PIM Join Messages on page 271
- **show pim join on page 1642** in the CLI Explorer
- **show policy on page 1628** in the CLI Explorer

Example: Rejecting Incoming PIM Register Messages on RP Routers

This example shows how to reject incoming PIM register messages on RP routers.

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.

4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Configuring the Session Announcement Protocol" on page 415.
8. Configure IGMP. See "Configuring IGMP" on page 23.
9. Configure the PIM static RP. See "Configuring Static RP" on page 237.

Overview

In this example, you configure the group address as **224.1.1.1/32** and the source address in the group as **10.10.10.1/32**. You set the match action to reject PIM register messages and assign reject-pim-register-msg-rp as the policy on the RP.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level and then enter **commit** from configuration mode.

```
set policy-options policy-statement reject-pim-register-msg-rp from route-filter
  224.1.1.1/32 exact
set policy-options policy-statement reject-pim-register-msg-rp from source-address-filter
  10.10.10.1/32 exact
set policy-options policy-statement reject-pim-register-msg-rp then reject
set protocols pim rp rp-register-policy reject-pim-register-msg-rp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To reject the incoming PIM register messages on an RP router:

1. Configure the policy options.

   ```
   [edit]
   user@host# edit policy-options
   ```

2. Set the group address.

   ```
   [edit policy-options]
   user@host# set policy statement reject-pim-register-msg-rp from route-filter
     224.1.1.1/32 exact
   ```

3. Set the source address.

   ```
   [edit policy-options]
   ```

```
user@host# set policy statement reject-pim-register-msg-rp from
  source-address-filter 10.10.10.1/32 exact
```

4. Set the match action.

```
[edit policy-options]
user@host# set policy statement reject-pim-register-msg-rp then reject
```

5. Configure the protocol.

```
[edit]
user@host# edit protocols pim rp
```

6. Assign the policy.

```
[edit]
user@host# set rp-register-policy reject-pim-register-msg-rp
```

Results From configuration mode, confirm your configuration by entering the **show policy-options** and **show protocols pim** command. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show policy-options
  policy-statement reject-pim-register-msg-rp {
  from {
  route-filter 224.1.1.1/32 exact;
  source-address-filter 10.10.10.1/32 exact;
  }
  then reject;
}
[edit]
  user@host# show protocols pim
  rp {
  rp-register-policy reject-pim-register-msg-rp;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 279
- Verifying the IGMP Version on page 279
- Verifying the PIM Mode and Interface Configuration on page 279
- Verifying the PIM Register Messages on page 279

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Verifying the PIM Register Messages

Purpose Verify whether the rejected policy on the RP router is enabled.

Action From operational mode, enter the **show policy-options** and **show protocols pim** command.

See Also
- Example: Stopping Outgoing PIM Register Messages on a Designated Router on page 272
- Configuring Register Message Filters on a PIM RP and DR on page 279
- Multicast Configuration Overview on page 16
- *Verifying a Multicast Configuration*

Configuring Register Message Filters on a PIM RP and DR

PIM register messages are sent to the rendezvous point (RP) by a designated router (DR). When a source for a group starts transmitting, the DR sends unicast PIM register packets to the RP.

Register messages have the following purposes:

- Notify the RP that a source is sending to a group.
- Deliver the initial multicast packets sent by the source to the RP for delivery down the shortest-path tree (SPT).

The PIM RP keeps track of all active sources in a single PIM sparse mode domain. In some cases, you want more control over which sources an RP discovers, or which sources a

DR notifies other RPs about. A high degree of control over PIM register messages is provided by RP or DR register message filtering. Message filtering prevents unauthorized groups and sources from registering with an RP router.

You configure RP or DR register message filtering to control the number and location of multicast sources that an RP discovers. You can apply register message filters on a DR to control outgoing register messages, or apply them on an RP to control incoming register messages.

When anycast RP is configured, all RPs in the anycast RP set need to be configured with the same register message filtering policy.

You can configure message filtering globally or for a routing instance. These examples show the global configuration.

To configure an RP filter to drop the register packets for multicast group range 224.1.1.0/24 from source address 10.10.94.2:

1. On the RP, configure the policy.

```
[edit policy-options policy-statement incoming-policy-for-rp from]
user@host# set route-filter 224.1.1.0/24 orlonger
user@host# set source-address-filter 10.10.94.2/32 exact
user@host# set then reject
user@host# exit
```

2. Apply the policy to the RP.

```
[edit protocols pim rp]
user@host# set rp-register-policy incoming-policy-for-rp
user@host# set local address 10.10.10.5
user@host# exit
```

To configure a DR filter to prevent sending register packets for group range 224.1.1.0/24 and source address 10.10.10.1/32:

1. On the DR, configure the policy.

```
[edit policy-options policy-statement outgoing-policy-for-rp]
user@host# set from route-filter 224.1.1.0/24 orlonger
user@host# set from source-address-filter 10.10.10.1/32 exact
user@host# set then reject
user@host# exit
```

2. Apply the policy to the DR.

   The static address is the address of the RP to which you do not want the DR to send the filtered register messages.

```
[edit protocols pim rp]
user@host# set dr-register-policy outgoing-policy-for-dr
user@host# set static 10.10.10.3
user@host# exit
```

To configure a policy expression to accept register messages for multicast group 224.1.1.5 but reject those for 224.1.1.1:

1. On the RP, configure the policies.

```
[edit policy-options policy-statement reject_224_1_1_1]
user@host# set from route-filter 224.1.1.0/24 orlonger
user@host# set from source-address-filter 10.10.94.2/32 exact
user@host# set then reject
user@host# exit

[edit policy-options policy-statement accept_224_1_1_5]
user@host# set term one from route-filter 224.1.1.5/32 exact
user@host# set term one from source-address-filter 10.10.94.2/32 exact
user@host# set term one then accept
user@host# set term two then reject
user@host# exit
```

2. Apply the policies to the RP.

```
[edit protocols pim rp]
user@host# set rp-register-policy [ reject_224_1_1_1 | accept_224_1_1_5 ]
user@host# set local address 10.10.10.5
```

To monitor the operation of the filters, run the **show pim statistics** command. The command output contains the following fields related to filtering:

- **RP Filtered Source**
- **Rx Joins/Prunes filtered**
- **Tx Joins/Prunes filtered**
- **Rx Register msgs filtering drop**
- **Tx Register msgs filtering drop**

See Also

- PIM Sparse Mode Source Registration on page 284
- Filtering RP and DR Register Messages on page 268
- **show pim statistics on page 1689**

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring PIM Dense Mode on page 203
- Configuring a Designated Router for PIM on page 301
- Example: Configuring Nonstop Active Routing for PIM on page 371
- Examples: Configuring PIM RPT and SPT Cutover on page 282
- Configuring PIM Sparse-Dense Mode on page 206

- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 357
- *Configuring Basic PIM Settings*

Examples: Configuring PIM RPT and SPT Cutover

Understanding Multicast Rendezvous Points, Shared Trees, and Rendezvous-Point Trees

In a shared tree, the root of the distribution tree is a router, not a host, and is located somewhere in the core of the network. In the primary sparse mode multicast routing protocol, Protocol Independent Multicast sparse mode (PIM SM), the core router at the root of the shared tree is the rendezvous point (RP). Packets from the upstream source and join messages from the downstream routers “rendezvous” at this core router.

In the RP model, other routers do not need to know the addresses of the sources for every multicast group. All they need to know is the IP address of the RP router. The RP router discovers the sources for all multicast groups.

The RP model shifts the burden of finding sources of multicast content from each router (the (S,G) notation) to the network (the (*,G) notation knows only the RP). Exactly how the RP finds the unicast IP address of the source varies, but there must be some method to determine the proper source for multicast content for a particular group.

Consider a set of multicast routers without any active multicast traffic for a certain group. When a router learns that an interested receiver for that group is on one of its directly connected subnets, the router attempts to join the distribution tree for that group back to the RP, not to the actual source of the content.

To join the shared tree, or *rendezvous-point tree* (*RPT*) as it is called in PIM sparse mode, the router must do the following:

- Determine the IP address of the RP for that group. Determining the address can be as simple as static configuration in the router, or as complex as a set of nested protocols.
- Build the shared tree for that group. The router executes an RPF check on the RP address in its routing table, which produces the interface closest to the RP. The router now detects that multicast packets from this RP for this group need to flow into the router on this RPF interface.

- Send a join message out on this interface using the proper multicast protocol (probably PIM sparse mode) to inform the upstream router that it wants to join the shared tree for that group. This message is a (*,G) join message because S is not known. Only the RP is known, and the RP is not actually the source of the multicast packets. The router receiving the (*,G) join message adds the interface on which the message was received to its outgoing interface list (OIL) for the group and also performs an RPF check on the RP address. The upstream router then sends a (*,G) join message out from the RPF interface toward the source, informing the upstream router that it also wants to join the group.

Each upstream router repeats this process, propagating join messages from the RPF interface, building the shared tree as it goes. The process stops when the join message reaches one of the following:

- The RP for the group that is being joined
- A router along the RPT that already has a multicast forwarding state for the group that is being joined

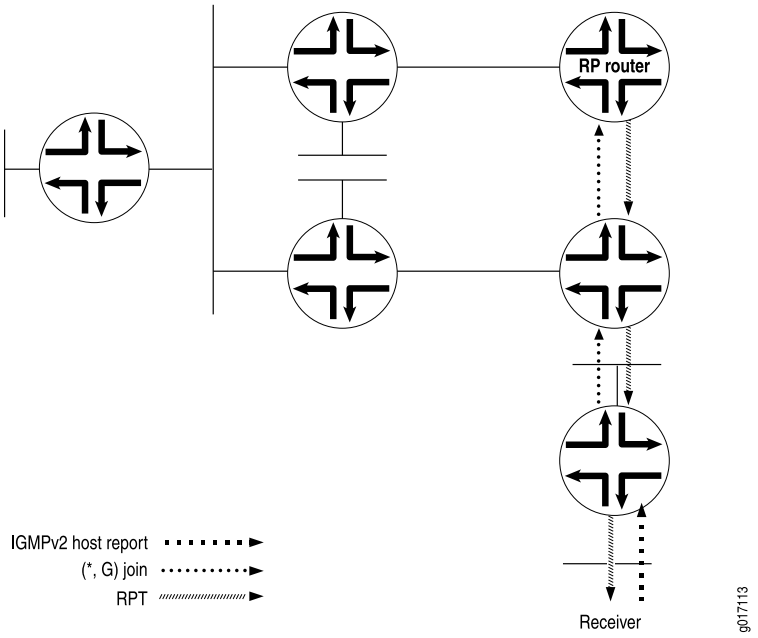
In either case, the branch is created, and packets can flow from the source to the RP and from the RP to the receiver. Note that there is no guarantee that the shared tree (RPT) is the shortest path tree to the source. Most likely it is not. However, there are ways to "migrate" a shared tree to an SPT once the flow of packets begins. In other words, the forwarding state can transition from (*,G) to (S,G). The formation of both types of tree depends heavily on the operation of the RPF check and the RPF table. For more information about the RPF table, see "Understanding Multicast Reverse Path Forwarding" on page 803.

Building an RPT Between the RP and Receivers

The RPT is the path between the RP and receivers (hosts) in a multicast group (see Figure 35 on page 284). The RPT is built by means of a PIM join message from a receiver's DR:

1. A receiver sends a request to join group (G) in an Internet Group Management Protocol (IGMP) host membership report. A PIM sparse-mode router, the receiver's DR, receives the report on a directly attached subnet and creates an RPT branch for the multicast group of interest.
2. The receiver's DR sends a PIM join message to its RPF neighbor, the next-hop address in the RPF table, or the unicast routing table.
3. The PIM join message travels up the tree and is multicast to the ALL-PIM-ROUTERS group (224.0.0.13). Each router in the tree finds its RPF neighbor by using either the RPF table or the unicast routing table. This is done until the message reaches the RP and forms the RPT. Routers along the path set up the multicast forwarding state to forward requested multicast traffic back down the RPT to the receiver.

Figure 35: Building an RPT Between the RP and the Receiver



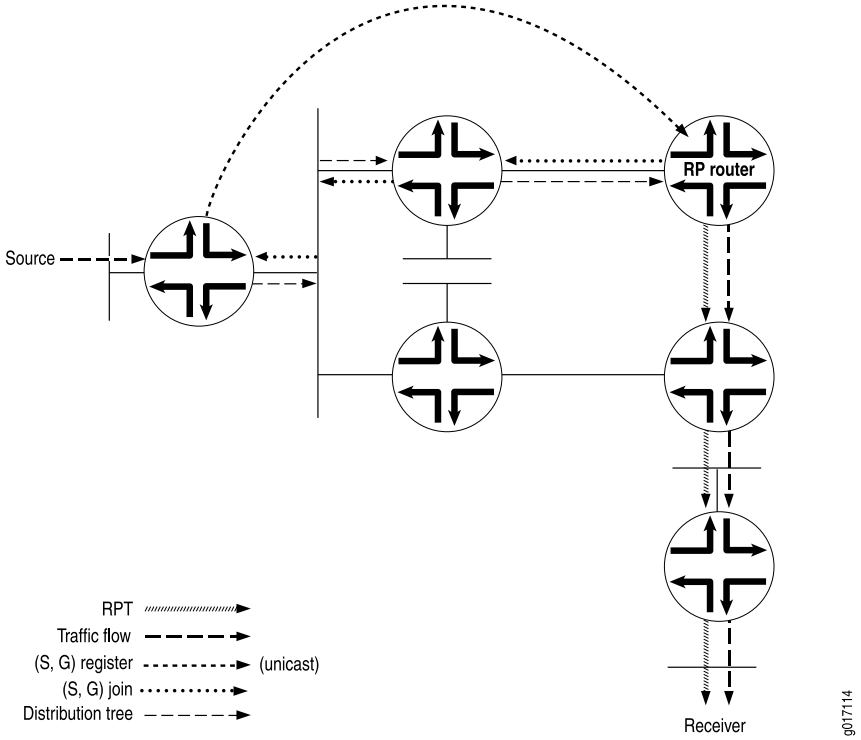
PIM Sparse Mode Source Registration

The RPT is a unidirectional tree, permitting traffic to flow down from the RP to the receiver in one direction. For multicast traffic to reach the receiver from the source, another branch of the distribution tree, called the shortest-path tree, needs to be built from the source's DR to the RP.

The shortest-path tree is created in the following way:

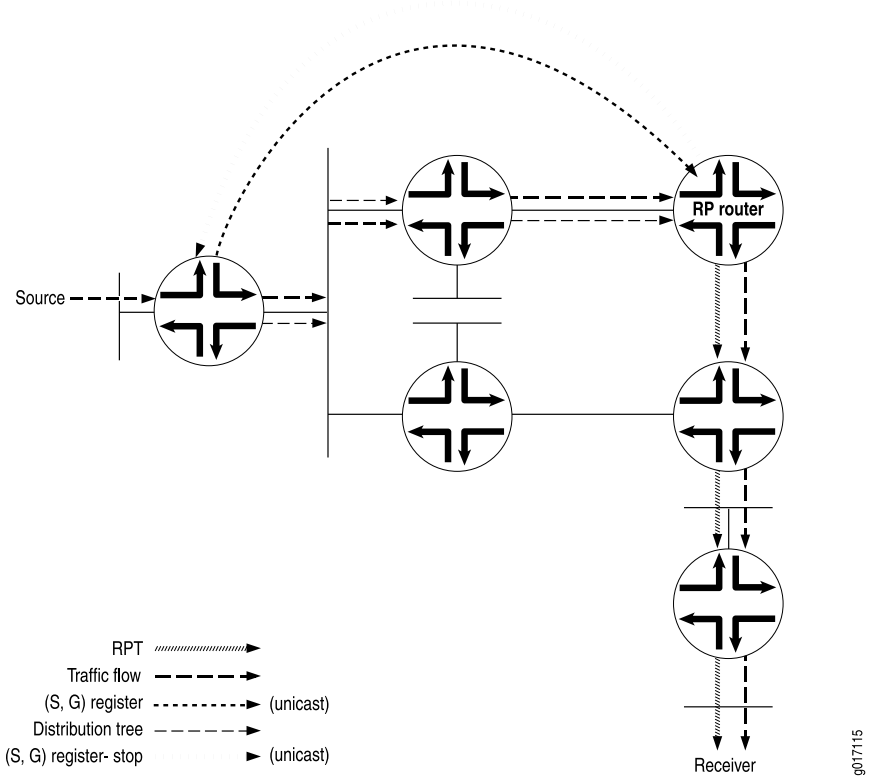
1. The source becomes active, sending out multicast packets on the LAN to which it is attached. The source's DR receives the packets and encapsulates them in a PIM register message, which it sends to the RP router (see Figure 36 on page 285).
2. When the RP router receives the PIM register message from the source, it sends a PIM join message back to the source.

Figure 36: PIM Register Message and PIM Join Message Exchanged



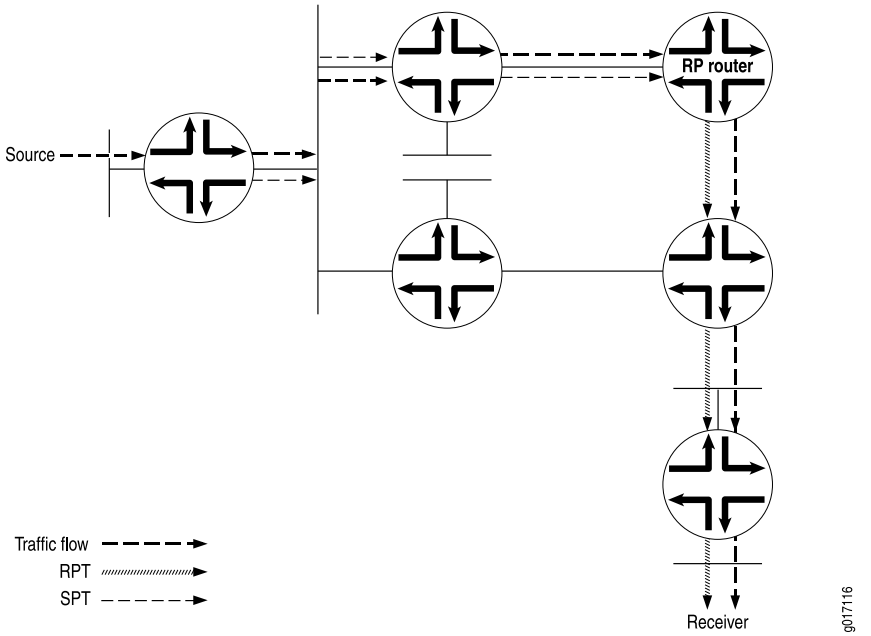
3. The source's DR receives the PIM join message and begins sending traffic down the SPT toward the RP router (see Figure 37 on page 286).
4. Once traffic is received by the RP router, it sends a register stop message to the source's DR to stop the register process.

Figure 37: Traffic Sent from the Source to the RP Router



5. The RP router sends the multicast traffic down the RPT toward the receiver (see Figure 38 on page 286).

Figure 38: Traffic Sent from the RP Router Toward the Receiver



Multicast Shortest-Path Tree

The distribution tree used for multicast is rooted at the source and is the shortest-path tree (SPT) as well. Consider a set of multicast routers without any active multicast traffic for a certain group (that is, they have no multicast forwarding state for that group). When a router learns that an interested receiver for that group is on one of its directly connected subnets, the router attempts to join the tree for that group.

To join the distribution tree, the router determines the unicast IP address of the source for that group. This address can be a simple static configuration on the router, or as complex as a set of protocols.

To build the SPT for that group, the router executes an a reverse path forwarding (RPF) check on the source address in its routing table. The RPF check produces the interface closest to the source, which is where multicast packets from this source for this group need to flow into the router.

The router next sends a join message out on this interface using the proper multicast protocol to inform the upstream router that it wants to join the distribution tree for that group. This message is an (S,G) join message because both S and G are known. The router receiving the (S,G) join message adds the interface on which the message was received to its output interface list (OIL) for the group and also performs an RPF check on the source address. The upstream router then sends an (S,G) join message out on the RPF interface toward the source, informing the upstream router that it also wants to join the group.

Each upstream router repeats this process, propagating joins out on the RPF interface, building the SPT as it goes. The process stops when the join message does one of two things:

- Reaches the router directly connected to the host that is the source.
- Reaches a router that already has multicast forwarding state for this source-group pair.

In either case, the branch is created, each of the routers has multicast forwarding state for the source-group pair, and packets can flow down the distribution tree from source to receiver. The RPF check at each router makes sure that the tree is an SPT.

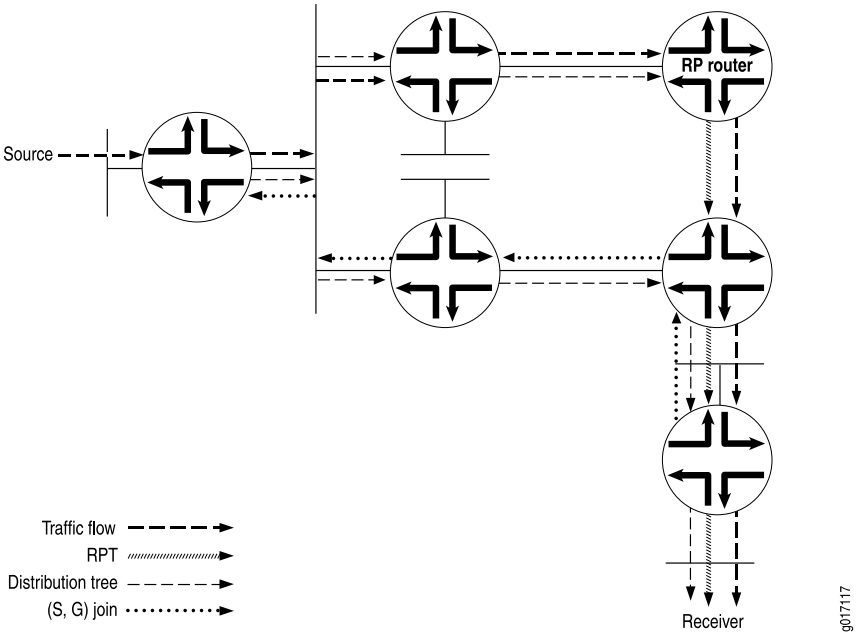
SPTs are always the shortest path, but they are not necessarily short. That is, sources and receivers tend to be on the periphery of a router network, not on the backbone, and multicast distribution trees have a tendency to sprawl across almost every router in the network. Because multicast traffic can overwhelm a slow interface, and one packet can easily become a hundred or a thousand on the opposite side of the backbone, it makes sense to provide a shared tree as a distribution tree so that the multicast source can be located more centrally in the network, on the backbone. This sharing of distribution trees with roots in the core network is accomplished by a multicast rendezvous point. For more information about RPs, see “Understanding Multicast Rendezvous Points, Shared Trees, and Rendezvous-Point Trees” on page 282.

SPT Cutover

Instead of continuing to use the SPT to the RP and the RPT toward the receiver, a direct SPT is created between the source and the receiver in the following way:

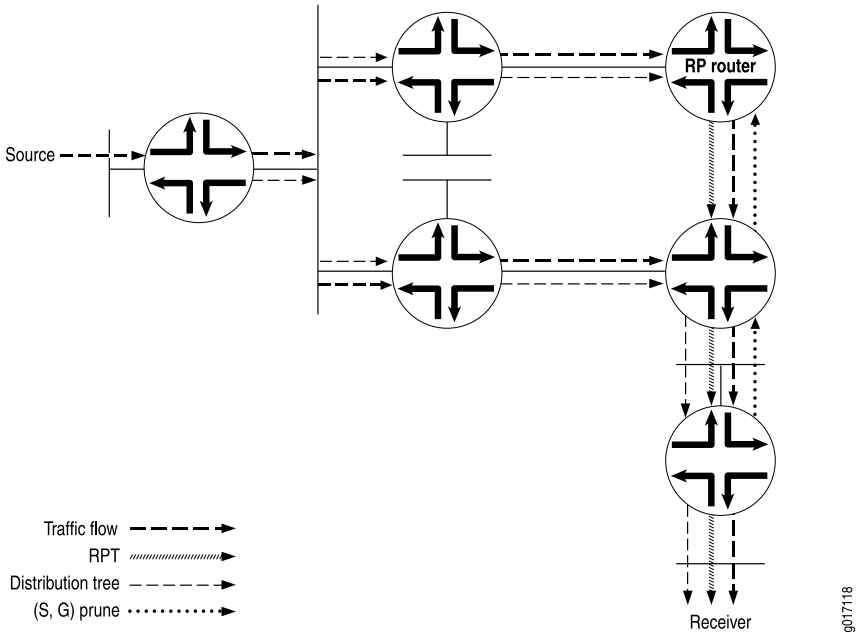
1. Once the receiver's DR receives the first multicast packet from the source, the DR sends a PIM join message to its RPF neighbor (see Figure 39 on page 288).
2. The source's DR receives the PIM join message, and an additional (S,G) state is created to form the SPT.
3. Multicast packets from that particular source begin coming from the source's DR and flowing down the new SPT to the receiver's DR. The receiver's DR is now receiving two copies of each multicast packet sent by the source—one from the RPT and one from the new SPT.

Figure 39: Receiver DR Sends a PIM Join Message to the Source



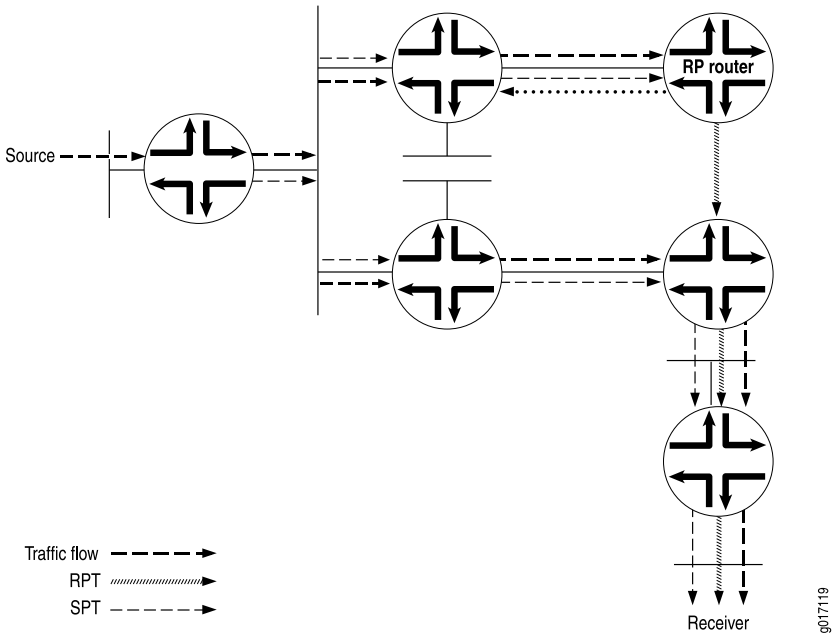
4. To stop duplicate multicast packets, the receiver's DR sends a PIM prune message toward the RP router, letting it know that the multicast packets from this particular source coming in from the RPT are no longer needed (see Figure 40 on page 289).

Figure 40: PIM Prune Message Is Sent from the Receiver's DR Toward the RP Router



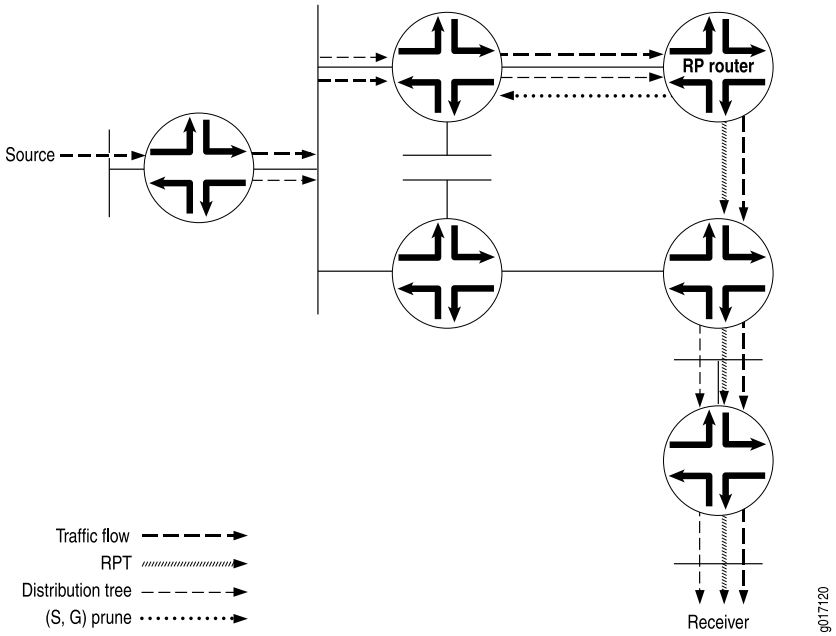
5. The PIM prune message is received by the RP router, and it stops sending multicast packets down to the receiver's DR. The receiver's DR is getting multicast packets only for this particular source over the new SPT. However, multicast packets from the source are still arriving from the source's DR toward the RP router (see Figure 41 on page 289).

Figure 41: RP Router Receives PIM Prune Message



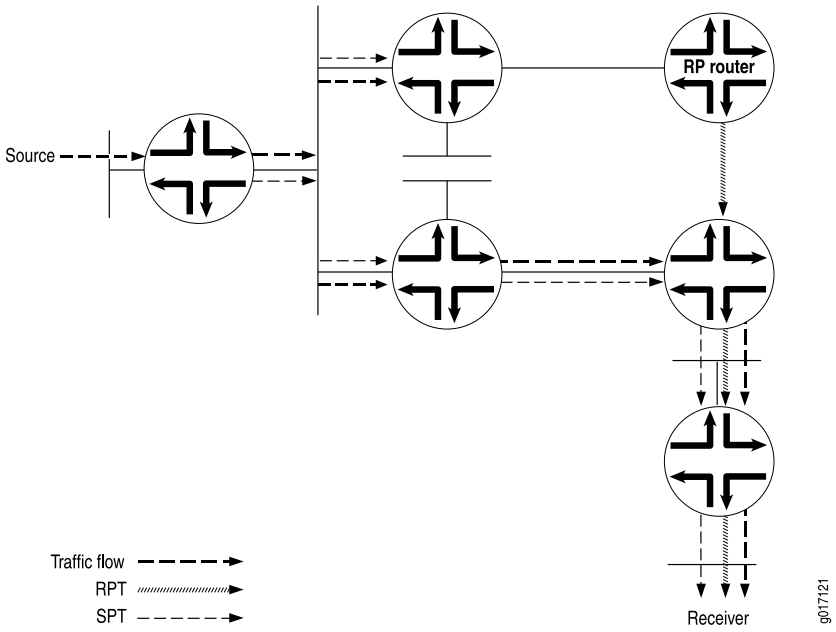
6. To stop the unneeded multicast packets from this particular source, the RP router sends a PIM prune message to the source's DR (see Figure 42 on page 290).

Figure 42: RP Router Sends a PIM Prune Message to the Source DR



7. The receiver's DR now receives multicast packets only for the particular source from the SPT (see Figure 43 on page 290).

Figure 43: Source's DR Stops Sending Duplicate Multicast Packets Toward the RP Router



SPT Cutover Control

In some cases, the last-hop router needs to stay on the shared tree to the RP and not transition to a direct SPT to the source. You might not want the last-hop router to

transition when, for example, a low-bandwidth multicast stream is forwarded from the RP to a last-hop router. All routers between last hop and source must maintain and refresh the SPT state. This can become a resource-intensive activity that does not add much to the network efficiency for a particular pair of source and multicast group addresses.

In these cases, you configure an SPT threshold policy on the last-hop router to control the transition to a direct SPT. An SPT cutover threshold of infinity applied to a source-group address pair means the last-hop router will never transition to a direct SPT. For all other source-group address pairs, the last-hop router transitions immediately to a direct SPT rooted at the source DR.

Example: Configuring the PIM Assert Timeout

This example shows how to configure the timeout period for a PIM assert forwarder.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 217.

Overview

The role of PIM assert messages is to determine the forwarder on a network with multiple routers. The forwarder is the router that forwards multicast packets to a network with multicast group members. The forwarder is generally the same as the PIM DR.

A router sends an assert message when it receives a multicast packet on an interface that is listed in the outgoing interface list of the matching routing entry. Receiving a message on an outgoing interface is an indication that more than one router forwards the same multicast packets to a network.

In Figure 44 on page 292, both routing devices R1 and R2 forward multicast packets for the same (S,G) entry on a network. Both devices detect this situation and both devices send assert messages on the Ethernet network. An assert message contains, in addition to a source address and group address, a unicast cost metric for sending packets to the source, and a preference metric for the unicast cost. The preference metric expresses a preference between unicast routing protocols. The routing device with the smallest preference metric becomes the forwarder (also called the assert winner). If the preference metrics are equal, the device that sent the lowest unicast cost metric becomes the forwarder. If the unicast metrics are also equal, the routing device with the highest IP

address becomes the forwarder. After the transmission of assert messages, only the forwarder continues to forward messages on the network.

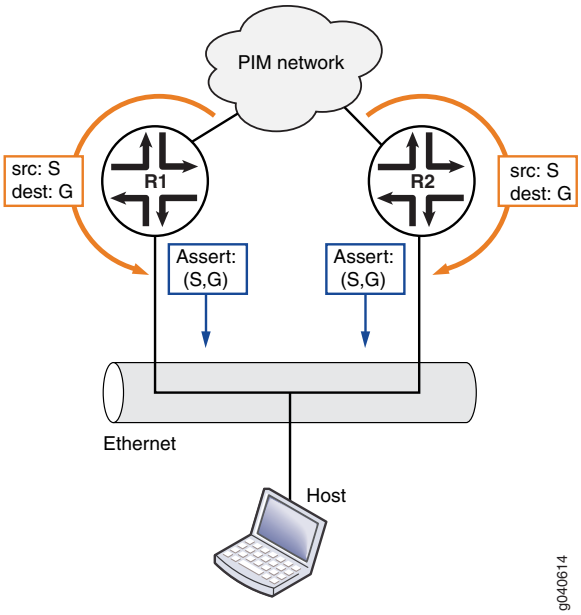
When an assert message is received and the RPF neighbor is changed to the assert winner, the assert timer is set to an assert timeout period. The assert timeout period is restarted every time a subsequent assert message for the route entry is received on the incoming interface. When the assert timer expires, the routing device resets its RPF neighbor according to its unicast routing table. Then, if multiple forwarders still exist, the forwarders reenter the assert message cycle. In effect, the assert timeout period determines how often multicast routing devices enter a PIM assert message cycle.

The range is from 5 through 210 seconds. The default is 180 seconds.

Assert messages are useful for LANs that connect multiple routing devices and no hosts.

Figure 44 on page 292 shows the topology for this example.

Figure 44: PIM Assert Topology



Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an assert timeout:

1. Configure the timeout period, in seconds.

```
[edit protocols pim]
user@host# set assert-timeout 60
```

2. (Optional) Trace assert messages.

```
[edit protocols pim]
user@host# set traceoptions file PIM.log
user@host# set traceoptions flag assert detail
```

3. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

4. To verify the configuration, run the following commands:
   - show pim join
   - show pim statistics

See Also
- Configuring PIM Trace Options on page 192
- SPT Cutover on page 288
- SPT Cutover Control on page 290

Example: Configuring the PIM SPT Threshold Policy

This example shows how to apply a policy that suppresses the transition from the rendezvous-point tree (RPT) rooted at the RP to the shortest-path tree (SPT) rooted at the source.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 217.

Overview

Multicast routing devices running PIM sparse mode can forward the same stream of multicast packets onto the same LAN through an RPT rooted at the RP or through an SPT rooted at the source. In some cases, the last-hop routing device needs to stay on the shared RPT to the RP and not transition to a direct SPT to the source. Receiving the multicast data traffic on SPT is optimal but introduces more state in the network, which

might not be desirable in some multicast deployments. Ideally, low-bandwidth multicast streams can be forwarded on the RPT, and high-bandwidth streams can use the SPT. This example shows how to configure such a policy.

This example includes the following settings:

- **spt-threshold**—Enables you to configure an SPT threshold policy on the last-hop routing device to control the transition to a direct SPT. When you include this statement in the main PIM instance, the PE router stays on the RPT for control traffic.
- **infinity**—Applies an SPT cutover threshold of infinity to a source-group address pair, so that the last-hop routing device never transitions to a direct SPT. For all other source-group address pairs, the last-hop routing device transitions immediately to a direct SPT rooted at the source DR. This statement must reference a properly configured policy to set the SPT cutover threshold for a particular source-group pair to infinity. The use of values other than infinity for the SPT threshold is not supported. You can configure more than one policy.
- **policy-statement**—Configures the policy. The simplest type of SPT threshold policy uses a route filter and source address filter to specify the multicast group and source addresses and to set the SPT threshold for that pair of addresses to infinity. The policy is applied to the main PIM instance.

  This example sets the SPT transition value for the source-group pair 10.10.10.1 and 224.1.1.1 to infinity. When the policy is applied to the last-hop router, multicast traffic from this source-group pair never transitions to a direct SPT to the source. Traffic will continue to arrive through the RP. However, traffic for any other source-group address combination at this router transitions to a direct SPT to the source.

Note these points when configuring the SPT threshold policy:

- Configuration changes to the SPT threshold policy affect how the routing device handles the SPT transition.

Note these points when configuring the SPT threshold policy:

- Configuration changes to the SPT threshold policy affect how the routing device handles the SPT transition.

Note these points when configuring the SPT threshold policy:

- Configuration changes to the SPT threshold policy affect how the routing device handles the SPT transition.
- When the policy is configured for the first time, the routing device continues to transition to the direct SPT for the source-group address pair until the PIM-join state is cleared with the **clear pim join** command.
- If you do not clear the PIM-join state when you apply the infinity policy configuration for the first time, you must apply it before the PE router is brought up.

- When the policy is deleted for a source-group address pair for the first time, the routing device does not transition to the direct SPT for that source-group address pair until the PIM-join state is cleared with the **clear pim join** command.
- When the policy is changed for a source-group address pair for the first time, the routing device does not use the new policy until the PIM-join state is cleared with the **clear pim join** command.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
[edit]
set policy-options policy-statement spt-infinity-policy term one from route-filter
  224.1.1.1/32 exact
set policy-options policy-statement spt-infinity-policy term one from source-address-filter
  10.10.10.1/32 exact
set policy-options policy-statement spt-infinity-policy term one then accept
set policy-options policy-statement spt-infinity-policy term two then reject
set protocols pim spt-threshold infinity spt-infinity-policy
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the *CLI User Guide*.

To configure an SPT threshold policy:

1. Apply the policy.

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set spt-threshold infinity spt-infinity-policy
[edit protocols pim]
user@host# exit
```

2. Configure the policy.

```
[edit]
user@host# edit policy-options policy-statement spt-infinity-policy
[edit policy-options policy-statement spt-infinity-policy]
user@host# set term one from route-filter 224.1.1.1/32 exact
[edit policy-options policy-statement spt-infinity-policy]
user@host# set term one from source-address-filter 10.10.10.1/32 exact
[edit policy-options policy-statement spt-infinity-policy]
user@host# set term one then accept
[edit policy-options policy-statement spt-infinity-policy]
user@host# set term two then reject
[edit policy-options policy-statement spt-infinity-policy]
user@host# exit
policy-statement {
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

4. Clear the PIM join cache to force the configuration to take effect.

```
[edit]
user@host# run clear pim join
```

Results

Confirm your configuration by entering the **show policy-options** command and the **show protocols** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement spt-infinity-policy {
  term one {
    from {
      route-filter 224.1.1.1/32 exact;
      source-address-filter 10.10.10.1/32 exact;
    }
    then accept;
  }
  term two {
    then reject;
  }
}
```

```
user@host# show protocols
pim {
  spt-threshold {
    infinity spt-infinity-policy;
  }
}
```

Verification

To verify the configuration, run the **show pim join** command.

See Also

- SPT Cutover Control on page 290

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring PIM Dense Mode on page 203
- Configuring a Designated Router for PIM on page 301
- Configuring PIM Filtering on page 267
- Example: Configuring Nonstop Active Routing for PIM on page 371

- Configuring PIM Sparse-Dense Mode on page 206
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 357
- *Configuring Basic PIM Settings*

Disabling PIM

By default, when you enable the PIM protocol it applies to the specified interface only. To enable PIM for all interfaces, include the **all** parameter (for example, **set protocol pim interface all**). You can disable PIM at the protocol, interface, or family hierarchy levels.

The hierarchy in which you configure PIM is critical. In general, the most specific configuration takes precedence. However, if PIM is disabled at the protocol level, then any disable statements with respect to an interface or family are ignored.

For example, the order of precedence for disabling PIM on a particular interface family is:

1. If PIM is disabled at the **[edit protocols pim interface *interface-name* family]** hierarchy level, then PIM is disabled for that interface family.
2. If PIM is not configured at the **[edit protocols pim interface *interface-name* family]** hierarchy level, but is disabled at the **[edit protocols pim interface *interface-name*]** hierarchy level, then PIM is disabled for all families on the specified interface.
3. If PIM is not configured at either the **[edit protocols pim interface *interface-name* family]** hierarchy level or the **[edit protocols pim interface *interface-name*]** hierarchy level, but is disabled at the **[edit protocols pim]** hierarchy level, then the PIM protocol is disabled globally for all interfaces and all families.

The following sections describe how to disable PIM at the various hierarchy levels.

- Disabling the PIM Protocol on page 297
- Disabling PIM on an Interface on page 298
- Disabling PIM for a Family on page 299
- Disabling PIM for a Rendezvous Point on page 300

Disabling the PIM Protocol

You can explicitly disable the PIM protocol. Disabling the PIM protocol disables the protocol for all interfaces and all families. This is accomplished at the **[edit protocols pim]** hierarchy level:

```
[edit protocols]
pim {
   disable;
}
```

To disable the PIM protocol:

1. Include the **disable** statement.

```
user@host# set protocols pim disable
```

2. (Optional) Verify your configuration settings before committing them by using the **show protocols pim** command.

```
user@host# run show protocols pim
```

See Also
- disable (PIM) on page 1011
- pim on page 1198

Disabling PIM on an Interface

You can disable the PIM protocol on a per-interface basis. This is accomplished at the **[edit protocols pim interface *interface-name*]** hierarchy level:

```
[edit protocols]
pim {
   interface interface-name {
      disable;
   }
}
```

To disable PIM on an interface:

1. Include the **disable** statement.

```
user@host# set protocols pim interface fe-0/1/0 disable
```

2. (Optional) Verify your configuration settings before committing them by using the **show protocols pim** command.

```
user@host# run show protocols pim
```

See Also
- disable (PIM) on page 1011
- pim on page 1198

Disabling PIM for a Family

You can disable the PIM protocol on a per-family basis. This is accomplished at the **[edit protocols pim family]** hierarchy level:

```
[edit protocols]
pim {
    family inet {
        disable;
    }
    family inet6 {
        disable;
    }
}
```

To disable PIM for a family:

1. Include the **disable** statement.

   ```
   user@host# set protocols pim family inet disable
   user@host# set protocols pim family inet6 disable
   ```

2. (Optional) Verify your configuration settings before committing them by using the **show protocols pim** command.

   ```
   user@host# run show protocols pim
   ```

See Also

- disable (PIM) on page 1011
- family (Protocols PIM) on page 1031
- pim on page 1198

Disabling PIM for a Rendezvous Point

You can disable the PIM protocol for a rendezvous point (RP) on a per-family basis. This is accomplished at the **[edit protocols pim rp local family]** hierarchy level:

```
[edit protocols]
pim {
  rp {
    local {
      family inet {
        disable;
      }
      family inet6 {
        disable;
      }
    }
  }
}
```

To disable PIM for an RP family:

1. Use the **disable** statement.

   ```
   user@host# set protocols pim rp local family inet disable
   user@host# set protocols pim rp local family inet6 disable
   ```

2. (Optional) Verify your configuration settings before committing them by using the **show protocols pim** command.

   ```
   user@host# run show protocols pim
   ```

See Also

- family (Local RP) on page 1026
- pim on page 1198

CHAPTER 10

Configuring Designated Routers

Understanding Designated Routers

In a PIM sparse mode (PIM-SM) domain, there are two types of designated routers to consider:

- The receiver DR sends PIM join and PIM prune messages from the receiver network toward the RP.
- The source DR sends PIM register messages from the source network to the RP.

Neighboring PIM routers multicast periodic PIM hello messages to each other every 30 seconds (the default). The PIM hello message usually includes a holdtime value for the neighbor to use, but this is not a requirement. If the PIM hello message does not include a holdtime value, a default timeout value (in Junos OS, 105 seconds) is used. On receipt of a PIM hello message, a router stores the IP address and priority for that neighbor. If the DR priorities match, the router with the highest IP address is selected as the DR.

If a DR fails, a new one is selected using the same process of comparing IP addresses.

NOTE: In PIM dense mode (PIM-DM), a DR is elected by the same process that PIM-SM uses. However, the only time that a DR has any effect in PIM-DM is when IGMPv1 is used on the interface. (IGMPv2 is the default.) In this case, the DR also functions as the IGMP Query Router because IGMPv1 does not have a Query Router election mechanism.

Configuring a Designated Router for PIM

Configuring Interface Priority for PIM Designated Router Selection

A designated router (DR) sends periodic join messages and prune messages toward a group-specific rendezvous point (RP) for each group for which it has active members. When a Protocol Independent Multicast (PIM) router learns about a source, it originates a Multicast Source Discovery Protocol (MSDP) source-address message if it is the DR on the upstream interface.

By default, every PIM interface has an equal probability (priority 1) of being selected as the DR. Configuring the interface DR priority helps ensure that changing an IP address does not alter your forwarding model.

To configure the interface designated router priority:

1. This example shows the configuration for the routing instance. Configure the interface globally or in the routing instance.

   ```
   [edit routing-instances PIM.master protocols pim interface ge-0/0/0.0 family inet]
   user@host# set priority 5
   ```

2. Verify the configuration by checking the **Hello Option DR Priority** field in the output of the **show pim neighbors detail** command.

   ```
   user@host> show pim neighbors detail

   Instance: PIM.master
   Interface: ge-0/0/0.0
   Address: 192.168.195.37, IPv4, PIM v2, Mode: Sparse
   Hello Option Holdtime: 65535 seconds
   Hello Option DR Priority: 5
   Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
   Join Suppression supported
   Rx Join: Group Source Timeout
   225.1.1.1 192.168.195.78 0
   225.1.1.1 0

   Interface: lo0.0
   Address: 10.255.245.91, IPv4, PIM v2, Mode: Sparse
   Hello Option Holdtime: 65535 seconds
   Hello Option DR Priority: 1
   Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
   Join Suppression supported

   Interface: pd-6/0/0.32768
   Address: 0.0.0.0, IPv4, PIM v2, Mode: Sparse
   Hello Option Holdtime: 65535 seconds
   Hello Option DR Priority: 0
   Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
   Join Suppression supported
   ```

See Also

- Configuring PIM Designated Router Election on Point-to-Point Links on page 303
- Understanding PIM Sparse Mode on page 209
- show pim neighbors on page 1657

Configuring PIM Designated Router Election on Point-to-Point Links

To comply with the latest PIM drafts, enable designated router (DR) election on all PIM interfaces, including point-to-point (P2P) interfaces. (DR election is enabled by default on all other interfaces.) One of the two routers might join a multicast group on its P2P link interface. The DR on that link is responsible for initiating the relevant join messages.

To enable DR election on point-to-point interfaces:

1. On both point-to-point link routers, configure the router globally or in the routing instance. This example shows the configuration for the routing instance.

   ```
   [edit routing-instances PIM.master protocols pim]
   user@host# set dr-election-on-p2p
   ```

2. Verify the configuration by checking the **State** field in the output of the **show pim interfaces** command. The possible values for the **State** field are DR, NotDR, and P2P. When a point-to-point link interface is elected to be the DR, the interface state becomes DR instead of P2P.

3. If the **show pim interfaces** command continues to report the P2P state, consider running the **restart routing** command on both routers on the point-to-point link. Then recheck the state.

   CAUTION: Do not restart a software process unless specifically asked to do so by your Juniper Networks customer support representative. Restarting a software process during normal operation of a routing platform could cause interruption of packet forwarding and loss of data.

   ```
   [edit]
   user@host# run restart routing
   ```

See Also

- Understanding PIM Sparse Mode on page 209
- Configuring Interface Priority for PIM Designated Router Selection on page 302
- show pim interfaces on page 1639

Related Documentation

- Configuring PIM Auto-RP on page 258
- Configuring PIM Bootstrap Router on page 253
- Configuring PIM Dense Mode on page 203
- Configuring PIM Filtering on page 267
- Example: Configuring Nonstop Active Routing for PIM on page 371
- Examples: Configuring PIM RPT and SPT Cutover on page 282
- Examples: Configuring PIM Sparse Mode on page 212

- Configuring PIM Sparse-Dense Mode on page 206
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 357
- *Configuring Basic PIM Settings*

Configuring Interface Priority for PIM Designated Router Selection

A designated router (DR) sends periodic join messages and prune messages toward a group-specific rendezvous point (RP) for each group for which it has active members. When a Protocol Independent Multicast (PIM) router learns about a source, it originates a Multicast Source Discovery Protocol (MSDP) source-address message if it is the DR on the upstream interface.

By default, every PIM interface has an equal probability (priority 1) of being selected as the DR. Configuring the interface DR priority helps ensure that changing an IP address does not alter your forwarding model.

To configure the interface designated router priority:

1. This example shows the configuration for the routing instance. Configure the interface globally or in the routing instance.

   [edit routing-instances PIM.master protocols pim interface ge-0/0/0.0 family inet]
   user@host# set priority 5

2. Verify the configuration by checking the **Hello Option DR Priority** field in the output of the **show pim neighbors detail** command.

   user@host> show pim neighbors detail

```
Instance: PIM.master
Interface: ge-0/0/0.0
Address: 192.168.195.37, IPv4, PIM v2, Mode: Sparse
Hello Option Holdtime: 65535 seconds
Hello Option DR Priority: 5
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Join Suppression supported
Rx Join: Group Source Timeout
225.1.1.1 192.168.195.78 0
225.1.1.1 0

Interface: lo0.0
Address: 10.255.245.91, IPv4, PIM v2, Mode: Sparse
Hello Option Holdtime: 65535 seconds
Hello Option DR Priority: 1
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Join Suppression supported

Interface: pd-6/0/0.32768
Address: 0.0.0.0, IPv4, PIM v2, Mode: Sparse
Hello Option Holdtime: 65535 seconds
Hello Option DR Priority: 0
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Join Suppression supported
```

Related Documentation
- Configuring PIM Designated Router Election on Point-to-Point Links on page 303
- Understanding PIM Sparse Mode on page 209
- show pim neighbors on page 1657

Configuring PIM Designated Router Election on Point-to-Point Links

To comply with the latest PIM drafts, enable designated router (DR) election on all PIM interfaces, including point-to-point (P2P) interfaces. (DR election is enabled by default on all other interfaces.) One of the two routers might join a multicast group on its P2P link interface. The DR on that link is responsible for initiating the relevant join messages.

To enable DR election on point-to-point interfaces:

1. On both point-to-point link routers, configure the router globally or in the routing instance. This example shows the configuration for the routing instance.

   ```
   [edit routing-instances PIM.master protocols pim]
   user@host# set dr-election-on-p2p
   ```

2. Verify the configuration by checking the **State** field in the output of the **show pim interfaces** command. The possible values for the **State** field are DR, NotDR, and P2P. When a point-to-point link interface is elected to be the DR, the interface state becomes DR instead of P2P.

3. If the **show pim interfaces** command continues to report the P2P state, consider running the **restart routing** command on both routers on the point-to-point link. Then recheck the state.

   CAUTION: Do not restart a software process unless specifically asked to do so by your Juniper Networks customer support representative. Restarting a software process during normal operation of a routing platform could cause interruption of packet forwarding and loss of data.

   ```
   [edit]
   user@host# run restart routing
   ```

Related Documentation
- Understanding PIM Sparse Mode on page 209
- Configuring Interface Priority for PIM Designated Router Selection on page 302
- show pim interfaces on page 1639

CHAPTER 11

Receiving Content Directly from the Source with SSM

Understanding PIM Source-Specific Mode

PIM source-specific multicast (SSM) uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to allow a client to receive multicast traffic directly from the source. PIM SSM uses the PIM sparse-mode functionality to create an SPT between the receiver and the source, but builds the SPT without the help of an RP.

This topic covers:

Any Source Multicast (ASM) was the Original Multicast

RFC 1112, the original multicast RFC, supported both many-to-many and one-to-many models. These came to be known collectively as any-source multicast (ASM) because ASM allowed one or many sources for a multicast group's traffic. However, an ASM network must be able to determine the locations of all sources for a particular multicast

group whenever there are interested listeners, no matter where the sources might be located in the network. In ASM, the key function of *source discovery* is a required function of the network itself.

Source Discovery in Sparse Mode vs Dense Mode

Multicast source discovery appears to be an easy process, but in sparse mode it is not. In dense mode, it is simple enough to flood traffic to every router in the whole network so that every router learns the source address of the content for that multicast group. However, the flooding presents scalability and network resource use issues and is not a viable option in sparse mode.

PIM sparse mode (like any sparse mode protocol) achieves the required source discovery functionality without flooding at the cost of a considerable amount of complexity. RP routers must be added and must know all multicast sources, and complicated shared distribution trees must be built to the RPs.

PIM SSM is a Subset of PIM Sparse Mode

PIM SSM is simpler than PIM sparse mode because only the one-to-many model is supported. Initial commercial multicast Internet applications are likely to be available to *subscribers* (that is, receivers that issue join messages) from only a single source (a special case of SSM covers the need for a backup source). PIM SSM therefore forms a subset of PIM sparse mode. PIM SSM builds shortest-path trees (SPTs) rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic. That is, PIM SSM bypasses the RP connection stage through shared distribution trees, as in PIM sparse mode, and goes directly to the source-based distribution tree.

Why Use PIM SSM

In an environment where many sources come and go, such as for a videoconferencing service, ASM is appropriate. However, by ignoring the many-to-many model and focusing attention on the one-to-many source-specific multicast (SSM) model, several commercially promising multicast applications, such as television channel distribution over the Internet, might be brought to the Internet much more quickly and efficiently than if full ASM functionality were required of the network.

An SSM-configured network has distinct advantages over a traditionally configured PIM sparse-mode network. There is no need for shared trees or RP mapping (no RP is required), or for RP-to-RP source discovery through MSDP.

PIM SSM is simpler than PIM sparse mode because only the one-to-many model is supported. Initial commercial multicast Internet applications are likely to be available to *subscribers* (that is, receivers that issue join messages) from only a single source (a special case of SSM covers the need for a backup source). PIM SSM therefore forms a subset of PIM sparse mode. PIM SSM builds shortest-path trees (SPTs) rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic. That is, PIM SSM bypasses the RP connection stage through shared distribution trees, as in PIM sparse mode, and goes directly to the source-based distribution tree.

PIM Terminology

PIM SSM introduces new terms for many of the concepts in PIM sparse mode. PIM SSM can technically be used in the entire 224/4 multicast address range, although PIM SSM operation is guaranteed only in the 232/8 range (232.0.0/24 is reserved). The new SSM terms are appropriate for Internet video applications and are summarized in Table 13 on page 309.

Table 13: ASM and SSM Terminology

Term	Any-Source Multicast	Source-Specific Multicast
Address identifier	G	S,G
Address designation	group	channel
Receiver operations	join, leave	subscribe, unsubscribe
Group address range	224/4 excluding 232/8	224/4 (guaranteed only for 232/8)

Although PIM SSM describes receiver operations as *subscribe* and *unsubscribe*, the same PIM sparse mode join and leave messages are used by both forms of the protocol. The terminology change distinguishes ASM from SSM even though the receiver messages are identical.

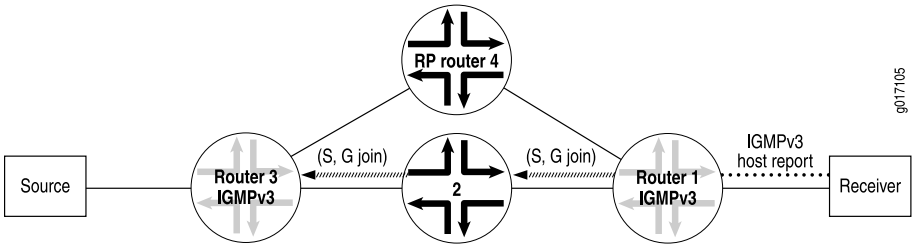
How PIM SSM Works

PIM source-specific multicast (SSM) uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to allow a client to receive multicast traffic directly from the source. PIM SSM uses the PIM sparse-mode functionality to create an SPT between the receiver and the source, but builds the SPT without the help of an RP.

By default, the SSM group multicast address is limited to the IP address range from 232.0.0.0 through 232.255.255.255. However, you can extend SSM operations into another Class D range by including the **ssm-groups** statement at the **[edit routing-options multicast]** hierarchy level. The default SSM address range from 232.0.0.0 through 232.255.255.255 cannot be used in the **ssm-groups** statement. This statement is for adding other multicast addresses to the default SSM group addresses. This statement does not override the default SSM group address range.

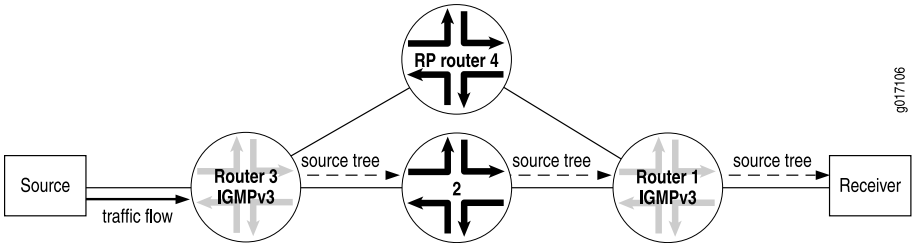
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3), announcing a desire to join group G and source S (see Figure 45 on page 310). The directly connected PIM sparse-mode router, the receiver's DR, sends an (S,G) join message to its RPF neighbor for the source. Notice in Figure 45 on page 310 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

Figure 45: Receiver Announces Desire to Join Group G and Source S



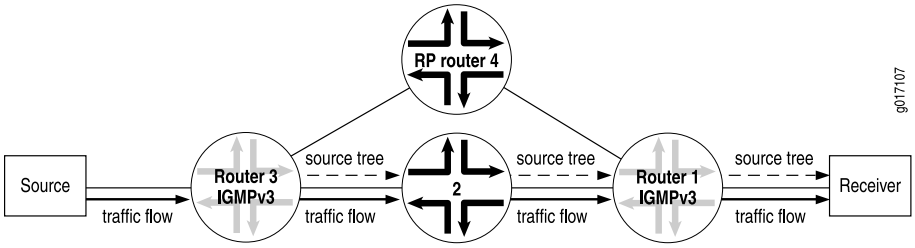
The (S,G) join message initiates the source tree and then builds it out hop by hop until it reaches the source. In Figure 46 on page 310, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 46: Router 3 (Last-Hop Router) Joins the Source Tree



Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 47 on page 310).

Figure 47: (S,G) State Is Built Between the Source and the Receiver



Using PIM SSM

You can configure Junos OS to accept any-source multicast (ASM) join messages (*,G) for group addresses that are within the default or configured range of source-specific multicast (SSM) groups. This allows you to support a mix of any-source and source-specific multicast groups simultaneously.

Deploying SSM is easy. You need to configure PIM sparse mode on all router interfaces and issue the necessary SSM commands, including specifying IGMPv3 on the receiver's LAN. If PIM sparse mode is not explicitly configured on both the source and group member interfaces, multicast packets are not forwarded. Source lists, supported in IGMPv3, are used in PIM SSM. As sources become active and start sending multicast packets, interested receivers in the SSM group receive the multicast packets.

To configure additional SSM groups, include the **ssm-groups** statement at the **[edit routing-options multicast]** hierarchy level.

Related Documentation

- Source-Specific Multicast Groups Overview on page 315
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316

Example: Configuring Source-Specific Multicast

Understanding PIM Source-Specific Mode

PIM source-specific multicast (SSM) uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to allow a client to receive multicast traffic directly from the source. PIM SSM uses the PIM sparse-mode functionality to create an SPT between the receiver and the source, but builds the SPT without the help of an RP.

This topic covers:

Any Source Multicast (ASM) was the Original Multicast

RFC 1112, the original multicast RFC, supported both many-to-many and one-to-many models. These came to be known collectively as any-source multicast (ASM) because ASM allowed one or many sources for a multicast group's traffic. However, an ASM network must be able to determine the locations of all sources for a particular multicast group whenever there are interested listeners, no matter where the sources might be located in the network. In ASM, the key function of *source discovery* is a required function of the network itself.

Source Discovery in Sparse Mode vs Dense Mode

Multicast source discovery appears to be an easy process, but in sparse mode it is not. In dense mode, it is simple enough to flood traffic to every router in the whole network so that every router learns the source address of the content for that multicast group. However, the flooding presents scalability and network resource use issues and is not a viable option in sparse mode.

PIM sparse mode (like any sparse mode protocol) achieves the required source discovery functionality without flooding at the cost of a considerable amount of complexity. RP routers must be added and must know all multicast sources, and complicated shared distribution trees must be built to the RPs.

PIM SSM is a Subset of PIM Sparse Mode

PIM SSM is simpler than PIM sparse mode because only the one-to-many model is supported. Initial commercial multicast Internet applications are likely to be available to *subscribers* (that is, receivers that issue join messages) from only a single source (a special case of SSM covers the need for a backup source). PIM SSM therefore forms a subset of PIM sparse mode. PIM SSM builds shortest-path trees (SPTs) rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic. That is, PIM SSM bypasses the RP connection stage through shared distribution trees, as in PIM sparse mode, and goes directly to the source-based distribution tree.

Why Use PIM SSM

In an environment where many sources come and go, such as for a videoconferencing service, ASM is appropriate. However, by ignoring the many-to-many model and focusing attention on the one-to-many source-specific multicast (SSM) model, several commercially promising multicast applications, such as television channel distribution over the Internet, might be brought to the Internet much more quickly and efficiently than if full ASM functionality were required of the network.

An SSM-configured network has distinct advantages over a traditionally configured PIM sparse-mode network. There is no need for shared trees or RP mapping (no RP is required), or for RP-to-RP source discovery through MSDP.

PIM SSM is simpler than PIM sparse mode because only the one-to-many model is supported. Initial commercial multicast Internet applications are likely to be available to *subscribers* (that is, receivers that issue join messages) from only a single source (a special case of SSM covers the need for a backup source). PIM SSM therefore forms a subset of PIM sparse mode. PIM SSM builds shortest-path trees (SPTs) rooted at the source immediately because in SSM, the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic. That is, PIM SSM bypasses the RP connection stage through shared distribution trees, as in PIM sparse mode, and goes directly to the source-based distribution tree.

PIM Terminology

PIM SSM introduces new terms for many of the concepts in PIM sparse mode. PIM SSM can technically be used in the entire 224/4 multicast address range, although PIM SSM

operation is guaranteed only in the 232/8 range (232.0.0/24 is reserved). The new SSM terms are appropriate for Internet video applications and are summarized in Table 13 on page 309.

Table 14: ASM and SSM Terminology

Term	Any-Source Multicast	Source-Specific Multicast
Address identifier	G	S,G
Address designation	group	channel
Receiver operations	join, leave	subscribe, unsubscribe
Group address range	224/4 excluding 232/8	224/4 (guaranteed only for 232/8)

Although PIM SSM describes receiver operations as *subscribe* and *unsubscribe*, the same PIM sparse mode join and leave messages are used by both forms of the protocol. The terminology change distinguishes ASM from SSM even though the receiver messages are identical.

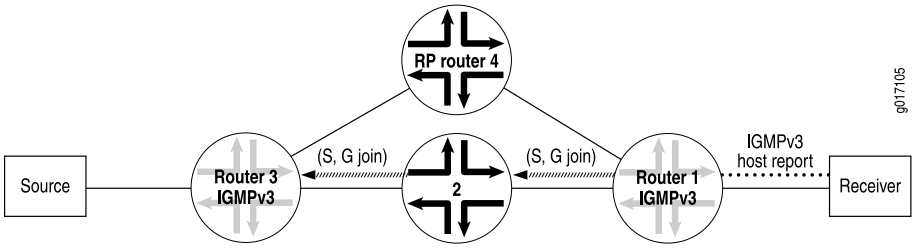
How PIM SSM Works

PIM source-specific multicast (SSM) uses a subset of PIM sparse mode and IGMP version 3 (IGMPv3) to allow a client to receive multicast traffic directly from the source. PIM SSM uses the PIM sparse-mode functionality to create an SPT between the receiver and the source, but builds the SPT without the help of an RP.

By default, the SSM group multicast address is limited to the IP address range from 232.0.0.0 through 232.255.255.255. However, you can extend SSM operations into another Class D range by including the **ssm-groups** statement at the **[edit routing-options multicast]** hierarchy level. The default SSM address range from 232.0.0.0 through 232.255.255.255 cannot be used in the **ssm-groups** statement. This statement is for adding other multicast addresses to the default SSM group addresses. This statement does not override the default SSM group address range.

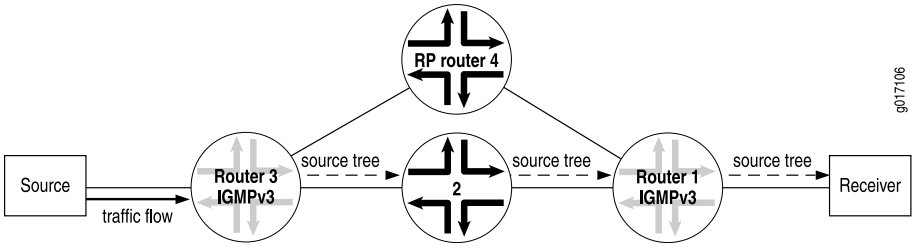
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3), announcing a desire to join group G and source S (see Figure 45 on page 310). The directly connected PIM sparse-mode router, the receiver's DR, sends an (S,G) join message to its RPF neighbor for the source. Notice in Figure 45 on page 310 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

Figure 48: Receiver Announces Desire to Join Group G and Source S



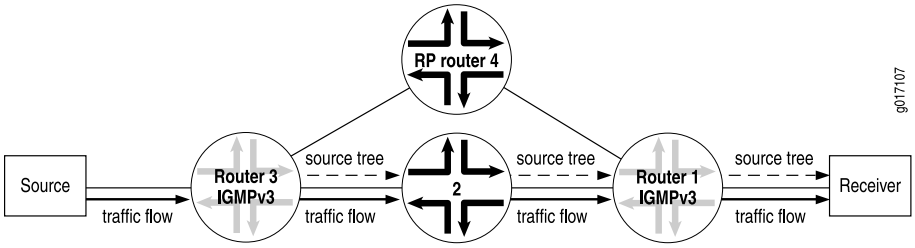
The (S,G) join message initiates the source tree and then builds it out hop by hop until it reaches the source. In Figure 46 on page 310, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 49: Router 3 (Last-Hop Router) Joins the Source Tree



Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 47 on page 310).

Figure 50: (S,G) State Is Built Between the Source and the Receiver



Using PIM SSM

You can configure Junos OS to accept any-source multicast (ASM) join messages (*,G) for group addresses that are within the default or configured range of source-specific multicast (SSM) groups. This allows you to support a mix of any-source and source-specific multicast groups simultaneously.

Deploying SSM is easy. You need to configure PIM sparse mode on all router interfaces and issue the necessary SSM commands, including specifying IGMPv3 on the receiver's LAN. If PIM sparse mode is not explicitly configured on both the source and group member interfaces, multicast packets are not forwarded. Source lists, supported in IGMPv3, are used in PIM SSM. As sources become active and start sending multicast packets, interested receivers in the SSM group receive the multicast packets.

To configure additional SSM groups, include the **ssm-groups** statement at the **[edit routing-options multicast]** hierarchy level.

See Also

- Source-Specific Multicast Groups Overview on page 315
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316

Source-Specific Multicast Groups Overview

Source-specific multicast (SSM) is a service model that identifies session traffic by both source and group address. SSM implemented in Junos OS has the efficient explicit join procedures of Protocol Independent Multicast (PIM) sparse mode but eliminates the immediate shared tree and rendezvous point (RP) procedures using (*,G) pairs. The (*) is a wildcard referring to any source sending to group G, and “G” refers to the IP multicast group. SSM builds shortest-path trees (SPTs) directly represented by (S,G) pairs. The “S” refers to the source's unicast IP address, and the “G” refers to the specific multicast group address. The SSM (S,G) pairs are called channels to differentiate them from any-source multicast (ASM) groups. Although ASM supports both one-to-many and many-to-many communications, ASM's complexity is in its method of source discovery. For example, if you click a link in a browser, the receiver is notified about the group information, but not the source information. With SSM, the client receives both source and group information.

SSM is ideal for one-to-many multicast services such as network entertainment channels. However, many-to-many multicast services might require ASM.

To deploy SSM successfully, you need an end-to-end multicast-enabled network and applications that use an Internet Group Management Protocol version 3 (IGMPv3) or Multicast Listener Discovery version 2 (MLDv2) stack, or you need to configure SSM mapping from IGMPv1 or IGMPv2 to IGMPv3. An IGMPv3 stack provides the capability of a host operating system to use the IGMPv3 protocol. IGMPv3 is available for Windows XP, Windows Vista, and most UNIX operating systems.

SSM mapping allows operators to support an SSM network without requiring all hosts to support IGMPv3. This support exists in static (S,G) configurations, but SSM mapping also supports dynamic per-source group state information, which changes as hosts join and leave the group using IGMP.

SSM is typically supported with a subset of IGMPv3 and PIM sparse mode known as *PIM SSM*. Using SSM, a client can receive multicast traffic directly from the source. PIM SSM uses the PIM sparse-mode functionality to create an SPT between the client and the source, but builds the SPT without the help of an RP.

An SSM-configured network has distinct advantages over a traditionally configured PIM sparse-mode network. There is no need for shared trees or RP mapping (no RP is required), or for RP-to-RP source discovery through the Multicast Source Discovery Protocol (MSDP).

Example: Configuring Source-Specific Multicast Groups with Any-Source Override

This example shows how to extend source-specific multicast (SSM) group operations beyond the default IP address range of 232.0.0.0 through 232.255.255.255. This example also shows how to accept any-source multicast (ASM) join messages (*,G) for group addresses that are within the default or configured range of SSM groups. This allows you to support a mix of any-source and source-specific multicast groups simultaneously.

Requirements

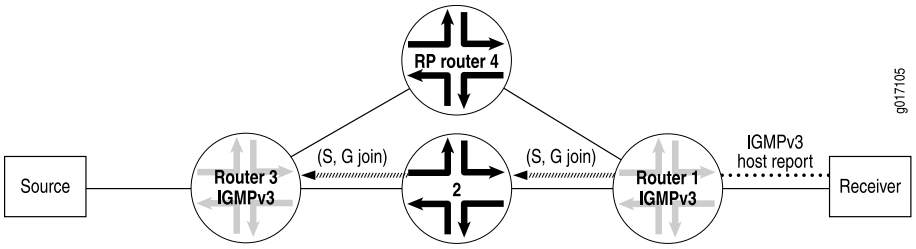
Before you begin, configure the router interfaces.

Overview

To deploy SSM, configure PIM sparse mode on all routing device interfaces and issue the necessary SSM commands, including specifying IGMPv3 or MLDv2 on the receiver's LAN. If PIM sparse mode is not explicitly configured on both the source and group members interfaces, multicast packets are not forwarded. Source lists, supported in IGMPv3 and MLDv2, are used in PIM SSM. Only sources that are specified send traffic to the SSM group.

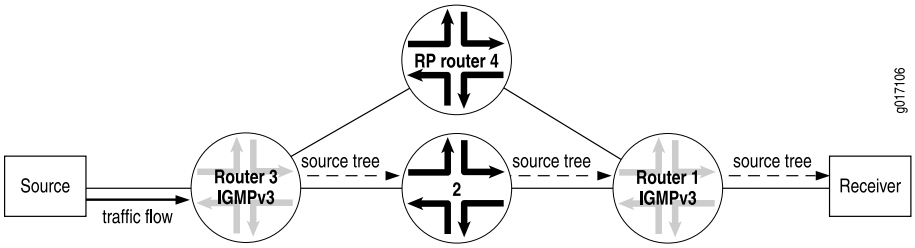
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3 or MLDv2) to join group G and source S (see Figure 51 on page 316). The directly connected PIM sparse-mode router, the receiver's designated router (DR), sends an (S,G) join message to its reverse-path forwarding (RPF) neighbor for the source. Notice in Figure 51 on page 316 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

Figure 51: Receiver Sends Messages to Join Group G and Source S



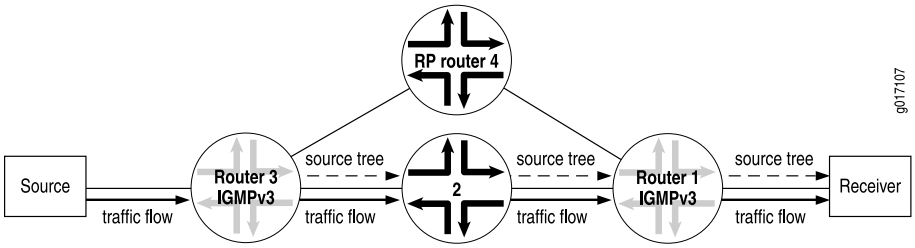
The (S,G) join message initiates the source tree and then builds it out hop by hop until it reaches the source. In Figure 52 on page 317, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 52: Router 3 (Last-Hop Router) Joins the Source Tree



Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 53 on page 317).

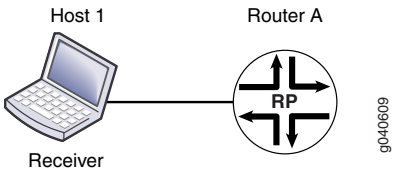
Figure 53: (S,G) State Is Built Between the Source and the Receiver



SSM can operate in include mode or in exclude mode. In exclude mode the receiver specifies a list of sources that it does not want to receive the multicast group traffic from. The routing device forwards traffic to the receiver from any source except the sources specified in the exclusion list. The receiver accepts traffic from any sources except the sources specified in the exclusion list.

This example works with the simple RPF topology shown in Figure 54 on page 317.

Figure 54: Simple RPF Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface all
set protocols pim rp local address 10.255.72.46
set protocols pim rp local group-ranges 239.0.0.0/24
set protocols pim interface fe-1/0/0.0 mode sparse
set protocols pim interface lo0.0 mode sparse
set routing-options multicast ssm-groups 232.0.0.0/8
```

```
set routing-options multicast ssm-groups 239.0.0.0/8
set routing-options multicast asm-override-ssm
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an RPF policy:

1. Configure OSPF.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface fxp0.0 disable
user@host# set area 0.0.0.0 interface all
```

2. Configure PIM sparse mode.

```
[edit protocols pim]
user@host# set rp local address 10.255.72.46
user@host# set rp local group-ranges 239.0.0.0/24
user@host# set interface fe-1/0/0.0 mode sparse
user@host# set interface lo0.0 mode sparse
```

3. Configure additional SSM groups.

```
[edit routing-options]
user@host# set ssm-groups [ 232.0.0.0/8 239.0.0.0/8 ]
```

4. Configure the RP to accept ASM join messages for groups within the SSM address range.

```
[edit routing-options]
user@host# set multicast asm-override-ssm
```

5. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show protocols** and **show routing-options** commands.

```
user@host# show protocols
ospf {
   area 0.0.0.0 {
      interface fxp0.0 {
         disable;
      }
      interface all;
   }
}
```

```
pim {
  rp {
    local {
      address 10.255.72.46;
      group-ranges {
        239.0.0.0/24;
      }
    }
  }
  interface fe-1/0/0.0 {
    mode sparse;
  }
  interface lo0.0 {
    mode sparse;
  }
}
```

```
user@host# show routing-options
multicast {
  ssm-groups [ 232.0.0.0/8 239.0.0.0/8 ];
  asm-override-ssm;
}
```

Verification

To verify the configuration, run the following commands:

- show igmp group
- **show igmp statistics**
- show pim join

See Also

- Source-Specific Multicast Groups Overview on page 315

Example: Configuring an SSM-Only Domain

Deploying an SSM-only domain is much simpler than deploying an ASM domain because it only requires a few configuration steps. Enable PIM sparse mode on all interfaces by adding the **mode** statement at the **[edit protocols pim interface all]** hierarchy level. When configuring all interfaces, exclude the **fxp0.0** management interface by adding the **disable** statement for that interface. Then configure IGMPv3 on all host-facing interfaces by adding the **version** statement at the **[edit protocols igmp interface *interface-name*]** hierarchy level.

In the following example, the host-facing interface is **fe-0/1/2**:

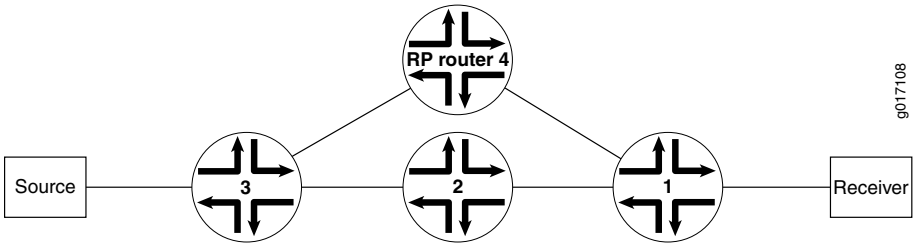
```
[edit]
protocols {
  pim {
    interface all {
      mode sparse;
      version 2;
    }
```

```
        interface fxp0.0 {
            disable;
        }
    }
    igmp {
        interface fe-0/1/2 {
            version 3;
        }
    }
}
```

Example: Configuring PIM SSM on a Network

The following example shows how PIM SSM is configured between a receiver and a source in the network illustrated in Figure 55 on page 320.

Figure 55: Network on Which to Configure PIM SSM



This example shows how to configure the IGMP version to IGMPv3 on all receiving host interfaces.

1. Enable IGMPv3 on all host-facing interfaces, and disable IGMP on the **fxp0.0** interface on Router 1.

   ```
   user@router1# set protocols igmp interface all version 3
   user@router1# set protocols igmp interface fxp0.0 disable
   ```

   NOTE: When you configure IGMPv3 on a router, hosts on interfaces configured with IGMPv2 cannot join the source tree.

2. After the configuration is committed, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

   ```
   user@router1> show configuration protocol igmp

   [edit protocols igmp]
   interface all {
       version 3;
   }
   interface fxp0.0 {
       disable;
   }
   ```

3. Use the **show igmp interface** command to verify that IGMP interfaces are configured.

```
user@router1> show igmp interface
Interface      State    Querier      Timeout Version Groups
fe-0/0/0.0     Up       198.51.100.245     213       3      0
fe-0/0/1.0     Up       198.51.100.241     220       3      0
fe-0/0/2.0     Up       198.51.100.237     218       3      0
Configured Parameters:
IGMP Query Interval (1/10 secs): 1250
IGMP Query Response Interval (1/10 secs): 100
IGMP Last Member Query Interval (1/10 secs): 10
IGMP Robustness Count: 2
Derived Parameters:
IGMP Membership Timeout (1/10 secs): 2600
IGMP Other Querier Present Timeout (1/10 secs): 2550
```

4. Use the **show pim join extensive** command to verify the PIM join state on Router 2 and Router 3 (the upstream routers).

```
user@router2> show pim join extensive
232.1.1.1       10.4.1.2                        sparse
     Upstream interface: fe-1/1/3.0
     Upstream State: Local Source
     Keepalive timeout: 209
     Downstream Neighbors:
        Interface: so-1/0/2.0
            10.10.71.1       State: Join    Flags: S    Timeout: 209
```

5. Use the **show pim join extensive** command to verify the PIM join state on Router 1 (the router connected to the receiver).

```
user@router1> show pim join extensive
232.1.1.1       10.4.1.2                        sparse
     Upstream interface: so-1/0/2.0
     Upstream State: Join to Source
     Keepalive timeout: 209
     Downstream Neighbors:
        Interface: fe-0/2/3.0
            10.3.1.1         State: Join    Flags: S    Timeout: Infinity
```

NOTE: IP version 6 (IPv6) multicast routers use the Multicast Listener Discovery (MLD) Protocol to manage the membership of hosts and routers in multicast groups and to learn which groups have interested listeners for each attached physical networks. Each routing device maintains a list of host multicast addresses that have listeners for each subnetwork, as well as a timer for each address. However, the routing device does not need to know the address of each listener—just the address of each host. The routing device provides addresses to the multicast routing protocol it uses, which ensures that multicast packets are delivered to all subnetworks where there are interested listeners. In this way, MLD is used as the transport for the Protocol Independent Multicast (PIM) Protocol. MLD is an integral part of IPv6 and must be enabled on all IPv6 routing devices and hosts that need to receive IP multicast traffic. The Junos OS supports MLD versions 1 and 2. Version 2 is supported for source-specific multicast (SSM) include and exclude modes.

See Also
- Example: Configuring SSM Mapping on page 322
-

Example: Configuring SSM Mapping

SSM mapping does not require that all hosts support IGMPv3. SSM mapping translates IGMPv1 or IGMPv2 membership reports to an IGMPv3 report. This enables hosts running IGMPv1 or IGMPv2 to participate in SSM until the hosts transition to IGMPv3.

SSM mapping applies to all group addresses that match the policy, not just those that conform to SSM addressing conventions (232/8 for IPv4, ff30::/32 through ff3F::/32 for IPv6).

We recommend separate SSM maps for IPv4 and IPv6 if both address families require SSM support. If you apply an SSM map containing both IPv4 and IPv6 addresses to an interface in an IPv4 context (using IGMP), only the IPv4 addresses in the list are used. If there are no such addresses, no action is taken. Similarly, if you apply an SSM map containing both IPv4 and IPv6 addresses to an interface in an IPv6 context (using MLD), only the IPv6 addresses in the list are used. If there are no such addresses, no action is taken.

In this example, you create a policy to match the group addresses that you want to translate to IGMPv3. Then you define the SSM map that associates the policy with the source addresses where these group addresses are found. Finally, you apply the SSM map to one or more IGMP (for IPv4) or MLD (for IPv6) interfaces.

1. Create an SSM policy named **ssm-policy-example**. The policy terms match the IPv4 SSM group address 232.1.1.1/32 and the IPv6 SSM group address ff35::1/128. All other addresses are rejected.

```
user@router1# set policy-options policy-statement ssm-policy-example term A from
  route-filter 232.1.1.1/32 exact
user@router1# set policy-options policy-statement ssm-policy-example term A then
  accept
user@router1# set policy-options policy-statement ssm-policy-example term B from
  route-filter ff35::1/128 exact
user@router1# set policy-options policy-statement ssm-policy-example term B then
  accept
```

2. After the configuration is committed, use the **show configuration policy-options** command to verify the policy configuration.

```
user@host> show configuration policy-options

[edit policy-options]
policy-statement ssm-policy-example {
  term A {
    from {
      route-filter 232.1.1.1/32 exact;
    }
    then accept;
  }
  term B {
```

```
    from {
      route-filter ff35::1/128 exact;
    }
    then accept;
  }
  then reject;
}
```

The group addresses must match the configured policy for SSM mapping to occur.

3. Define two SSM maps, one called **ssm-map-ipv6-example** and one called **ssm-map-ipv4-example**, by applying the policy and configuring the source addresses as a multicast routing option.

```
user@host# set routing-options multicast ssm-map ssm-map-ipv6-example policy
  ssm-policy-example
user@host# set routing-options multicast ssm-map ssm-map-ipv6-example source
  fec0::1 fec0::12
user@host# set routing-options multicast ssm-map ssm-map-ipv4-example policy
  ssm-policy-example
user@host# set routing-options multicast ssm-map ssm-map-ipv4-example source
  10.10.10.4
user@host# set routing-options multicast ssm-map ssm-map-ipv4-example source
  192.168.43.66
```

4. After the configuration is committed, use the **show configuration routing-options** command to verify the policy configuration.

```
user@host> show configuration routing-options
```

```
[edit routing-options]
multicast {
  ssm-map ssm-map-ipv6-example {
    policy ssm-policy-example;
    source [ fec0::1 fec0::12 ];
  }
  ssm-map ssm-map-ipv4-example {
    policy ssm-policy-example;
    source [ 10.10.10.4 192.168.43.66 ];
  }
}
```

We recommend separate SSM maps for IPv4 and IPv6.

5. Apply SSM maps for IPv4-to-IGMP interfaces and SSM maps for IPv6-to-MLD interfaces:

```
user@host# set protocols igmp interface fe-0/1/0.0 ssm-map ssm-map-ipv4-example
user@host# set protocols mld interface fe-0/1/1.0 ssm-map ssm-map-ipv6-example
```

6. After the configuration is committed, use the **show configuration protocol** command to verify the IGMP and MLD protocol configuration.

```
user@router1> show configuration protocol
```

```
[edit protocols]
igmp {
  interface fe-0/1/0.0 {
    ssm-map ssm-map-ipv4-example;
  }
}
mld {
  interface fe-/0/1/1.0 {
    ssm-map ssm-map-ipv6-example;
  }
}
```

7. Use the **show igmp interface** and the **show mld interface** commands to verify that the SSM maps are applied to the interfaces.

```
user@host> show igmp interface fe-0/1/0.0
Interface: fe-0/1/0.0
     Querier: 192.168.224.28
     State:         Up Timeout:    None Version:   2 Groups:   2
     SSM Map: ssm-map-ipv4-example

user@host> show mld interface fe-0/1/1.0
Interface: fe-0/1/1.0
     Querier: fec0:0:0:0:1::12
     State:         Up Timeout:    None Version:   2 Groups:   2
     SSM Map: ssm-map-ipv6-example
```

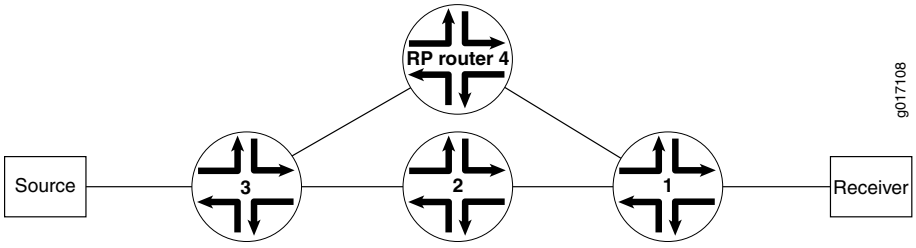
Related Documentation

- *Configuring Basic PIM Settings*

Example: Configuring PIM SSM on a Network

The following example shows how PIM SSM is configured between a receiver and a source in the network illustrated in Figure 55 on page 320.

Figure 56: Network on Which to Configure PIM SSM



This example shows how to configure the IGMP version to IGMPv3 on all receiving host interfaces.

1. Enable IGMPv3 on all host-facing interfaces, and disable IGMP on the **fxp0.0** interface on Router 1.

```
user@router1# set protocols igmp interface all version 3
user@router1# set protocols igmp interface fxp0.0 disable
```

NOTE: When you configure IGMPv3 on a router, hosts on interfaces configured with IGMPv2 cannot join the source tree.

2. After the configuration is committed, use the **show configuration protocol igmp** command to verify the IGMP protocol configuration.

```
user@router1> show configuration protocol igmp

[edit protocols igmp]
interface all {
    version 3;
}
interface fxp0.0 {
    disable;
}
```

3. Use the **show igmp interface** command to verify that IGMP interfaces are configured.

```
user@router1> show igmp interface
Interface      State    Querier      Timeout Version Groups
fe-0/0/0.0     Up       198.51.100.245     213       3      0
fe-0/0/1.0     Up       198.51.100.241     220       3      0
fe-0/0/2.0     Up       198.51.100.237     218       3      0
Configured Parameters:
IGMP Query Interval (1/10 secs): 1250
IGMP Query Response Interval (1/10 secs): 100
IGMP Last Member Query Interval (1/10 secs): 10
IGMP Robustness Count: 2
Derived Parameters:
IGMP Membership Timeout (1/10 secs): 2600
IGMP Other Querier Present Timeout (1/10 secs): 2550
```

4. Use the **show pim join extensive** command to verify the PIM join state on Router 2 and Router 3 (the upstream routers).

```
user@router2> show pim join extensive
232.1.1.1       10.4.1.2                        sparse
     Upstream interface: fe-1/1/3.0
     Upstream State: Local Source
     Keepalive timeout: 209
     Downstream Neighbors:
        Interface: so-1/0/2.0
            10.10.71.1       State: Join    Flags: S    Timeout: 209
```

5. Use the **show pim join extensive** command to verify the PIM join state on Router 1 (the router connected to the receiver).

```
user@router1> show pim join extensive
232.1.1.1       10.4.1.2                        sparse
     Upstream interface: so-1/0/2.0
     Upstream State: Join to Source
     Keepalive timeout: 209
     Downstream Neighbors:
```

```
Interface: fe-0/2/3.0
    10.3.1.1          State: Join   Flags: S    Timeout: Infinity
```

NOTE: IP version 6 (IPv6) multicast routers use the Multicast Listener Discovery (MLD) Protocol to manage the membership of hosts and routers in multicast groups and to learn which groups have interested listeners for each attached physical networks. Each routing device maintains a list of host multicast addresses that have listeners for each subnetwork, as well as a timer for each address. However, the routing device does not need to know the address of each listener—just the address of each host. The routing device provides addresses to the multicast routing protocol it uses, which ensures that multicast packets are delivered to all subnetworks where there are interested listeners. In this way, MLD is used as the transport for the Protocol Independent Multicast (PIM) Protocol. MLD is an integral part of IPv6 and must be enabled on all IPv6 routing devices and hosts that need to receive IP multicast traffic. The Junos OS supports MLD versions 1 and 2. Version 2 is supported for source-specific multicast (SSM) include and exclude modes.

Related Documentation

- Example: Configuring SSM Mapping on page 322
-

Example: Configuring an SSM-Only Domain

Deploying an SSM-only domain is much simpler than deploying an ASM domain because it only requires a few configuration steps. Enable PIM sparse mode on all interfaces by adding the **mode** statement at the **[edit protocols pim interface all]** hierarchy level. When configuring all interfaces, exclude the **fxp0.0** management interface by adding the **disable** statement for that interface. Then configure IGMPv3 on all host-facing interfaces by adding the **version** statement at the **[edit protocols igmp interface *interface-name*]** hierarchy level.

In the following example, the host-facing interface is **fe-0/1/2**:

```
[edit]
protocols {
  pim {
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
  igmp {
    interface fe-0/1/2 {
      version 3;
    }
```

```
    }
  }
```

Example: Configuring SSM Mapping

SSM mapping does not require that all hosts support IGMPv3. SSM mapping translates IGMPv1 or IGMPv2 membership reports to an IGMPv3 report. This enables hosts running IGMPv1 or IGMPv2 to participate in SSM until the hosts transition to IGMPv3.

SSM mapping applies to all group addresses that match the policy, not just those that conform to SSM addressing conventions (232/8 for IPv4, ff30::/32 through ff3F::/32 for IPv6).

We recommend separate SSM maps for IPv4 and IPv6 if both address families require SSM support. If you apply an SSM map containing both IPv4 and IPv6 addresses to an interface in an IPv4 context (using IGMP), only the IPv4 addresses in the list are used. If there are no such addresses, no action is taken. Similarly, if you apply an SSM map containing both IPv4 and IPv6 addresses to an interface in an IPv6 context (using MLD), only the IPv6 addresses in the list are used. If there are no such addresses, no action is taken.

In this example, you create a policy to match the group addresses that you want to translate to IGMPv3. Then you define the SSM map that associates the policy with the source addresses where these group addresses are found. Finally, you apply the SSM map to one or more IGMP (for IPv4) or MLD (for IPv6) interfaces.

1. Create an SSM policy named **ssm-policy-example**. The policy terms match the IPv4 SSM group address 232.1.1.1/32 and the IPv6 SSM group address ff35::1/128. All other addresses are rejected.

   ```
   user@router1# set policy-options policy-statement ssm-policy-example term A from
     route-filter 232.1.1.1/32 exact
   user@router1# set policy-options policy-statement ssm-policy-example term A then
     accept
   user@router1# set policy-options policy-statement ssm-policy-example term B from
     route-filter ff35::1/128 exact
   user@router1# set policy-options policy-statement ssm-policy-example term B then
     accept
   ```

2. After the configuration is committed, use the **show configuration policy-options** command to verify the policy configuration.

   ```
   user@host> show configuration policy-options

   [edit policy-options]
   policy-statement ssm-policy-example {
     term A {
       from {
         route-filter 232.1.1.1/32 exact;
       }
       then accept;
     }
     term B {
       from {
   ```

```
        route-filter ff35::1/128 exact;
      }
      then accept;
    }
    then reject;
  }
```

The group addresses must match the configured policy for SSM mapping to occur.

3. Define two SSM maps, one called **ssm-map-ipv6-example** and one called **ssm-map-ipv4-example**, by applying the policy and configuring the source addresses as a multicast routing option.

```
user@host# set routing-options multicast ssm-map ssm-map-ipv6-example policy
  ssm-policy-example
user@host# set routing-options multicast ssm-map ssm-map-ipv6-example source
  fec0::1 fec0::12
user@host# set routing-options multicast ssm-map ssm-map-ipv4-example policy
  ssm-policy-example
user@host# set routing-options multicast ssm-map ssm-map-ipv4-example source
  10.10.10.4
user@host# set routing-options multicast ssm-map ssm-map-ipv4-example source
  192.168.43.66
```

4. After the configuration is committed, use the **show configuration routing-options** command to verify the policy configuration.

```
user@host> show configuration routing-options

[edit routing-options]
multicast {
  ssm-map ssm-map-ipv6-example {
    policy ssm-policy-example;
    source [ fec0::1 fec0::12 ];
  }
  ssm-map ssm-map-ipv4-example {
    policy ssm-policy-example;
    source [ 10.10.10.4 192.168.43.66 ];
  }
}
```

We recommend separate SSM maps for IPv4 and IPv6.

5. Apply SSM maps for IPv4-to-IGMP interfaces and SSM maps for IPv6-to-MLD interfaces:

```
user@host# set protocols igmp interface fe-0/1/0.0 ssm-map ssm-map-ipv4-example
user@host# set protocols mld interface fe-0/1/1.0 ssm-map ssm-map-ipv6-example
```

6. After the configuration is committed, use the **show configuration protocol** command to verify the IGMP and MLD protocol configuration.

```
user@router1> show configuration protocol

[edit protocols]
```

```
igmp {
  interface fe-0/1/0.0 {
    ssm-map ssm-map-ipv4-example;
  }
}
mld {
  interface fe-/0/1/1.0 {
    ssm-map ssm-map-ipv6-example;
  }
}
```

7. Use the **show igmp interface** and the **show mld interface** commands to verify that the SSM maps are applied to the interfaces.

```
user@host> show igmp interface fe-0/1/0.0
Interface: fe-0/1/0.0
     Querier: 192.168.224.28
     State:         Up Timeout:    None Version:   2 Groups:   2
     SSM Map: ssm-map-ipv4-example

user@host> show mld interface fe-0/1/1.0
Interface: fe-0/1/1.0
     Querier: fec0:0:0:0:1::12
     State:         Up Timeout:    None Version:   2 Groups:   2
     SSM Map: ssm-map-ipv6-example
```

Example: Configuring Source-Specific Multicast Groups with Any-Source Override

This example shows how to extend source-specific multicast (SSM) group operations beyond the default IP address range of 232.0.0.0 through 232.255.255.255. This example also shows how to accept any-source multicast (ASM) join messages (*,G) for group addresses that are within the default or configured range of SSM groups. This allows you to support a mix of any-source and source-specific multicast groups simultaneously.

Requirements

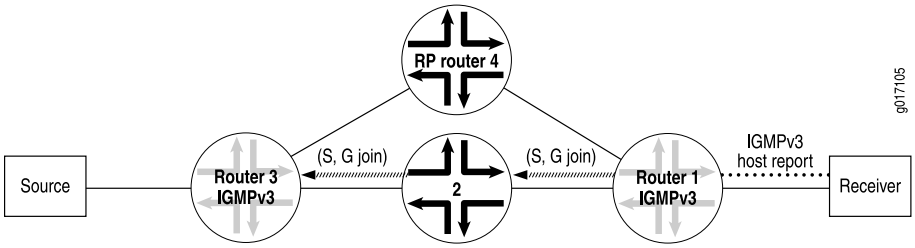
Before you begin, configure the router interfaces.

Overview

To deploy SSM, configure PIM sparse mode on all routing device interfaces and issue the necessary SSM commands, including specifying IGMPv3 or MLDv2 on the receiver's LAN. If PIM sparse mode is not explicitly configured on both the source and group members interfaces, multicast packets are not forwarded. Source lists, supported in IGMPv3 and MLDv2, are used in PIM SSM. Only sources that are specified send traffic to the SSM group.

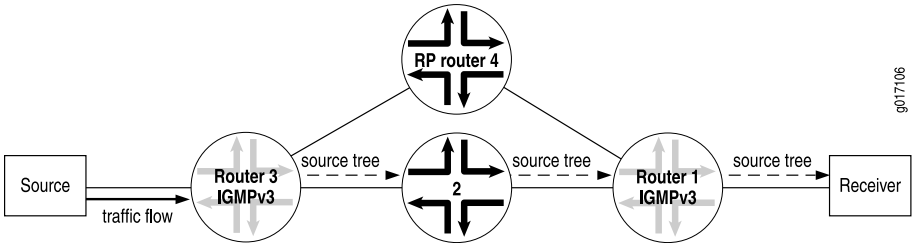
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3 or MLDv2) to join group G and source S (see Figure 51 on page 316). The directly connected PIM sparse-mode router, the receiver's designated router (DR), sends an (S,G) join message to its reverse-path forwarding (RPF) neighbor for the source. Notice in Figure 51 on page 316 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

Figure 57: Receiver Sends Messages to Join Group G and Source S



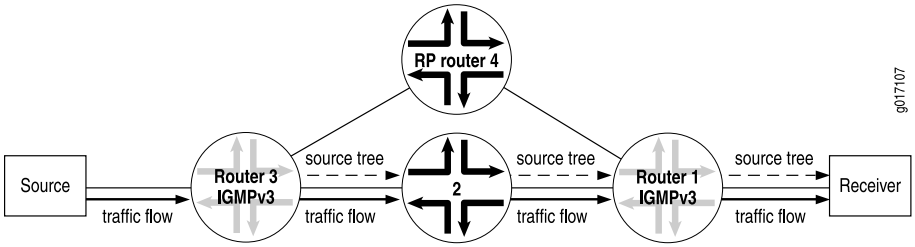
The (S,G) join message initiates the source tree and then builds it out hop by hop until it reaches the source. In Figure 52 on page 317, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 58: Router 3 (Last-Hop Router) Joins the Source Tree



Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 53 on page 317).

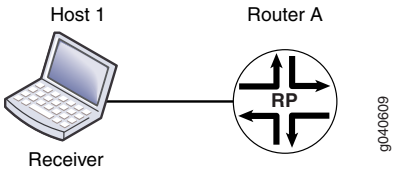
Figure 59: (S,G) State Is Built Between the Source and the Receiver



SSM can operate in include mode or in exclude mode. In exclude mode the receiver specifies a list of sources that it does not want to receive the multicast group traffic from. The routing device forwards traffic to the receiver from any source except the sources specified in the exclusion list. The receiver accepts traffic from any sources except the sources specified in the exclusion list.

This example works with the simple RPF topology shown in Figure 54 on page 317.

Figure 60: Simple RPF Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface all
set protocols pim rp local address 10.255.72.46
set protocols pim rp local group-ranges 239.0.0.0/24
set protocols pim interface fe-1/0/0.0 mode sparse
set protocols pim interface lo0.0 mode sparse
set routing-options multicast ssm-groups 232.0.0.0/8
set routing-options multicast ssm-groups 239.0.0.0/8
set routing-options multicast asm-override-ssm
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an RPF policy:

1. Configure OSPF.

   ```
   [edit protocols ospf]
   user@host# set area 0.0.0.0 interface fxp0.0 disable
   user@host# set area 0.0.0.0 interface all
   ```

2. Configure PIM sparse mode.

   ```
   [edit protocols pim]
   user@host# set rp local address 10.255.72.46
   user@host# set rp local group-ranges 239.0.0.0/24
   user@host# set interface fe-1/0/0.0 mode sparse
   user@host# set interface lo0.0 mode sparse
   ```

3. Configure additional SSM groups.

   ```
   [edit routing-options]
   user@host# set ssm-groups [ 232.0.0.0/8 239.0.0.0/8 ]
   ```

4. Configure the RP to accept ASM join messages for groups within the SSM address range.

```
[edit routing-options]
user@host# set multicast asm-override-ssm
```

5. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show protocols** and **show routing-options** commands.

```
user@host# show protocols
ospf {
  area 0.0.0.0 {
    interface fxp0.0 {
      disable;
    }
    interface all;
  }
}
pim {
  rp {
    local {
      address 10.255.72.46;
      group-ranges {
        239.0.0.0/24;
      }
    }
  }
  interface fe-1/0/0.0 {
    mode sparse;
  }
  interface lo0.0 {
    mode sparse;
  }
}
```

```
user@host# show routing-options
multicast {
  ssm-groups [ 232.0.0.0/8 239.0.0.0/8 ];
  asm-override-ssm;
}
```

Verification

To verify the configuration, run the following commands:

- show igmp group
- **show igmp statistics**

- show pim join

Related Documentation

- Source-Specific Multicast Groups Overview on page 315

Example: Configuring SSM Maps for Different Groups to Different Sources

- Multiple SSM Maps and Groups for Interfaces on page 333
- Example: Configuring Multiple SSM Maps Per Interface on page 333

Multiple SSM Maps and Groups for Interfaces

You can configure multiple source-specific multicast (SSM) maps so that different groups map to different sources, which enables a single multicast group to map to different sources for different interfaces.

See Also

- Example: Configuring Multiple SSM Maps Per Interface on page 333

Example: Configuring Multiple SSM Maps Per Interface

This example shows how to assign more than one SSM map to an IGMP interface.

Requirements

This example requires Junos OS Release 11.4 or later.

Overview

In this example, you configure a routing policy, POLICY-ipv4-example1, that maps multicast group join messages over an IGMP logical interface to IPv4 multicast source addresses based on destination IP address as follows:

Routing Policy Name	Multicast Group Join Messages for a Route Filter at This Destination Address	Multicast Source Addresses
POLICY-ipv4-example1 term 1	232.1.1.1	10.10.10.4, 192.168.43.66
POLICY-ipv4-example1 term 2	232.1.1.2	10.10.10.5, 192.168.43.67

You apply routing policy POLICY-ipv4-example1 to IGMP logical interface fe-0/1/0.0.

Configuration

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the *CLI User Guide*.

To configure this example, perform the following task:

CLI Quick Configuration

To quickly configure this example, copy the following configuration commands into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set policy-options policy-statement POLICY-ipv4-example1 term 1 from route-filter
  232.1.1.1/32 exact
set policy-options policy-statement POLICY-ipv4-example1 term 1 then ssm-source
  10.10.10.4
set policy-options policy-statement POLICY-ipv4-example1 term 1 then ssm-source
  192.168.43.66
set policy-options policy-statement POLICY-ipv4-example1 term 1 then accept
set policy-options policy-statement POLICY-ipv4-example1 term 2 from route-filter
  232.1.1.2/32 exact
set policy-options policy-statement POLICY-ipv4-example1 term 2 then ssm-source
  10.10.10.5
set policy-options policy-statement POLICY-ipv4-example1 term 2 then ssm-source
  192.168.43.67
set policy-options policy-statement POLICY-ipv4-example1 term 2 then accept
set protocols igmp interface fe-0/1/0.0 ssm-map-policy POLICY-ipv4-example1
```

Step-by-Step Procedure

To configure multiple SSM maps per interface:

1. Configure protocol-independent routing options for route filter 232.1.1.1, and specify the multicast source addresses to which matching multicast groups are to be mapped.

   ```
   [edit policy-options policy-statement POLICY-ipv4-example1 term 1]
   user@host# set from route-filter 232.1.1.1/32 exact
   user@host# set then ssm-source 10.10.10.4
   user@host# set then ssm-source 192.168.43.66
   user@host# set then accept
   ```

2. Configure protocol-independent routing options for route filter 232.1.1.2, and specify the multicast source addresses to which matching multicast groups are to be mapped.

   ```
   [edit policy-options policy-statement POLICY-ipv4-example1 term 2]
   user@host# set from route-filter 232.1.1.2/32 exact
   user@host# set then ssm-source 10.10.10.5
   user@host# set then ssm-source 192.168.43.67
   user@host# set then accept
   ```

3. Apply the policy map POLICY-ipv4-example1 to IGMP logical interface fe-0/1/1/0.

   ```
   [edit protocols igmp interface fe-0/1/0.0]
   user@host# set ssm-map-policy POLICY-ipv4-example1
   ```

Results After the configuration is committed, confirm the configuration by entering the **show policy-options** and **show protocols** configuration mode commands. If the command output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
user@host# show policy-options
policy-statement POLICY-ipv4-example1 {
  term 1 {
    from {
      route-filter 232.1.1.1/32 exact;
    }
    then {
      ssm-source [ 10.10.10.4 192.168.43.66 ];
      accept;
    }
  }
  term 2{
    from {
      route-filter 232.1.1.2/32 exact;
    }
    then {
      ssm-source [ 10.10.10.5 192.168.43.67 ];
      accept;
    }
  }
}

user@host# show protocols
igmp {
  interface fe-0/1/0.0 {
    ssm-map-policy POLICY-ipv4-example1;
  }
}
```

Verification

Confirm that the configuration is working properly.

- Displaying Information About IGMP-Enabled Interfaces on page 335
- Displaying the PIM Groups on page 336
- Displaying the Entries in the IP Multicast Forwarding Table on page 336

Displaying Information About IGMP-Enabled Interfaces

Purpose Verify that the SSM map policy POLICY-ipv4-example1 is applied to logical interface fe-0/1/0.0.

Action Use the **show igmp interface** operational mode command for the IGMP logical interface to which you applied the SSM map policy.

```
user@host> show igmp interface
Interface: fe-0/1/0.0
    Querier: 10.111.30.1
    State:         Up Timeout:    None Version:  2 Groups:      2
```

```
    SSM Map Policy: POLICY-ipv4-example1;

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

The command output displays the name of the IGMP logical interface (fe-0/1/0.0), which is the address of the routing device that has been elected to send membership queries and group information.

Displaying the PIM Groups

Purpose Verify the Protocol Independent Multicast (PIM) source and group pair (S,G) entries.

Action Use the **show pim join extensive 232.1.1.1** operational mode command to display the PIM source and group pair (S,G) entries for the 232.1.1.1 group.

Displaying the Entries in the IP Multicast Forwarding Table

Purpose Verify that the IP multicast forwarding table displays the multicast route state.

Action Use the **show multicast route extensive** operational mode command to display the entries in the IP multicast forwarding table to verify that the **Route state** is active and that the **Forwarding state** is forwarding.

Related Documentation

- Example: Configuring Source-Specific Multicast on page 311
- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 489

CHAPTER 12

Minimizing Routing State Information with Bidirectional PIM

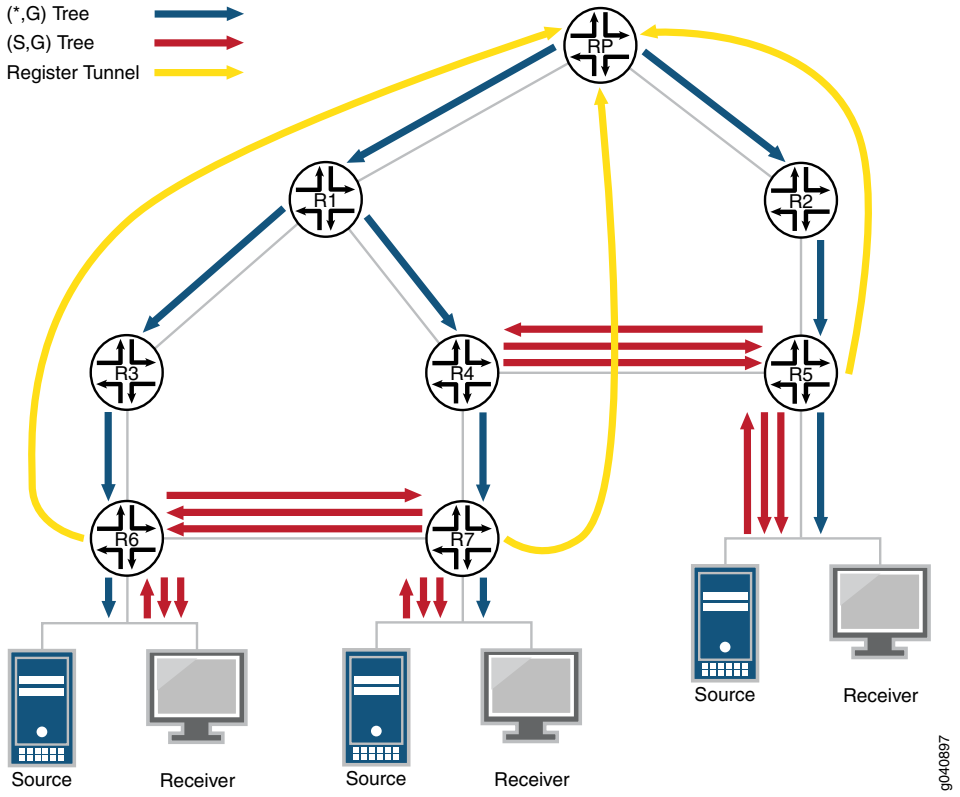
Example: Configuring Bidirectional PIM

Understanding Bidirectional PIM

Bidirectional PIM (PIM-Bidir) is specified by the IETF in RFC 5015, *Bidirectional Protocol Independent Multicast (BIDIR-PIM)*. It provides an alternative to other PIM modes, such as PIM sparse mode (PIM-SM), PIM dense mode (PIM-DM), and PIM source-specific multicast (SSM). In bidirectional PIM, multicast groups are carried across the network over bidirectional shared trees. This type of tree minimizes the amount of PIM routing state information that must be maintained, which is especially important in networks with numerous and dispersed senders and receivers. For example, one important application for bidirectional PIM is distributed inventory polling. In many-to-many applications, a multicast query from one station generates multicast responses from many stations. For each multicast group, such an application generates a large number of (S,G) routes for each station in PIM-SM, PIM-DM, or SSM. The problem is even worse in applications that use bursty sources, resulting in frequently changing multicast tables and, therefore, performance problems in routers.

Figure 61 on page 338 shows the traffic flows generated to deliver traffic for one group to and from three stations in a PIM-SM network.

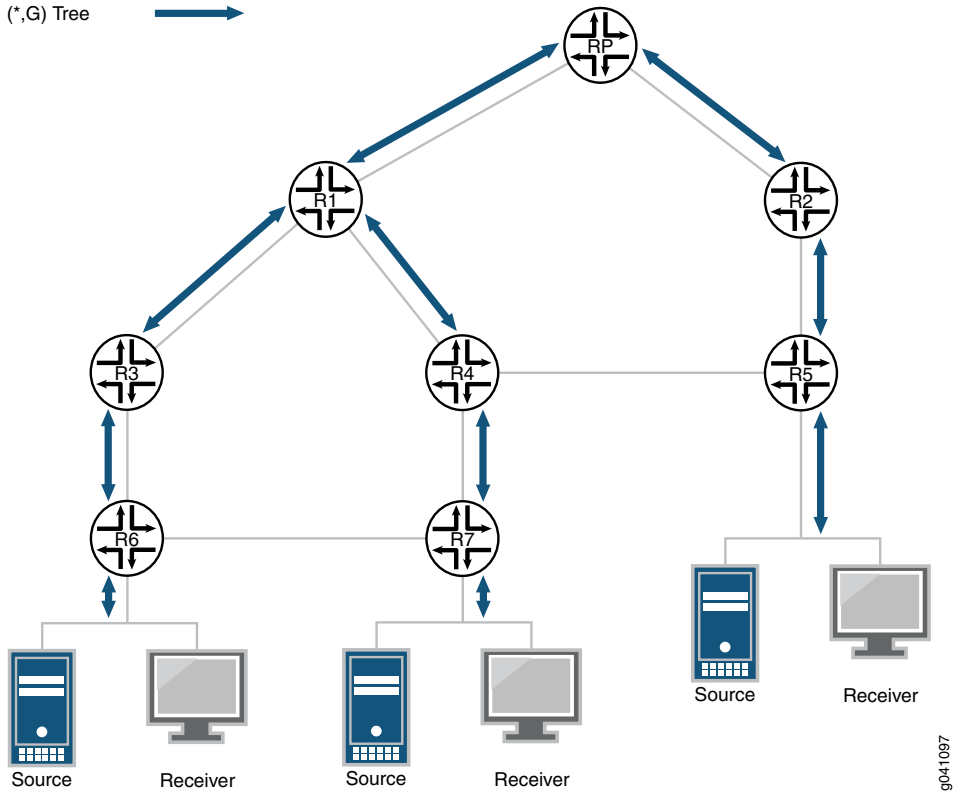
Figure 61: Example PIM Sparse-Mode Tree



Bidirectional PIM solves this problem by building only group-specific (*,G) state. Thus, only a single (*,G) route is needed for each group to deliver traffic to and from all the sources.

Figure 62 on page 339 shows the traffic flows generated to deliver traffic for one group to and from three stations in a bidirectional PIM network.

Figure 62: Example Bidirectional PIM Tree



Bidirectional PIM builds bidirectional shared trees that are rooted at a rendezvous point (RP) address. Bidirectional traffic does not switch to shortest path trees (SPTs) as in PIM-SM and is therefore optimized for routing state size instead of path length. Bidirectional PIM routes are always wildcard-source (*,G) routes. The protocol eliminates the need for (S,G) routes and data-triggered events. The bidirectional (*,G) group trees carry traffic both upstream from senders toward the RP, and downstream from the RP to receivers. As a consequence, the strict reverse path forwarding (RPF)-based rules found in other PIM modes do not apply to bidirectional PIM. Instead, bidirectional PIM routes forward traffic from all sources and the RP. Thus, bidirectional PIM routers must have the ability to accept traffic on many potential incoming interfaces.

Designated Forwarder Election

To prevent forwarding loops, only one router on each link or subnet (including point-to-point links) is a designated forwarder (DF). The responsibilities of the DF are to forward downstream traffic onto the link toward the receivers and to forward upstream traffic from the link toward the RP address. Bidirectional PIM relies on a process called DF election to choose the DF router for each interface and for each RP address. Each bidirectional PIM router in a subnet advertises its interior gateway protocol (IGP) unicast route to the RP address. The router with the best IGP unicast route to the RP address wins the DF election. Each router advertises its IGP route metrics in DF Offer, Winner, Backoff, and Pass messages.

Junos OS implements the DF election procedures as stated in RFC 5015, except that Junos OS checks RP unicast reachability before accepting incoming DF messages. DF messages for unreachable rendezvous points are ignored.

Bidirectional PIM Modes

In the Junos OS implementation, there are two modes for bidirectional PIM: bidirectional-sparse and bidirectional-sparse-dense. The differences between bidirectional-sparse and bidirectional-sparse-dense modes are the same as the differences between sparse mode and sparse-dense mode. Sparse-dense mode allows the interface to operate on a per-group basis in either sparse or dense mode. A group specified as "dense" is not mapped to an RP. Use bidirectional-sparse-dense mode when you have a mix of bidirectional groups, sparse groups, and dense groups in your network. One typical scenario for this is the use of auto-RP, which uses dense-mode flooding to bootstrap itself for sparse mode or bidirectional mode. In general, the dense groups could be for any flows that the network design requires to be flooded.

Each group-to-RP mapping is controlled by the RP **group-ranges** statement and the **ssm-groups** statement.

The choice of PIM mode is closely tied to controlling how groups are mapped to PIM modes, as follows:

- **bidirectional-sparse**—Use if all multicast groups are operating in bidirectional, sparse, or SSM mode.
- **bidirectional-sparse-dense**—Use if multicast groups, except those that are specified in the **dense-groups** statement, are operating in bidirectional, sparse, or SSM mode.

Bidirectional Rendezvous Points

You can configure group-range-to-RP mappings network-wide statically, or only on routers connected to the RP addresses and advertise them dynamically. Unlike rendezvous points for PIM-SM, which must de-encapsulate PIM Register messages and perform other specific protocol actions, bidirectional PIM rendezvous points implement no specific functionality. RP addresses are simply locations in the network to rendezvous toward. In fact, RP addresses need not be loopback interface addresses or even be addresses configured on any router, as long as they are covered by a subnet that is connected to a bidirectional PIM-capable router and advertised to the network.

Thus, for bidirectional PIM, there is no meaningful distinction between static and local RP addresses. Therefore, bidirectional PIM rendezvous points are configured at the **[edit protocols pim rp bidirectional]** hierarchy level, not under **static** or **local**.

The settings at the **[edit protocol pim rp bidirectional]** hierarchy level function like the settings at the **[edit protocols pim rp local]** hierarchy level, except that they create bidirectional PIM RP state instead of PIM-SM RP state.

Where only a single local RP can be configured, multiple bidirectional rendezvous points can be configured having group ranges that are the same, different, or overlapping. It is also permissible for a group range or RP address to be configured as bidirectional and either static or local for sparse-mode.

If a bidirectional PIM RP is configured without a group range, the default group range is 224/4 for IPv4. For IPv6, the default is ff00::/8. You can configure a bidirectional PIM RP group range to cover an SSM group range, but in that case the SSM or DM group range takes precedence over the bidirectional PIM RP configuration for those groups. In other words, because SSM always takes precedence, it is not permitted to have a bidirectional group range equal to or more specific than an SSM or DM group range.

PIM Bootstrap and Auto-RP Support

Group ranges for the specified RP address are flagged by PIM as bidirectional PIM group-to-RP mappings and, if configured, are advertised using PIM bootstrap or auto-RP. Dynamic advertisement of bidirectional PIM-flagged group-to-RP mappings using PIM bootstrap, and auto-RP is controlled as normal using the **bootstrap** and **auto-rp** statements.

Bidirectional PIM RP addresses configured at the **[edit protocols pim rp bidirectional address]** hierarchy level are advertised by auto-RP or PIM bootstrap if the following prerequisites are met:

- The routing instance must be configured to advertise candidate rendezvous points by way of auto-RP or PIM bootstrap, and an auto-RP mapping agent or bootstrap router, respectively, must be elected.
- The RP address must either be configured locally on an interface in the routing instance, or the RP address must belong to a subnet connected to an interface in the routing instance.

IGMP and MLD Support

Internet Group Management Protocol (IGMP) version 1, version 2, and version 3 are supported with bidirectional PIM. Multicast Listener Discovery (MLD) version 1 and version 2 are supported with bidirectional PIM. However, in all cases, only anysource multicast (ASM) state is supported for bidirectional PIM membership.

The following rules apply to bidirectional PIM:

- IGMP and MLD (*,G) membership reports trigger the PIM DF to originate bidirectional PIM (*,G) join messages.
- IGMP and MLD (S,G) membership reports do not trigger the PIM DF to originate bidirectional PIM (*,G) join messages.

Bidirectional PIM and Graceful Restart

Bidirectional PIM accepts packets for a bidirectional route on multiple interfaces. This means that some topologies might develop multicast routing loops if all PIM neighbors are not synchronized with regard to the identity of the designated forwarder (DF) on each link. If one router is forwarding without actively participating in DF elections, particularly after unicast routing changes, multicast routing loops might occur.

If graceful restart for PIM is enabled and bidirectional PIM is enabled, the default graceful restart behavior is to continue forwarding packets on bidirectional routes. If the gracefully

restarting router was serving as a DF for some interfaces to rendezvous points, the restarting router sends a DF Winner message with a metric of 0 on each of these RP interfaces. This ensures that a neighbor router does not become the DF due to unicast topology changes that might occur during the graceful restart period. Sending a DF Winner message with a metric of 0 prevents another PIM neighbor from assuming the DF role until after graceful restart completes. When graceful restart completes, the gracefully restarted router sends another DF Winner message with the actual converged unicast metric.

The no-bidirectional-mode statement at the **[edit protocols pim graceful-restart]** hierarchy level overrides the default behavior and disables forwarding for bidirectional PIM routes during graceful restart recovery, both in cases of simple routing protocol process (rpd) restart and graceful Routing Engine switchover. This configuration statement provides a very conservative alternative to the default graceful restart behavior for bidirectional PIM routes. The reason to discontinue forwarding of packets on bidirectional routes is that the continuation of forwarding might lead to short-duration multicast loops in rare double-failure circumstances.

Junos OS Enhancements to Bidirectional PIM

In addition to the functionality specified in RFC 5015, the following functions are included in the Junos OS implementation of bidirectional PIM:

- Source-only branches without PIM join state
- Support for both IPv4 and IPv6 domain and multicast addresses
- Nonstop routing (NSR) for bidirectional PIM routes
- Support for bidirectional PIM in logical systems
- Support for non-forwarding and virtual router instances

The following caveats are applicable for the bidirectional PIM configuration on the PTX5000:

- PTX5000 routers can be configured both as a bidirectional PIM rendezvous point and the source node.
- For PTX5000 routers, you can configure the auto-rp statement at the **[edit protocols pim rp]** or the **[edit routing-instances *routing-instance-name* protocols pim rp]** hierarchy level with the **mapping** option, but not the **announce** option.

Limitations of Bidirectional PIM

The Junos OS implementation of bidirectional PIM does not support the following functionality:

Starting with Release 12.2, Junos OS extends the nonstop active routing PIM support to draft-rosen MVPNs.

PTX5000 routers do not support nonstop active routing or in-service software upgrade (ISSU) in Junos OS Release 13.3.

Nonstop active routing PIM support for draft-rosen MVPNs enables nonstop active routing-enabled devices to preserve draft-rosen MPVN-related information—such as default and data MDT states—across switchovers.

- SNMP for bidirectional PIM.
- Graceful Routing Engine switchover is configurable with bidirectional PIM enabled, but bidirectional routes do not forward packets during the switchover.
- Multicast VPNs (Draft Rosen and NextGen).

The bidirectional PIM protocol does not support the following functionality:

- Embedded RP
- Anycast RP

See Also

- Example: Configuring Bidirectional PIM on page 343
- Configuring PIM Auto-RP on page 258 in the *Multicast Protocols Feature Guide*
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255 in the *Multicast Protocols Feature Guide*

Example: Configuring Bidirectional PIM

This example shows how to configure bidirectional PIM, as specified in RFC 5015, *Bidirectional Protocol Independent Multicast (BIDIR-PIM)*.

Requirements

This example uses the following hardware and software components:

- Eight Juniper Networks routers that can be M120, M320, MX Series, or T Series platforms. To support bidirectional PIM, M Series platforms must have I-chip FPCs. M7i, M10i, M40e, and other older M Series routers do not support bidirectional PIM.

- Junos OS Release 12.1 or later running on all eight routers.

Overview

Compared to PIM sparse mode, bidirectional PIM requires less PIM router state information. Because less state information is required, bidirectional PIM scales well and is useful in deployments with many dispersed sources and receivers.

In this example, two rendezvous points are configured statically. One RP is configured as a phantom RP. A phantom RP is an RP address that is a valid address on a subnet, but is not assigned to a PIM router interface. The subnet must be reachable by the bidirectional PIM routers in the network. For the other (non-phantom) RP in this example, the RP address is assigned to a PIM router interface. It can be assigned to either the loopback interface or any physical interface on the router. In this example, it is assigned to a physical interface.

OSPF is used as the interior gateway protocol (IGP) in this example. The OSPF metric determines the designated forwarder (DF) election process. In bidirectional PIM, the DF establishes a loop-free shortest-path tree that is rooted at the RP. On every network segment and point-to-point link, all PIM routers participate in DF election. The procedure selects one router as the DF for every RP of bidirectional groups. This router forwards multicast packets received on that network upstream to the RP. The DF election uses the same tie-break rules used by PIM assert processes.

This example uses the default DF election parameters. Optionally, at the **[edit protocols pim interface (*interface-name* | all)** bidirectional**]** hierarchy level, you can configure the following parameters related to the DF election:

- The robustness-count is the minimum number of DF election messages that must be lost for election to fail.
- The offer period is the interval to wait between repeated DF Offer and Winner messages.
- The backoff period is the period that the acting DF waits between receiving a better DF Offer and sending the Pass message to transfer DF responsibility.

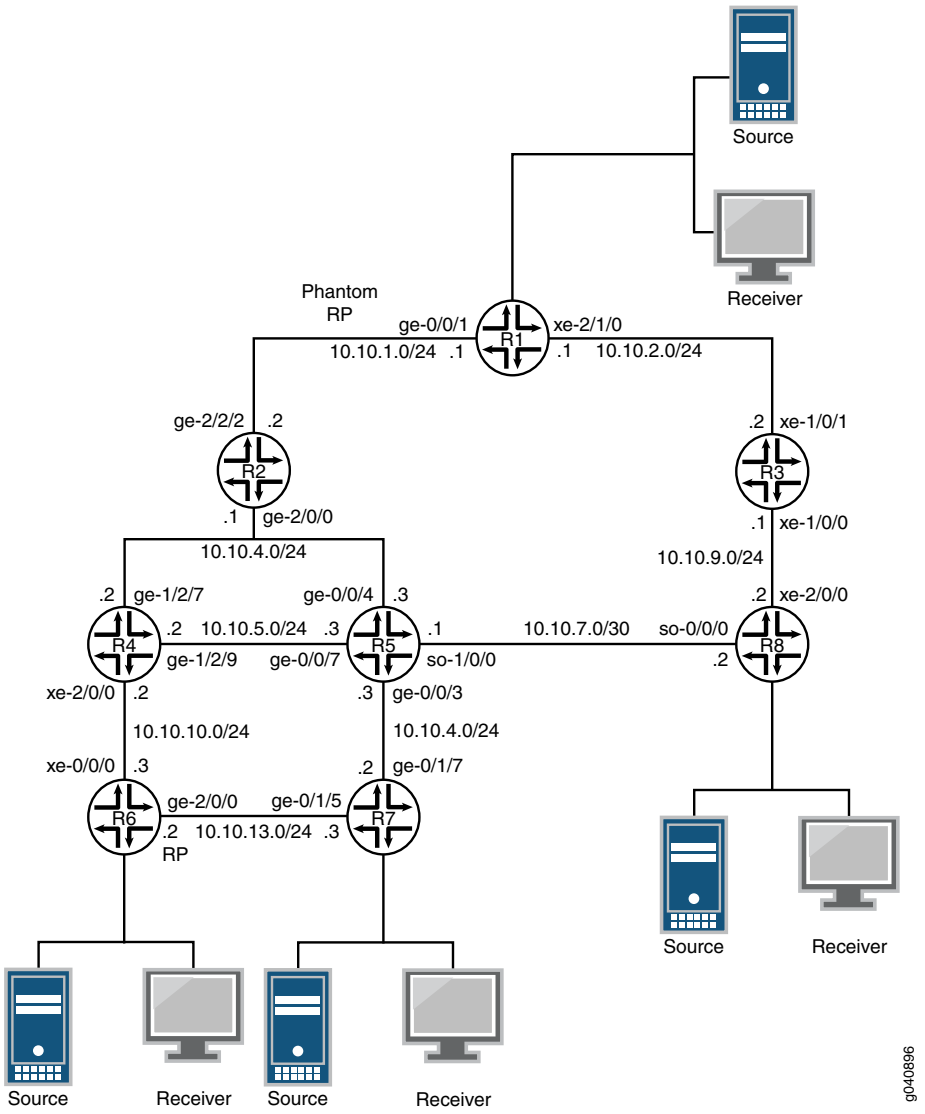
This example uses bidirectional-sparse-dense mode on the interfaces. The choice of PIM mode is closely tied to controlling how groups are mapped to PIM modes, as follows:

- **bidirectional-sparse**—Use if all multicast groups are operating in bidirectional, sparse, or SSM mode.
- **bidirectional-sparse-dense**—Use if multicast groups, except those that are specified in the **dense-groups** statement, are operating in bidirectional, sparse, or SSM mode.

Topology Diagram

Figure 63 on page 345 shows the topology used in this example.

Figure 63: Bidirectional PIM with Statically Configured Rendezvous Points



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

Router R1

```
set interfaces ge-0/0/1 unit 0 family inet address 10.10.1.1/24
set interfaces xe-2/1/0 unit 0 family inet address 10.10.2.1/24
set interfaces lo0 unit 0 family inet address 10.255.11.11/32
set protocols ospf area 0.0.0.0 interface ge-0/0/1.0
set protocols ospf area 0.0.0.0 interface xe-2/1/0.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

```
set protocols pim traceoptions file df
set protocols pim traceoptions flag bidirectional-df-election detail
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim interface ge-0/0/1.0 mode bidirectional-sparse-dense
set protocols pim interface xe-2/1/0.0 mode bidirectional-sparse-dense
```

Router R2

```
set interfaces ge-2/0/0 unit 0 family inet address 10.10.4.1/24
set interfaces ge-2/2/2 unit 0 family inet address 10.10.1.2/24
set interfaces lo0 unit 0 family inet address 10.255.22.22/32
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-2/2/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
set protocols pim traceoptions file df
set protocols pim traceoptions flag bidirectional-df-election detail
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim interface fxp0.0 disable
set protocols pim interface ge-2/0/0.0 mode bidirectional-sparse-dense
set protocols pim interface ge-2/2/2.0 mode bidirectional-sparse-dense
```

Router R3

```
set interfaces xe-1/0/0 unit 0 family inet address 10.10.9.1/24
set interfaces xe-1/0/1 unit 0 family inet address 10.10.2.2/24
set interfaces lo0 unit 0 family inet address 10.255.33.33/32
set protocols ospf area 0.0.0.0 interface xe-1/0/1.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface xe-1/0/0.0
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim interface xe-1/0/1.0 mode bidirectional-sparse-dense
set protocols pim interface xe-1/0/0.0 mode bidirectional-sparse-dense
```

Router R4

```
set interfaces ge-1/2/7 unit 0 family inet address 10.10.4.2/24
set interfaces ge-1/2/8 unit 0 family inet address 10.10.5.2/24
set interfaces xe-2/0/0 unit 0 family inet address 10.10.10.2/24
set interfaces lo0 unit 0 family inet address 10.255.44.44/32
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-1/2/7.0
set protocols ospf area 0.0.0.0 interface ge-1/2/8.0
set protocols ospf area 0.0.0.0 interface xe-2/0/0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols pim traceoptions file df
set protocols pim traceoptions flag bidirectional-df-election detail
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
```

```
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim interface xe-2/0/0.0 mode bidirectional-sparse-dense
set protocols pim interface ge-1/2/7.0 mode bidirectional-sparse-dense
set protocols pim interface ge-1/2/8.0 mode bidirectional-sparse-dense
```

Router R5

```
set interfaces ge-0/0/3 unit 0 family inet address 10.10.12.3/24
set interfaces ge-0/0/4 unit 0 family inet address 10.10.4.3/24
set interfaces ge-0/0/7 unit 0 family inet address 10.10.5.3/24
set interfaces so-1/0/0 unit 0 family inet address 10.10.7.1/30
set interfaces lo0 unit 0 family inet address 10.255.55.55/32
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/0/7.0
set protocols ospf area 0.0.0.0 interface ge-0/0/4.0
set protocols ospf area 0.0.0.0 interface so-1/0/0.0
set protocols ospf area 0.0.0.0 interface ge-0/0/3.0
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim interface ge-0/0/7.0 mode bidirectional-sparse-dense
set protocols pim interface ge-0/0/4.0 mode bidirectional-sparse-dense
set protocols pim interface so-1/0/0.0 mode bidirectional-sparse-dense
set protocols pim interface ge-0/0/3.0 mode bidirectional-sparse-dense
```

Router R6

```
set interfaces xe-0/0/0 unit 0 family inet address 10.10.10.3/24
set interfaces ge-2/0/0 unit 0 family inet address 10.10.13.2/24
set interfaces lo0 unit 0 family inet address 10.255.66.66/32
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface ge-2/0/0.0
set protocols ospf area 0.0.0.0 interface xe-0/0/0.0
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim interface fxp0.0 disable
set protocols pim interface xe-0/0/0.0 mode bidirectional-sparse-dense
set protocols pim interface ge-2/0/0.0 mode bidirectional-sparse-dense
```

Router R7

```
set interfaces ge-0/1/5 unit 0 family inet address 10.10.13.3/24
set interfaces ge-0/1/7 unit 0 family inet address 10.10.12.2/24
set interfaces lo0 unit 0 family inet address 10.255.77.77/32
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface ge-0/1/5.0
set protocols ospf area 0.0.0.0 interface ge-0/1/7.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim interface ge-0/1/5.0 mode bidirectional-sparse-dense
set protocols pim interface ge-0/1/7.0 mode bidirectional-sparse-dense
```

Router R8

```
set interfaces so-0/0/0 unit 0 family inet address 10.10.7.2/30
set interfaces xe-2/0/0 unit 0 family inet address 10.10.9.2/24
set interfaces lo0 unit 0 family inet address 10.255.88.88/32
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface xe-2/0/0.0
set protocols ospf area 0.0.0.0 interface so-0/0/0.0
set protocols pim traceoptions file df
set protocols pim traceoptions flag bidirectional-df-election detail
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 224.1.1.0/24
set protocols pim rp bidirectional address 10.10.13.2 group-ranges 225.1.1.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 224.1.3.0/24
set protocols pim rp bidirectional address 10.10.1.3 group-ranges 225.1.3.0/24
set protocols pim interface xe-2/0/0.0 mode bidirectional-sparse-dense
set protocols pim interface so-0/0/0.0 mode bidirectional-sparse-dense
```

Router R1

Step-by-Step Procedure

To configure Router R1:

1. Configure the router interfaces.

```
[edit interfaces]
user@R1# set ge-0/0/1 unit 0 family inet address 10.10.1.1/24
user@R1# set xe-2/1/0 unit 0 family inet address 10.10.2.1/24
user@R1# set lo0 unit 0 family inet address 10.255.11.11/32
```

2. Configure OSPF on the interfaces.

```
[edit protocols ospf area 0.0.0.0]
user@R1# set interface ge-0/0/1.0
user@R1# set interface xe-2/1/0.0
user@R1# set interface lo0.0
user@R1# set interface fxp0.0 disable
```

3. Configure the group-to-RP mappings.

```
[edit protocols pim rp bidirectional]
user@R1# set address 10.10.1.3 group-ranges 224.1.3.0/24
user@R1# set address 10.10.1.3 group-ranges 225.1.3.0/24
user@R1# set address 10.10.13.2 group-ranges 224.1.1.0/24
user@R1# set address 10.10.13.2 group-ranges 225.1.1.0/24
```

The RP represented by IP address 10.10.1.3 is a phantom RP. The 10.10.1.3 address is not assigned to any interface on any of the routers in the topology. It is, however, a reachable address. It is in the subnet between Routers R1 and R2.

The RP represented by address 10.10.13.2 is assigned to the **ge-2/0/0** interface on Router R6.

4. Enable bidirectional PIM on the interfaces.

```
[edit protocols pim]
user@R1# set interface ge-0/0/1.0 mode bidirectional-sparse-dense
user@R1# set interface xe-2/1/0.0 mode bidirectional-sparse-dense
```

5. (Optional) Configure tracing operations for the DF election process.

```
[edit protocols pim]
user@R1# set traceoptions file df
user@R1# set traceoptions flag bidirectional-df-election detail
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
ge-0/0/1 {
    unit 0 {
        family inet {
            address 10.10.1.1/24;
        }
    }
}
xe-2/1/0 {
    unit 0 {
        family inet {
            address 10.10.2.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.11.11/32;
        }
    }
}
```

```
user@R1# show protocols
ospf {
    area 0.0.0.0 {
        interface ge-0/0/1.0;
        interface xe-2/1/0.0;
        interface lo0.0;
        interface fxp0.0 {
            disable;
        }
    }
}
pim {
    rp {
        bidirectional {
            address 10.10.1.3 { # phantom RP
                group-ranges {
                    224.1.3.0/24;
                    225.1.3.0/24;
                }
            }
```

```
            address 10.10.13.2 {
              group-ranges {
                224.1.1.0/24;
                225.1.1.0/24;
              }
            }
          }
        }
        interface ge-0/0/1.0 {
          mode bidirectional-sparse-dense;
        }
        interface xe-2/1/0.0 {
          mode bidirectional-sparse-dense;
        }
        traceoptions {
          file df;
          flag bidirectional-df-election detail;
        }
    }
```

If you are done configuring the router, enter **commit** from configuration mode.

Repeat the procedure for every Juniper Networks router in the bidirectional PIM network, using the appropriate interface names and addresses for each router.

Verification

Confirm that the configuration is working properly.

Verifying Rendezvous Points

Purpose Verify the group-to-RP mapping information.

Action

```
user@R1> show pim rps
Instance: PIM.master
Address family INET
RP address      Type        Mode   Holdtime Timeout Groups Group prefixes
10.10.1.3       static      bidir       150    None      2 224.1.3.0/24
                                                           225.1.3.0/24
10.10.13.2      static      bidir       150    None      2 224.1.1.0/24
                                                           225.1.1.0/24
```

Verifying Messages

Purpose Check the number of DF election messages sent and received, and check bidirectional join and prune error statistics.

Action

```
user@R1> show pim statistics
PIM Message type        Received       Sent  Rx errors
V2 Hello                      16         34          0
...
V2 DF Election                18         38          0
...

Global Statistics

...
Rx Bidir Join/Prune on non-Bidir if          0
Rx Bidir Join/Prune on non-DF if             0
```

Checking the PIM Join State

Purpose Confirm the upstream interface, neighbor, and state information.

Action

```
user@R1> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.0
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.13.2
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0
    Upstream neighbor: 10.10.1.2
    Upstream state: None
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)

Group: 224.1.3.0
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.1.3
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0 (RP Link)
    Upstream neighbor: Direct
    Upstream state: Local RP
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)
        Interface: xe-2/1/0.0    (DF Winner)

Group: 225.1.1.0
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.13.2
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0
    Upstream neighbor: 10.10.1.2
    Upstream state: None
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)

Group: 225.1.3.0
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.1.3
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0 (RP Link)
    Upstream neighbor: Direct
    Upstream state: Local RP
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)
        Interface: xe-2/1/0.0    (DF Winner)
```

Meaning The output shows a (*,G-range) entry for each active bidirectional RP group range. These entries provide a hierarchy from which the individual (*,G) routes inherit RP-derived state (upstream information and accepting interfaces). These entries also provide the control

plane basis for the (*, G-range) forwarding routes that implement the sender-only branches of the tree.

Displaying the Designated Forwarder

Purpose Display RP address information and confirm the DF elected.

Action

```
user@R1> show pim bidirectional df-election
Instance: PIM.master Family: INET

RPA: 10.10.1.3
Group ranges: 224.1.3.0/24, 225.1.3.0/24
Interfaces:
    ge-0/0/1.0    (RPL)      DF: none
    lo0.0         (Win)      DF: 10.255.179.246
    xe-2/1/0.0    (Win)      DF: 10.10.2.1

RPA: 10.10.13.2
Group ranges: 224.1.1.0/24, 225.1.1.0/24
Interfaces:
    ge-0/0/1.0    (Lose)     DF: 10.10.1.2
    lo0.0         (Win)      DF: 10.255.179.246
    xe-2/1/0.0    (Lose)     DF: 10.10.2.2
```

Displaying the PIM Interfaces

Purpose Verify that the PIM interfaces have bidirectional-sparse-dense (**SDB**) mode assigned.

Action

```
user@R1> show pim interfaces
Instance: PIM.master

Stat = Status, V = Version, NbrCnt = Neighbor Count,
S = Sparse, D = Dense, B = Bidirectional,
DR = Designated Router, P2P = Point-to-point link,
Active = Bidirectional is active, NotCap = Not Bidirectional Capable

Name               Stat Mode IP V State        NbrCnt JoinCnt(sg/*g) DR address
ge-0/0/1.0          Up   SDB   4 2 NotDR,Active      1 0/0            10.10.1.2
lo0.0              Up   SDB   4 2 DR,Active        0 9901/100      10.255.179.246
xe-2/1/0.0          Up   SDB   4 2 NotDR,Active      1 0/0            10.10.2.2
```

Checking the PIM Neighbors

Purpose Check that the router detects that its neighbors are enabled for bidirectional PIM by verifying that the **B** option is displayed.

Action

```
user@R1> show pim neighbors
Instance: PIM.master
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking Bit

Interface           IP V Mode        Option       Uptime Neighbor addr
ge-0/0/1.0           4 2             HPLGBT     00:06:46 10.10.1.2
xe-2/1/0.0           4 2             HPLGBT     00:06:46 10.10.2.2
```

Checking the Route to the Rendezvous Points

Purpose Check the interface route to the rendezvous points.

Action

```
user@R1> show route 10.10.13.2
inet.0: 56 destinations, 56 routes (55 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.13.0/24      *[OSPF/10] 00:04:35, metric 4
                    > to 10.10.1.2 via ge-0/0/1.0
```

```
user@R1> show route 10.10.1.3
inet.0: 56 destinations, 56 routes (55 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

10.10.1.0/24       *[Direct/0] 00:06:25
                    > via ge-0/0/1.0
```

Verifying Multicast Routes

Purpose Verify the multicast traffic route for each group.

For bidirectional PIM, the **show multicast route extensive** command shows the (*, G*/prefix*) forwarding routes and the list of interfaces that accept bidirectional PIM traffic.

Action

```
user@R1> show multicast route extensive
Family: INET

Group: 224.0.0.0/4
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0 xe-4/1/0.0
    Downstream interface list:
        ge-0/0/1.0
    Session description: zeroconfaddr
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097157
    Incoming interface list ID: 559
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Group: 224.1.1.0/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0
    Downstream interface list:
        ge-0/0/1.0
    Session description: NOB Cross media facilities
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097157
    Incoming interface list ID: 579
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Group: 224.1.3.0/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0 xe-4/1/0.0
    Downstream interface list:
        ge-0/0/1.0
    Session description: NOB Cross media facilities
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097157
    Incoming interface list ID: 556
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Group: 225.1.1.0/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0
    Downstream interface list:
        ge-0/0/1.0
    Session description: Unknown
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097157
```

```
    Incoming interface list ID: 579
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Group: 225.1.3.0/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0 xe-4/1/0.0
    Downstream interface list:
        ge-0/0/1.0
    Session description: Unknown
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097157
    Incoming interface list ID: 556
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
```

Meaning For information about how the incoming and outgoing interface lists are derived, see the forwarding rules in RFC 5015.

Viewing Multicast Next Hops

Purpose Verify that the correct accepting interfaces are shown in the incoming interface list.

Action

```
user@R1> show multicast next-hops
Family: INET
ID          Refcount KRefcount Downstream interface
2097157           10         5 ge-0/0/1.0


Family: Incoming interface list
ID          Refcount KRefcount Downstream interface
579                5         2 lo0.0
                               ge-0/0/1.0
556                5         2 lo0.0
                               ge-0/0/1.0
                               xe-4/1/0.0
559                3         1 lo0.0
                               ge-0/0/1.0
                               xe-4/1/0.0
```

Meaning The nexthop IDs for the outgoing and incoming next hops are referenced directly in the **show multicast route extensive** command.

See Also

- Understanding Bidirectional PIM on page 337

CHAPTER 13

Rapidly Detecting Communication Failures with PIM and the BFD Protocol

Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol

Understanding Bidirectional Forwarding Detection Authentication for PIM

Bidirectional Forwarding Detection (BFD) enables rapid detection of communication failures between adjacent systems. By default, authentication for BFD sessions is disabled. However, when you run BFD over Network Layer protocols, the risk of service attacks can be significant. We strongly recommend using authentication if you are running BFD over multiple hops or through insecure tunnels.

Beginning with Junos OS Release 9.6, Junos OS supports authentication for BFD sessions running over PIM. BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

You authenticate BFD sessions by specifying an authentication algorithm and keychain, and then associating that configuration information with a security authentication keychain using the keychain name.

The following sections describe the supported authentication algorithms, security keychains, and level of authentication that can be configured:

BFD Authentication Algorithms

Junos OS supports the following algorithms for BFD authentication:

- **simple-password**—Plain-text password. One to 16 bytes of plain text are used to authenticate the BFD session. One or more passwords can be configured. This method is the least secure and should be used only when BFD sessions are not subject to packet interception.
- **keyed-md5**—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed MD5 uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than or equal to the last sequence number received. Although more secure than a simple password, this method is vulnerable to replay attacks. Increasing the rate at which the sequence number is updated can reduce this risk.
- **meticulous-keyed-md5**—Meticulous keyed Message Digest 5 hash algorithm. This method works in the same manner as keyed MD5, but the sequence number is updated with every packet. Although more secure than keyed MD5 and simple passwords, this method might take additional time to authenticate the session.
- **keyed-sha-1**—Keyed Secure Hash Algorithm I for sessions with transmit and receive intervals greater than 100 ms. To authenticate the BFD session, keyed SHA uses one or more secret keys (generated by the algorithm) and a sequence number that is updated periodically. The key is not carried within the packets. With this method, packets are accepted at the receiving end of the session if one of the keys matches and the sequence number is greater than the last sequence number received.
- **meticulous-keyed-sha-1**—Meticulous keyed Secure Hash Algorithm I. This method works in the same manner as keyed SHA, but the sequence number is updated with every packet. Although more secure than keyed SHA and simple passwords, this method might take additional time to authenticate the session.

NOTE: **Nonstop active routing (NSR) is not supported with meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.**

Security Authentication Keychains

The security authentication keychain defines the authentication attributes used for authentication key updates. When the security authentication keychain is configured and associated with a protocol through the keychain name, authentication key updates can occur without interrupting routing and signaling protocols.

The authentication keychain contains one or more keychains. Each keychain contains one or more keys. Each key holds the secret data and the time at which the key becomes valid. The algorithm and keychain must be configured on both ends of the BFD session,

and they must match. Any mismatch in configuration prevents the BFD session from being created.

BFD allows multiple clients per session, and each client can have its own keychain and algorithm defined. To avoid confusion, we recommend specifying only one security authentication keychain.

NOTE: Security Authentication Keychain is not supported on SRX Series devices.

Strict Versus Loose Authentication

By default, strict authentication is enabled, and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is configured, packets are accepted without authentication being checked at each end of the session. This feature is intended for transitional periods only.

See Also

- Configuring BFD Authentication for PIM on page 196
- Configuring BFD for PIM on page 194
- *authentication-key-chains*
- bfd-liveness-detection on page 984
- *show bfd session*

Configuring BFD for PIM

The Bidirectional Forwarding Detection (BFD) Protocol is a simple hello mechanism that detects failures in a network. BFD works with a wide variety of network environments and topologies. A pair of routing devices exchanges BFD packets. Hello packets are sent at a specified, regular interval. A neighbor failure is detected when the routing device stops receiving a reply after a specified interval. The BFD failure detection timers have shorter time limits than the Protocol Independent Multicast (PIM) hello hold time, so they provide faster detection.

The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

You must specify the minimum transmit and minimum receive intervals to enable BFD on PIM.

To enable failure detection:

1. Configure the interface globally or in a routing instance.

   This example shows the global configuration.

   ```
   [edit protocols pim]
   user@host# edit interface fe-1/0/0.0 family inet bfd-liveness-detection
   ```

2. Configure the minimum transmit interval.

   This is the minimum interval after which the routing device transmits hello packets to a neighbor with which it has established a BFD session. Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set transmit-interval 350
   ```

3. Configure the minimum interval after which the routing device expects to receive a reply from a neighbor with which it has established a BFD session.

   Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set minimum-receive-interval 350
   ```

4. (Optional) Configure other BFD settings.

   As an alternative to setting the receive and transmit intervals separately, configure one interval for both.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set minimum-interval 350
   ```

5. Configure the threshold for the adaptation of the BFD session detection time.

   When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set detection-time threshold 800
   ```

6. Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set multiplier 50
   ```

7. Configure the BFD version.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set version 1
   ```

8. Specify that BFD sessions should not adapt to changing network conditions.

   We recommend that you not disable BFD adaptation unless it is preferable not to have BFD adaptation enabled in your network.

   ```
   [edit protocols pim interface fe-1/0/0.0 family inet bfd-liveness-detection]
   user@host# set no-adaptation
   ```

9. Verify the configuration by checking the output of the **show bfd session** command.

See Also
- *show bfd session*

Configuring BFD Authentication for PIM

1. Specify the BFD authentication algorithm for the PIM protocol.
2. Associate the authentication keychain with the PIM protocol.
3. Configure the related security authentication keychain.

Beginning with Junos OS Release 9.6, you can configure authentication for Bidirectional Forwarding Detection (BFD) sessions running over Protocol Independent Multicast (PIM). Routing instances are also supported.

The following sections provide instructions for configuring and viewing BFD authentication on PIM:

- Configuring BFD Authentication Parameters on page 361
- Viewing Authentication Information for BFD Sessions on page 362

Configuring BFD Authentication Parameters

BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.

To configure BFD authentication:

1. Specify the algorithm (**keyed-md5**, **keyed-sha-1**, **meticulous-keyed-md5**, **meticulous-keyed-sha-1**, or **simple-password**) to use for BFD authentication on a PIM route or routing instance.

   ```
   [edit protocols pim]
   user@host# set interface ge-0/1/5 family inet bfd-liveness-detection authentication
     algorithm keyed-sha-1
   ```

NOTE: Nonstop active routing (NSR) is not supported with the meticulous-keyed-md5 and meticulous-keyed-sha-1 authentication algorithms. BFD sessions using these algorithms might go down after a switchover.

2. Specify the keychain to be used to associate BFD sessions on the specified PIM route or routing instance with the unique security authentication keychain attributes.

   The keychain you specify must match the keychain name configured at the **[edit security authentication key-chains]** hierarchy level.

   ```
   [edit protocols pim]
   user@host# set interface ge-0/1/5 family inet bfd-liveness-detection authentication
     keychain bfd-pim
   ```

   NOTE: The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

3. Specify the unique security authentication information for BFD sessions:

   - The matching keychain name as specified in Step 2.
   - At least one key, a unique integer between **0** and **63**. Creating multiple keys allows multiple clients to use the BFD session.
   - The secret data used to allow access to the session.
   - The time at which the authentication key becomes active, in the format *yyyy-mm-dd.hh:mm:ss*.

   ```
   [edit security]
   user@host# set authentication-key-chains key-chain bfd-pim key 53 secret $ABC123$/
     start-time 2009-06-14.10:00:00
   ```

   NOTE:
   Security Authentication Keychain is not supported on SRX Series devices.

4. (Optional) Specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

   ```
   [edit protocols pim]
   user@host# set interface ge-0/1/5 family inet bfd-liveness-detection authentication
     loose-check
   ```

5. (Optional) View your configuration by using the **show bfd session detail** or **show bfd session extensive** command.

6. Repeat these steps to configure the other end of the BFD session.

Viewing Authentication Information for BFD Sessions

You can view the existing BFD authentication configuration by using the **show bfd session detail** and **show bfd session extensive** commands.

The following example shows BFD authentication configured for the **ge-0/1/5** interface. It specifies the keyed SHA-1 authentication algorithm and a keychain name of **bfd-pim**. The authentication keychain is configured with two keys. Key **1** contains the secret data "**$ABC123/**" and a start time of June 1, 2009, at 9:46:02 AM PST. Key **2** contains the secret data "**$ABC123/**" and a start time of June 1, 2009, at 3:29:20 PM PST.

```
[edit protocols pim]
interface ge-0/1/5 {
  family inet {
    bfd-liveness-detection {
      authentication {
        key-chain bfd-pim;
        algorithm keyed-sha-1;
      }
    }
  }
}
[edit security]
authentication key-chains {
  key-chain bfd-pim {
    key 1 {
      secret "$ABC123/";
      start-time "2009-6-1.09:46:02 -0700";
    }
    key 2 {
      secret "$ABC123/";
      start-time "2009-6-1.15:29:20 -0700";
    }
  }
}
```

If you commit these updates to your configuration, you see output similar to the following example. In the output for the **show bfd session detail** command, **Authenticate** is displayed to indicate that BFD authentication is configured. For more information about the configuration, use the **show bfd session extensive** command. The output for this command provides the keychain name, the authentication algorithm and mode for each client in the session, and the overall BFD authentication configuration status, keychain name, and authentication algorithm and mode.

show bfd session detail

```
user@host# show bfd session detail

                                                Detect   Transmit
Address                 State     Interface     Time     Interval  Multiplier
192.0.2.2                Up        ge-0/1/5.0     0.900     0.300        3
 Client PIM, TX interval 0.300, RX interval 0.300, Authenticate
 Session up time 3d 00:34
 Local diagnostic None, remote diagnostic NbrSignal
 Remote state Up, version 1
 Replicated
```

show bfd session extensive

```
user@host# show bfd session extensive
                                                Detect   Transmit
```

```
Address                  State     Interface      Time      Interval  Multiplier
192.0.2.2                 Up        ge-0/1/5.0     0.900     0.300        3
 Client PIM, TX interval 0.300, RX interval 0.300, Authenticate
        keychain bfd-pim, algo keyed-sha-1, mode strict
  Session up time 00:04:42
 Local diagnostic None, remote diagnostic NbrSignal
 Remote state Up, version 1
 Replicated
 Min async interval 0.300, min slow interval 1.000
 Adaptive async TX interval 0.300, RX interval 0.300
 Local min TX interval 0.300, minimum RX interval 0.300, multiplier 3
 Remote min TX interval 0.300, min RX interval 0.300, multiplier 3
 Local discriminator 2, remote discriminator 2
 Echo mode disabled/inactive
 Authentication enabled/active, keychain bfd-pim, algo keyed-sha-1, mode strict
```

See Also

- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 357
- Configuring BFD for PIM on page 194
- *authentication-key-chains*
- bfd-liveness-detection on page 984
- *show bfd session*

Example: Configuring BFD Liveness Detection for PIM IPv6

This example shows how to configure Bidirectional Forwarding Detection (BFD) liveness detection for IPv6 interfaces configured for the Protocol Independent Multicast (PIM) topology. BFD is a simple hello mechanism that detects failures in a network.

The following steps are needed to configure BFD liveness detection:

1. Configure the interface.
2. Configure the related security authentication keychain.
3. Specify the BFD authentication algorithm for the PIM protocol.
4. Configure PIM, associating the authentication keychain with the desired protocol.
5. Configure BFD authentication for the routing instance.

NOTE: **You must perform these steps on both ends of the BFD session.**

- Requirements on page 365
- Overview on page 365
- Configuration on page 366
- Verification on page 369

Requirements

This example uses the following hardware and software components:

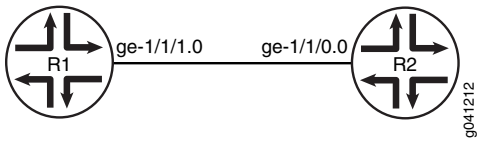
- Two peer routers.
- Junos OS 12.2 or later.

Overview

In this example. Device R1 and Device R2 are peers. Each router runs PIM, connected over a common medium.

Figure 64 on page 365 shows the topology used in this example.

Figure 64: BFD Liveness Detection for PIM IPv6 Topology



Assume that the routers initialize. No BFD session is yet established. For each router, PIM informs the BFD process to monitor the IPv6 address of the neighbor that is configured in the routing protocol. Addresses are not learned dynamically and must be configured.

Configure the IPv6 address and BFD liveness detection at the [**edit protocols pim**] hierarchy level for each router.

```
[edit protocols pim]
user@host# set interface interface-name family inet6 bfd-liveness-detection
```

Configure BFD liveness detection for the routing instance at the [**edit routing-instances*instance-name* protocols pim interface all family inet6**] hierarchy level (here, the *instance-name* is **instance1**:

```
[edit routing-instances instance1 protocols pim]
user@host# set bfd-liveness-detection
```

You will also configure the authentication algorithm and authentication keychain values for BFD.

In a BFD-configured network, when a client launches a BFD session with a peer, BFD begins sending slow, periodic BFD control packets that contain the interval values that you specified when you configured the BFD peers. This is known as the initialization state. BFD does not generate any up or down notifications in this state. When another BFD interface acknowledges the BFD control packets, the session moves into an up state and begins to more rapidly send periodic control packets. If a data path failure occurs and BFD does not receive a control packet within the configured amount of time, the data path is declared down and BFD notifies the BFD client. The BFD client can then perform the necessary actions to reroute traffic. This process can be different for different BFD clients.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces ge-0/1/5 unit 0 description toRouter2
set interfaces ge-0/1/5 unit 0 family inet6
set interfaces ge-0/1/5 unit 0 family inet6 address e80::21b:c0ff:fed5:e4dd
set protocols pim interface ge-0/1/5 family inet6 bfd-liveness-detection authentication
  algorithm keyed-sha-1
set protocols pim interface ge-0/1/5 family inet6 bfd-liveness-detection authentication
  key-chain bfd-pim
set routing-instances instance1 protocols pim interface all family inet6
  bfd-liveness-detection authentication algorithm keyed-sha-1
set routing-instances instance1 protocols pim interface all family inet6
  bfd-liveness-detection authentication key-chain bfd-pim
set security authentication key-chain bfd-pim key 1 secret "v"
set security authentication key-chain bfd-pim key 1 start-time "2012-01-01.09:46:02
  -0700"
set security authentication key-chain bfd-pim key 2 secret "$ABC123abc123"
set security authentication key-chain bfd-pim key 2 start-time "2012-01-01.15:29:20
  -0700"
```

Device R2

```
set interfaces ge-1/1/0 unit 0 description toRouter1
set interfaces ge-1/1/0 unit 0 family inet6 address e80::21b:c0ff:fed5:e5dd
set protocols pim interface ge-1/1/0 family inet6 bfd-liveness-detection authentication
  algorithm keyed-sha-1
set protocols pim interface ge-1/1/0 family inet6 bfd-liveness-detection authentication
  key-chain bfd-pim
set routing-instances instance1 protocols pim interface all family inet6
  bfd-liveness-detection authentication algorithm keyed-sha-1
set routing-instances instance1 protocols pim interface all family inet6
  bfd-liveness-detection authentication key-chain bfd-pim
set security authentication key-chain bfd-pim key 1 secret "$ABC123abc123"
set security authentication key-chain bfd-pim key 1 start-time "2012-01-01.09:46:02
  -0700"
set security authentication key-chain bfd-pim key 2 secret "$ABC123abc123"
set security authentication key-chain bfd-pim key 2 start-time "2012-01-01.15:29:20
  -0700"
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure BFD liveness detection for PIM IPv6 interfaces on Device R1:

NOTE: This procedure is for Device R1. Repeat this procedure for Device R2, after modifying the appropriate interface names, addresses, and any other parameters.

1. Configure the interface, using the **inet6** statement to specify that this is an IPv6 address.

```
[edit interfaces]
user@R1# set ge-0/1/5 unit 0 description toRouter2
user@R1# set ge-0/1/5 unit 0 family inet6 address e80::21b:c0ff:fed5:e4dd
```

2. Specify the BFD authentication algorithm and keychain for the PIM protocol.

   The keychain is used to associate BFD sessions on the specified PIM route or routing instance with the unique security authentication keychain attributes. This keychain name should match the keychain name configured at the **[edit security authentication]** hierarchy level.

```
[edit protocols]
user@R1# set pim interface ge-0/1/5.0 family inet6 bfd-liveness-detection
  authentication algorithm keyed-sha-1
user@R1# set pim interface ge-0/1/5 family inet6 bfd-liveness-detection
  authentication key-chain bfd-pim
```

NOTE: The algorithm and keychain must be configured on both ends of the BFD session, and they must match. Any mismatch in configuration prevents the BFD session from being created.

3. Configure a routing instance (here, **instance1**), specifying BFD authentication and associating the security authentication algorithm and keychain.

```
[edit routing-instances]
user@R1# set instance1 protocols pim interface all family inet6
  bfd-liveness-detection authentication algorithm keyed-sha-1
user@R1# set instance1 protocols pim interface all family inet6
  bfd-liveness-detection authentication key-chain bfd-pim
```

4. Specify the unique security authentication information for BFD sessions:

- The matching keychain name as specified in Step 2.
- At least one key, a unique integer between **0** and **63**. Creating multiple keys allows multiple clients to use the BFD session.
- The secret data used to allow access to the session.
- The time at which the authentication key becomes active, in the format **YYYY-MM-DD.hh:mm:ss**.

```
[edit security authentication]
user@R1# set key-chain bfd-pim key 1 secret "$ABC123abc123"
user@R1# set key-chain bfd-pim key 1 start-time "2012-01-01.09:46:02 -0700"
user@R1# set key-chain bfd-pim key 2 secret "$ABC123abc123"
user@R1# set key-chain bfd-pim key 2 start-time "2012-01-01.15:29:20 -0700"
```

Results

Confirm your configuration by issuing the **show interfaces**, **show protocols**, **show routing-instances**, and **show security** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R1# show interfaces
ge-0/1/5 {
    unit 0 {
        description toRouter2;
        family inet6 {
            address e80::21b:c0ff:fed5:e4dd {
            }
        }
    }
}

user@R1# show protocols
pim {
    interface ge-0/1/5.0 {
        family inet6;
            bfd-liveness-detection {
                authentication {
                    algorithm keyed-sha-1;
                    key-chain bfd-pim;
                }
            }
    }
}

user@R1# show routing-instances
instance1 {
    protocols {
        pim {
            interface all {
                family inet6 {
                    bfd-liveness-detection {
                        authentication {
                            algorithm keyed-sha-1;
```

```
                key-chain bfd-pim;
            }
        }
    }
  }
 }
}
```

```
user@R1# show security
authentication {
   key-chain bfd-pim {
      key 1 {
         secret “$ABC123abc123”;
         start-time “2012-01-01.09:46:02 -0700”;
      }
      key 2 {
         secret “$ABC123abc123”;
         start-time “2012-01-01.15:29:20 -0700”;
      }
   }
}
```

Verification

Confirm that the configuration is working properly.

Verifying the BFD Session

Purpose Verify that BFD liveness detection is enabled.

Action

```
user@R1# run show pim neighbors detail

Instance: PIM.master
    Interface: ge-0/1/5.0

       Address: fe80::21b:c0ff:fed5:e4dd,  IPv6, PIM v2, Mode: Sparse, sg Join
Count: 0, tsg Join Count: 0
            Hello Option Holdtime: 65535 seconds
            Hello Option DR Priority: 1
            Hello Option Generation ID: 1417610277
            Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                          Join Suppression supported

       Address: fe80::21b:c0ff:fedc:28dd,  IPv6, PIM v2, sg Join Count: 0, tsg
Join Count: 0
            Secondary address: beef::2
            BFD: Enabled, Operational state: Up
            Hello Option Holdtime: 105 seconds 80 remaining
            Hello Option DR Priority: 1
            Hello Option Generation ID: 1648636754
            Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                          Join Suppression supported
```

Meaning

The display from the **show pim neighbors detail** command shows **BFD: Enabled, Operational state: Up**, indicating that BFD is operating between the two PIM neighbors. For additional information about the BFD session (including the session ID number), use the **show bfd session extensive** command.

See Also

- *authentication-key-chains*
- bfd-liveness-detection (Protocols PIM) on page 984
- *show bfd session*

Release History Table

Release	Description
9.6	Beginning with Junos OS Release 9.6, Junos OS supports authentication for BFD sessions running over PIM. BFD authentication is only supported in the Canada and United States version of the Junos OS image and is not available in the export version.
9.6	Beginning with Junos OS Release 9.6, you can configure authentication for Bidirectional Forwarding Detection (BFD) sessions running over Protocol Independent Multicast (PIM). Routing instances are also supported.

Related Documentation

- *Configuring Basic PIM Settings*
- *Example: Configuring BFD for BGP*
- *Example: Configuring BFD Authentication for BGP*

CHAPTER 14

Configuring PIM Options

Example: Configuring Nonstop Active Routing for PIM

Understanding Nonstop Active Routing for PIM

Nonstop active routing configurations include two Routing Engines that share information so that routing is not interrupted during Routing Engine failover. When nonstop active routing is configured on a dual Routing Engine platform, the PIM control state is replicated on both Routing Engines.

This PIM state information includes:

- Neighbor relationships
- Join and prune information
- RP-set information
- Synchronization between routes and next hops and the forwarding state between the two Routing Engines

The PIM control state is maintained on the backup Routing Engine by the replication of state information from the master to the backup Routing Engine and having the backup Routing Engine react to route installation and modification in the **[instance].inet.1** routing table on the master Routing Engine. The backup Routing Engine does not send or receive PIM protocol packets directly. In addition, the backup Routing Engine uses the dynamic interfaces created by the master Routing Engine. These dynamic interfaces include PIM encapsulation, de-encapsulation, and multicast tunnel interfaces.

NOTE: **The clear pim join, clear pim register, and clear pim statistics operational mode commands are not supported on the backup Routing Engine when nonstop active routing is enabled.**

To enable nonstop active routing for PIM (in addition to the PIM configuration on the master Routing Engine), you must include the following statements at the **[edit]** hierarchy level:

- **chassis redundancy graceful-switchover**
- **routing-options nonstop-routing**
- **system commit synchronize**

See Also
- IGMP and Nonstop Active Routing on page 49

Example: Configuring Nonstop Active Routing with PIM

This example shows how to configure nonstop active routing for PIM-based multicast IPv4 and IPv6 traffic.

Requirements

For nonstop active routing for PIM-based multicast traffic to work with IPv6, the routing device must be running Junos OS Release 10.4 or above.

Before you begin:

- Configure the router interfaces. See the *Network Interfaces Configuration Guide*.
- Configure an interior gateway protocol or static routing. See the *Routing Protocols Configuration Guide*.
- Configure a multicast group membership protocol (IGMP or MLD). See "Understanding IGMP" on page 25 and "Understanding MLD" on page 51.

Overview

Junos OS supports nonstop active routing in the following PIM scenarios:

- Dense mode
- Sparse mode
- SSM
- Static RP
- Auto-RP (for IPv4 only)
- Bootstrap router
- Embedded RP on the non-RP router (for IPv6 only)

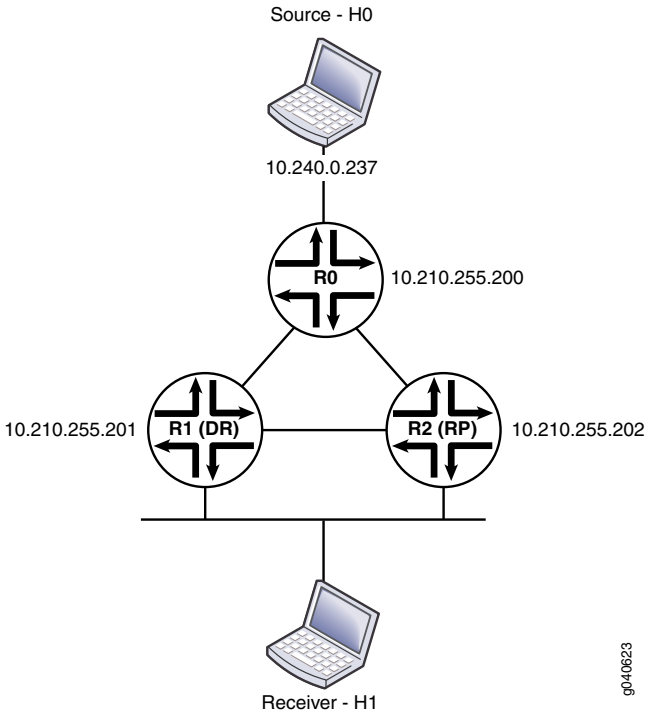
- BFD support
- Draft Rosen Multicast VPNs and BGP Multicast VPNs (use the **advertise-from-main-vpn-tables** option at the **[edit protocols bgp]** hierarchy level, to synchronize MVPN routes, cmcast, provider-tunnel and forwarding information between the master and the backup Routing Engines).
- Policy features such as neighbor policy, bootstrap router export and import policies, scope policy, flow maps, and reverse path forwarding (RPF) check policies

In Junos OS release 13.3, multicast VPNs are not supported with nonstop active routing. Policy-based features (such as neighbor policy, join policy, BSR policy, scope policy, flow maps, and RPF check policy) are not supported with nonstop active routing.

This example uses static RP. The interfaces are configured to receive both IPv4 and IPv6 traffic. R2 provides RP services as the local RP. Note that nonstop active routing is not supported on the RP router. The configuration shown in this example is on R1.

Figure 65 on page 373 shows the topology used in this example.

Figure 65: Nonstop Active Routing in PIM Domain



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

R1
```
set system syslog archive size 10m
set system syslog file messages any info
set system commit synchronize
set chassis redundancy graceful-switchover
set interfaces traceoptions file dcd-trace
set interfaces traceoptions file size 10m
set interfaces traceoptions file files 10
set interfaces traceoptions flag all
set interfaces so-0/0/1 unit 0 description "to R0 so-0/0/1.0"
set interfaces so-0/0/1 unit 0 family inet address 10.210.1.2/30
set interfaces so-0/0/1 unit 0 family inet6 address FDCA:9E34:50CE:0001::2/126
set interfaces fe-0/1/3 unit 0 description "to R2 fe-0/1/3.0"
set interfaces fe-0/1/3 unit 0 family inet address 10.210.12.1/30
set interfaces fe-0/1/3 unit 0 family inet6 address FDCA:9E34:50CE:0012::1/126
set interfaces fe-1/1/0 unit 0 description "to H1"
set interfaces fe-1/1/0 unit 0 family inet address 10.240.0.250/30
set interfaces fe-1/1/0 unit 0 family inet6 address ::10.240.0.250/126
set interfaces lo0 unit 0 description "R1 Loopback"
set interfaces lo0 unit 0 family inet address 10.210.255.201/32 primary
set interfaces lo0 unit 0 family iso address
  47.0005.80ff.f800.0000.0108.0001.0102.1025.5201.00
set interfaces lo0 unit 0 family inet6 address abcd::10:210:255:201/128
set protocols ospf traceoptions file r1-nsr-ospf2
set protocols ospf traceoptions file size 10m
set protocols ospf traceoptions file files 10
set protocols ospf traceoptions file world-readable
set protocols ospf traceoptions flag error
set protocols ospf traceoptions flag lsa-update detail
set protocols ospf traceoptions flag flooding detail
set protocols ospf traceoptions flag lsa-request detail
set protocols ospf traceoptions flag state detail
set protocols ospf traceoptions flag event detail
set protocols ospf traceoptions flag hello detail
set protocols ospf traceoptions flag nsr-synchronization detail
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface so-0/0/1.0 metric 100
set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface fe-1/1/0.0 passive
set protocols ospf3 traceoptions file r1-nsr-ospf3
set protocols ospf3 traceoptions file size 10m
set protocols ospf3 traceoptions file world-readable
set protocols ospf3 traceoptions flag lsa-update detail
set protocols ospf3 traceoptions flag flooding detail
set protocols ospf3 traceoptions flag lsa-request detail
set protocols ospf3 traceoptions flag state detail
set protocols ospf3 traceoptions flag event detail
set protocols ospf3 traceoptions flag hello detail
set protocols ospf3 traceoptions flag nsr-synchronization detail
set protocols ospf3 area 0.0.0.0 interface fe-1/1/0.0 passive
set protocols ospf3 area 0.0.0.0 interface fe-1/1/0.0 metric 1
set protocols ospf3 area 0.0.0.0 interface lo0.0 passive
set protocols ospf3 area 0.0.0.0 interface so-0/0/1.0 metric 1
set protocols ospf3 area 0.0.0.0 interface fe-0/1/3.0 metric 1
set protocols pim traceoptions file r1-nsr-pim
```

```
set protocols pim traceoptions file size 10m
set protocols pim traceoptions file files 10
set protocols pim traceoptions file world-readable
set protocols pim traceoptions flag mdt detail
set protocols pim traceoptions flag rp detail
set protocols pim traceoptions flag register detail
set protocols pim traceoptions flag packets detail
set protocols pim traceoptions flag autorp detail
set protocols pim traceoptions flag join detail
set protocols pim traceoptions flag hello detail
set protocols pim traceoptions flag assert detail
set protocols pim traceoptions flag normal detail
set protocols pim traceoptions flag state detail
set protocols pim traceoptions flag nsr-synchronization
set protocols pim rp static address 10.210.255.202
set protocols pim rp static address abcd::10:210:255:202
set protocols pim interface lo0.0
set protocols pim interface fe-0/1/3.0 mode sparse
set protocols pim interface fe-0/1/3.0 version 2
set protocols pim interface so-0/0/1.0 mode sparse
set protocols pim interface so-0/0/1.0 version 2
set protocols pim interface fe-1/1/0.0 mode sparse
set protocols pim interface fe-1/1/0.0 version 2
set policy-options policy-statement load-balance then load-balance per-packet
set routing-options nonstop-routing
set routing-options router-id 10.210.255.201
set routing-options forwarding-table export load-balance
set routing-options forwarding-table traceoptions file r1-nsr-krt
set routing-options forwarding-table traceoptions file size 10m
set routing-options forwarding-table traceoptions file world-readable
set routing-options forwarding-table traceoptions flag queue
set routing-options forwarding-table traceoptions flag route
set routing-options forwarding-table traceoptions flag routes
set routing-options forwarding-table traceoptions flag synchronous
set routing-options forwarding-table traceoptions flag state
set routing-options forwarding-table traceoptions flag asynchronous
set routing-options forwarding-table traceoptions flag consistency-checking
set routing-options traceoptions file r1-nsr-sync
set routing-options traceoptions file size 10m
set routing-options traceoptions flag nsr-synchronization
set routing-options traceoptions flag commit-synchronize
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure nonstop active routing on R1:

1. Synchronize the Routing Engines.

```
[edit]
user@host# edit system
[edit system]
user@host# set commit synchronize
user@host# exit
```

2. Enable graceful Routing Engine switchover.

```
[edit]
user@host# set chassis redundancy graceful-switchover
```

3. Configure R1's interfaces.

```
[edit]
user@host# edit interfaces
[edit interfaces]
user@host# set so-0/0/1 unit 0 description "to R0 so-0/0/1.0"
user@host# set so-0/0/1 unit 0 family inet address 10.210.1.2/30
user@host# set so-0/0/1 unit 0 family inet6 address FDCA:9E34:50CE:0001::2/126
user@host# set fe-0/1/3 unit 0 description "to R2 fe-0/1/3.0"
user@host# set fe-0/1/3 unit 0 family inet address 10.210.12.1/30
user@host# set fe-0/1/3 unit 0 family inet6 address FDCA:9E34:50CE:0012::1/126
user@host# set fe-1/1/0 unit 0 description "to H1"
user@host# set fe-1/1/0 unit 0 family inet address 10.240.0.250/30
user@host# set fe-1/1/0 unit 0 family inet6 address ::10.240.0.250/126
user@host# set lo0 unit 0 description "R1 Loopback"
user@host# set lo0 unit 0 family inet address 10.210.255.201/32 primary
user@host# set lo0 unit 0 family iso address
  47.0005.80ff.f800.0000.0108.0001.0102.1025.5201.00
user@host# set lo0 unit 0 family inet6 address abcd::10:210:255:201/128
user@host# exit
```

4. Configure OSPF for IPv4 on R1.

```
[edit]
user@host# edit protocols ospf
[edit protocols ospf]
user@host# set traffic-engineering
user@host# set area 0.0.0.0 interface so-0/0/1.0 metric 100
user@host# set area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@host# set area 0.0.0.0 interface lo0.0 passive
user@host# set area 0.0.0.0 interface fxp0.0 disable
user@host# set area 0.0.0.0 interface fe-1/1/0.0 passive
```

5. Configure OSPF for IPv6 on R1.

```
[edit]
user@host# edit protocols ospf3
[edit protocols ospf3]
user@host# set area 0.0.0.0 interface fe-1/1/0.0 passive
user@host# set area 0.0.0.0 interface fe-1/1/0.0 metric 1
user@host# set area 0.0.0.0 interface lo0.0 passive
user@host# set area 0.0.0.0 interface so-0/0/1.0 metric 1
user@host# set area 0.0.0.0 interface fe-0/1/3.0 metric 1
```

6. Configure PIM on R1. The PIM static address points to the RP router (R2).

```
[edit]
user@host# edit
[edit protocols pim]
```

```
user@host# set protocols pim rpstatic address 10.210.255.202
user@host# set protocols pim rp static address abcd::10:210:255:202
user@host# set protocols pim interface lo0.0
user@host# set protocols pim interface fe-0/1/3.0 mode sparse
user@host# set protocols pim interface fe-0/1/3.0 version 2
user@host# set protocols pim interface so-0/0/1.0 mode sparse
user@host# set protocols pim interface so-0/0/1.0 version 2
user@host# set protocols pim interface fe-1/1/0.0 mode sparse
user@host# set protocols pim interface fe-1/1/0.0 version 2
```

7. Configure per-packet load balancing on R1.

```
[edit]
user@host# edit policy-options policy-statement load-balance
[edit policy-options policy-statement load-balance]
user@host# set then load-balance per-packet
```

8. Apply the load-balance policy on R1.

```
[edit]
user@host# set routing-options forwarding-table export load-balance
```

9. Configure nonstop routing on R1.

```
[edit]
user@host# set routing-options nonstop-routing
user@host# set routing-options router-id 10.210.255.201
```

Step-by-Step Procedure

For troubleshooting, configure system log and tracing operations.

1. Enable system log messages.

```
[edit]
user@host# set system syslog archive size 10m
user@host# set system syslog file messages any info
```

2. Trace interface operations.

```
[edit]
user@host# set interfaces traceoptions file dcd-trace
user@host# set interfaces traceoptions file size 10m
user@host# set interfaces traceoptions file files 10
user@host# set interfaces traceoptions flag all
```

3. Trace IGP operations for IPv4.

```
[edit]
user@host# set protocols ospf traceoptions file r1-nsr-ospf2
user@host# set protocols ospf traceoptions file size 10m
user@host# set protocols ospf traceoptions file files 10
user@host# set protocols ospf traceoptions file world-readable
user@host# set protocols ospf traceoptions flag error
user@host# set protocols ospf traceoptions flag lsa-update detail
```

```
user@host# set protocols ospf traceoptions flag flooding detail
user@host# set protocols ospf traceoptions flag lsa-request detail
user@host# set protocols ospf traceoptions flag state detail
user@host# set protocols ospf traceoptions flag event detail
user@host# set protocols ospf traceoptions flag hello detail
user@host# set protocols ospf traceoptions flag nsr-synchronization detail
```

4. Trace IGP operations for IPv6.

```
[edit]
user@host# set protocols ospf3 traceoptions file r1-nsr-ospf3
user@host# set protocols ospf3 traceoptions file size 10m
user@host# set protocols ospf3 traceoptions file world-readable
user@host# set protocols ospf3 traceoptions flag lsa-update detail
user@host# set protocols ospf3 traceoptions flag flooding detail
user@host# set protocols ospf3 traceoptions flag lsa-request detail
user@host# set protocols ospf3 traceoptions flag state detail
user@host# set protocols ospf3 traceoptions flag event detail
user@host# set protocols ospf3 traceoptions flag hello detail
user@host# set protocols ospf3 traceoptions flag nsr-synchronization detail
```

5. Trace PIM operations.

```
[edit]
user@host# set protocols pim traceoptions file r1-nsr-pim
user@host# set protocols pim traceoptions file size 10m
user@host# set protocols pim traceoptions file files 10
user@host# set protocols pim traceoptions file world-readable
user@host# set protocols pim traceoptions flag mdt detail
user@host# set protocols pim traceoptions flag rp detail
user@host# set protocols pim traceoptions flag register detail
user@host# set protocols pim traceoptions flag packets detail
user@host# set protocols pim traceoptions flag autorp detail
user@host# set protocols pim traceoptions flag join detail
user@host# set protocols pim traceoptions flag hello detail
user@host# set protocols pim traceoptions flag assert detail
user@host# set protocols pim traceoptions flag normal detail
user@host# set protocols pim traceoptions flag state detail
user@host# set protocols pim traceoptions flag nsr-synchronization
```

6. Trace all routing protocol functionality.

```
[edit]
user@host# set routing-options traceoptions file r1-nsr-sync
user@host# set routing-options traceoptions file size 10m
user@host# set routing-options traceoptions flag nsr-synchronization
user@host# set routing-options traceoptions flag commit-synchronize
```

7. Trace forwarding table operations.

```
[edit]
user@host# set routing-options forwarding-table traceoptions file r1-nsr-krt
user@host# set routing-options forwarding-table traceoptions file size 10m
user@host# set routing-options forwarding-table traceoptions file world-readable
user@host# set routing-options forwarding-table traceoptions flag queue
```

```
user@host# set routing-options forwarding-table traceoptions flag route
user@host# set routing-options forwarding-table traceoptions flag routes
user@host# set routing-options forwarding-table traceoptions flag synchronous
user@host# set routing-options forwarding-table traceoptions flag state
user@host# set routing-options forwarding-table traceoptions flag asynchronous
user@host# set routing-options forwarding-table traceoptions flag
  consistency-checking
```

8. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show chassis**, **show interfaces**, **show policy-options**, **show protocols**, **show routing-options**, and **show system** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show chassis
redundancy {
  graceful-switchover;
}
user@host# show interfaces
traceoptions {
  file dcd-trace size 10m files 10;
  flag all;
}
so-0/0/1 {
  unit 0 {
    description "to R0 so-0/0/1.0";
    family inet {
      address 10.210.1.2/30;
    }
    family inet6 {
      address FDCA:9E34:50CE:0001::2/126;
    }
  }
}
fe-0/1/3 {
  unit 0 {
    description "to R2 fe-0/1/3.0";
    family inet {
      address 10.210.12.1/30;
    }
    family inet6 {
      address FDCA:9E34:50CE:0012::1/126;
    }
  }
}
fe-1/1/0 {
  unit 0 {
```

```
      description "to H1";
      family inet {
        address 10.240.0.250/30;
      }
      family inet6 {
        address ::10.240.0.250/126;
      }
    }
  }
  lo0 {
    unit 0 {
      description "R1 Loopback";
      family inet {
        address 10.210.255.201/32 {
          primary;
        }
      }
      family iso {
        address 47.0005.80ff.f800.0000.0108.0001.0102.1025.5201.00;
      }
      family inet6 {
        address abcd::10:210:255:201/128;
      }
    }
  }

user@host# show policy-options
policy-statement load-balance {
  then {
    load-balance per-packet;
  }
}

user@host# show protocols
ospf {
  traceoptions {
    file r1-nsr-ospf2 size 10m files 10 world-readable;
    flag error;
    flag lsa-update detail;
    flag flooding detail;
    flag lsa-request detail;
    flag state detail;
    flag event detail;
    flag hello detail;
    flag nsr-synchronization detail;
  }
  traffic-engineering;
  area 0.0.0.0 {
    interface so-0/0/1.0 {
      metric 100;
    }
    interface fe-0/1/3.0 {
      metric 100;
    }
    interface lo0.0 {
      passive;
    }
```

```
        interface fxp0.0 {
            disable;
        }
        interface fe-1/1/0.0 {
            passive;
        }
    }
}
ospf3 {
    traceoptions {
        file r1-nsr-ospf3 size 10m world-readable;
        flag lsa-update detail;
        flag flooding detail;
        flag lsa-request detail;
        flag state detail;
        flag event detail;
        flag hello detail;
        flag nsr-synchronization detail;
    }
    area 0.0.0.0 {
        interface fe-1/1/0.0 {
            passive;
            metric 1;
        }
        interface lo0.0 {
            passive;
        }
        interface so-0/0/1.0 {
            metric 1;
        }
        interface fe-0/1/3.0 {
            metric 1;
        }
    }
}
pim {
    traceoptions {
        file r1-nsr-pim size 10m files 10 world-readable;
        flag mdt detail;
        flag rp detail;
        flag register detail;
        flag packets detail;
        flag autorp detail;
        flag join detail;
        flag hello detail;
        flag assert detail;
        flag normal detail;
        flag state detail;
        flag nsr-synchronization;
    }
    rp {
        static {
            address 10.210.255.202;
            address abcd::10:210:255:202;
        }
    }
```

```
    interface lo0.0;
    interface fe-0/1/3.0 {
      mode sparse;
      version 2;
    }
    interface so-0/0/1.0 {
      mode sparse;
      version 2;
    }
    interface fe-1/1/0.0 {
      mode sparse;
      version 2;
    }
}
```

```
user@host# show routing-options
traceoptions {
    file r1-nsr-sync size 10m;
    flag nsr-synchronization;
    flag commit-synchronize;
}
nonstop-routing;
router-id 10.210.255.201;
forwarding-table {
    traceoptions {
        file r1-nsr-krt size 10m world-readable;
        flag queue;
        flag route;
        flag routes;
        flag synchronous;
        flag state;
        flag asynchronous;
        flag consistency-checking;
    }
    export load-balance;
}
```

```
user@host# show system
syslog {
    archive size 10m;
    file messages {
        any info;
    }
}
commit synchronize;
```

Verification

To verify the configuration, run the following commands:

- show pim join **extensive**
- show pim neighbors **inet detail**
- show pim neighbors **inet6 detail**
- show pim rps **inet detail**

- show pim rps **inet6 detail**
- show multicast route **inet extensive**
- show multicast route **inet6 extensive**
- show route table **inet.1 detail**
- show route table **inet6.1 detail**

See Also
- Understanding Nonstop Active Routing for PIM on page 371

Configuring PIM Sparse Mode Graceful Restart

You can configure PIM sparse mode to continue to forward existing multicast packet streams during a routing process failure and restart. Only PIM sparse mode can be configured this way. The routing platform does not forward multicast packets for protocols other than PIM during graceful restart, because all other multicast protocols must restart after a routing process failure. If you configure PIM sparse-dense mode, only sparse multicast groups benefit from a graceful restart.

The routing platform does not forward new streams until after the restart is complete. After restart, the routing platform refreshes the forwarding state with any updates that were received from neighbors during the restart period. For example, the routing platform relearns the join and prune states of neighbors during the restart, but it does not apply the changes to the forwarding table until after the restart.

When PIM sparse mode is enabled, the routing platform generates a unique 32-bit random number called a generation identifier. Generation identifiers are included by default in PIM hello messages, as specified in the Internet draft **draft-ietf-pim-sm-v2-new-10.txt**. When a routing platform receives PIM hello messages containing generation identifiers on a point-to-point interface, the Junos OS activates an algorithm that optimizes graceful restart.

Before PIM sparse mode graceful restart occurs, each routing platform creates a generation identifier and sends it to its multicast neighbors. If a routing platform with PIM sparse mode restarts, it creates a new generation identifier and sends it to neighbors. When a neighbor receives the new identifier, it resends multicast updates to the restarting router to allow it to exit graceful restart efficiently. The restart phase is complete when the restart duration timer expires.

Multicast forwarding can be interrupted in two ways. First, if the underlying routing protocol is unstable, multicast RPF checks can fail and cause an interruption. Second, because the forwarding table is not updated during the graceful restart period, new multicast streams are not forwarded until graceful restart is complete.

You can configure graceful restart globally or for a routing instance. This example shows how to configure graceful restart globally.

To configure graceful restart for PIM sparse mode:

1. Enable graceful restart.

```
[edit protocols pim]
user@host# set graceful-restart
```

2. (Optional) Configure the amount of time the routing device waits (in seconds) to complete PIM sparse mode graceful restart. By default, the router allows 60 seconds. The range is from 30 through 300 seconds. After this restart time, the Routing Engine resumes normal multicast operation.

   ```
   [edit protocols pim graceful-restart]
   user@host# set restart-duration 120
   ```

3. Monitor the operation of PIM graceful restart by running the show pim neighbors command. In the command output, look for the **G** flag in the **Option** field. The **G** flag stands for generation identifier. Also run the **show task replication** command to verify the status of GRES and NSR.

See Also

- Understanding Nonstop Active Routing for PIM on page 371
- *Junos OS High Availability Library for Routing Devices*

Release History Table

Release	Description
13.3	In Junos OS release 13.3, multicast VPNs are not supported with nonstop active routing. Policy-based features (such as neighbor policy, join policy, BSR policy, scope policy, flow maps, and RPF check policy) are not supported with nonstop active routing.
10.4	For nonstop active routing for PIM-based multicast traffic to work with IPv6, the routing device must be running Junos OS Release 10.4 or above.

Related Documentation

- *Configuring Basic PIM Settings*

Configuring PIM-to-IGMP and PIM-to-MLD Message Translation

- Understanding PIM-to-IGMP and PIM-to-MLD Message Translation on page 384
- Configuring PIM-to-IGMP Message Translation on page 386
- Configuring PIM-to-MLD Message Translation on page 387

Understanding PIM-to-IGMP and PIM-to-MLD Message Translation

Routing devices can translate Protocol Independent Multicast (PIM) join and prune messages into corresponding Internet Group Management Protocol (IGMP) or Multicast Listener Discovery (MLD) report or leave messages. You can use this feature to forward multicast traffic across PIM domains in certain network topologies.

In some network configurations, customers are unable to run PIM between the customer edge-facing PIM domain and the core-facing PIM domain, even though PIM is running in

sparse mode within each of these domains. Because PIM is not running between the domains, customers with this configuration cannot use PIM to forward multicast traffic across the domains. Instead, they might want to use IGMP to forward IPv4 multicast traffic, or MLD to forward IPv6 multicast traffic across the domains.

To enable the use of IGMP or MLD to forward multicast traffic across the PIM domains in such topologies, you can configure the rendezvous point (RP) router that resides between the edge domain and core domain to translate PIM join or prune messages received from PIM neighbors on downstream interfaces into corresponding IGMP or MLD report or leave messages. The router then transmits the report or leave messages by proxying them to one or two upstream interfaces that you configure on the RP router. As a result, this feature is sometimes referred to as *PIM-to-IGMP proxy* or *PIM-to-MLD proxy*.

To configure the RP router to translate PIM join or prune messages into IGMP report or leave messages, include the **pim-to-igmp-proxy** statement at the **[edit routing-options multicast]** hierarchy level. Similarly, to configure the RP router to translate PIM join or prune messages into MLD report or leave messages, include the **pim-to-mld-proxy** statement at the **[edit routing-options multicast]** hierarchy level. As part of the configuration, you must specify the full name of at least one, but not more than two, upstream interfaces on which to enable the PIM-to-IGMP proxy or PIM-to-MLD proxy feature.

The following guidelines apply when you configure PIM-to-IGMP or PIM-to-MLD message translation:

- Make sure that the router connecting the PIM edge domain and the PIM core domain is the static or elected RP router.
- Make sure that the RP router is using the PIM sparse mode (PIM-SM) multicast routing protocol.
- When you configure an upstream interface, use the full logical interface specification (for example, **ge-0/0/1.0**) and not just the physical interface specification (**ge-0/0/1**).
- When you configure two upstream interfaces, the RP router transmits the same IGMP or MLD report messages and multicast traffic on both upstream interfaces. As a result, make sure that reverse-path forwarding (RPF) is running in the PIM-SM core domain to verify that multicast packets are received on the correct incoming interface and to avoid sending duplicate packets.
- The router transmits IGMP or MLD report messages on one or both upstream interfaces only for the first PIM join message that it receives among all of the downstream interfaces. Similarly, the router transmits IGMP or MLD leave messages on one or both upstream interfaces only if it receives a PIM prune message for the last downstream interface.
- Upstream interfaces support both local sources and remote sources.
- Multicast traffic received from an upstream interface is accepted as if it came from a host.

See Also
- Configuring PIM-to-IGMP Message Translation on page 386

- Configuring PIM-to-MLD Message Translation on page 387
- Understanding PIM Sparse Mode on page 209
- Enabling PIM Sparse Mode on page 217

Configuring PIM-to-IGMP Message Translation

You can configure the rendezvous point (RP) routing device to translate PIM join or prune messages into corresponding IGMP report or leave messages. To do so, include the **pim-to-igmp-proxy** statement at the **[edit routing-options multicast]** hierarchy level:

```
[edit routing-options multicast]
pim-to-igmp-proxy {
  upstream-interface [ interface-names ];
}
```

Enabling the routing device to perform PIM-to-IGMP message translation, also referred to as *PIM-to-IGMP proxy*, is useful when you want to use IGMP to forward IPv4 multicast traffic between a PIM sparse mode edge domain and a PIM sparse mode core domain in certain network topologies.

Before you begin configuring PIM-to-IGMP message translation:

- Make sure that the routing device connecting the PIM edge domain and that the PIM core domain is the static or elected RP routing device.
- Make sure that the PIM sparse mode (PIM-SM) routing protocol is running on the RP routing device.
- If you plan to configure two upstream interfaces, make sure that reverse-path forwarding (RPF) is running in the PIM-SM core domain. Because the RP router transmits the same IGMP messages and multicast traffic on both upstream interfaces, you need to run RPF to verify that multicast packets are received on the correct incoming interface and to avoid sending duplicate packets.

To configure the RP routing device to translate PIM join or prune messages into corresponding IGMP report or leave messages:

1. Include the **pim-to-igmp-proxy** statement, specifying the names of one or two logical interfaces to function as the upstream interfaces on which the routing device transmits IGMP report or leave messages.

   The following example configures PIM-to-IGMP message translation on a single upstream interface, **ge-0/1/0.1**.

   ```
   [edit routing-options multicast]
   user@host# set pim-to-igmp-proxy upstream-interface ge-0/1/0.1
   ```

   The following example configures PIM-to-IGMP message translation on two upstream interfaces, **ge-0/1/0.1** and **ge-0/1/0.2**. You must include the logical interface names within square brackets ([ ]) when you configure a set of two upstream interfaces.

   ```
   [edit routing-options multicast]
   user@host# set pim-to-igmp-proxy upstream-interface [ge-0/1/0.1 ge-0/1/0.2]
   ```

2. Use the **show multicast pim-to-igmp-proxy** command to display the PIM-to-IGMP proxy state (enabled or disabled) and the name or names of the configured upstream interfaces.

```
user@host# run show multicast pim-to-igmp-proxy
Proxy state: enabled
ge-0/1/0.1
ge-0/1/0.2
```

See Also

- Understanding PIM-to-IGMP and PIM-to-MLD Message Translation on page 384
- pim-to-igmp-proxy on page 1206
- upstream-interface on page 1365

Configuring PIM-to-MLD Message Translation

You can configure the rendezvous point (RP) routing device to translate PIM join or prune messages into corresponding MLD report or leave messages. To do so, include the **pim-to-mld-proxy** statement at the **[edit routing-options multicast]** hierarchy level:

```
[edit routing-options multicast]
pim-to-mld-proxy {
  upstream-interface [ interface-names ];
}
```

Enabling the routing device to perform PIM-to-MLD message translation, also referred to as *PIM-to-MLD proxy*, is useful when you want to use MLD to forward IPv6 multicast traffic between a PIM sparse mode edge domain and a PIM sparse mode core domain in certain network topologies.

Before you begin configuring PIM-to-MLD message translation:

- Make sure that the routing device connecting the PIM edge domain and that the PIM core domain is the static or elected RP routing device.
- Make sure that the PIM sparse mode (PIM-SM) routing protocol is running on the RP routing device.
- If you plan to configure two upstream interfaces, make sure that reverse-path forwarding (RPF) is running in the PIM-SM core domain. Because the RP routing device transmits the same MLD messages and multicast traffic on both upstream interfaces, you need to run RPF to verify that multicast packets are received on the correct incoming interface and to avoid sending duplicate packets.

To configure the RP routing device to translate PIM join or prune messages into corresponding MLD report or leave messages:

1. Include the **pim-to-mld-proxy** statement, specifying the names of one or two logical interfaces to function as the upstream interfaces on which the router transmits MLD report or leave messages.

The following example configures PIM-to-MLD message translation on a single upstream interface, **ge-0/5/0.1**.

```
[edit routing-options multicast]
user@host# set pim-to-mld-proxy upstream-interface ge-0/5/0.1
```

The following example configures PIM-to-MLD message translation on two upstream interfaces, **ge-0/5/0.1** and **ge-0/5/0.2**. You must include the logical interface names within square brackets ([ ]) when you configure a set of two upstream interfaces.

```
[edit routing-options multicast]
user@host# set pim-to-mld-proxy upstream-interface [ge-0/5/0.1 ge-0/5/0.2]
```

2. Use the **show multicast pim-to-mld-proxy** command to display the PIM-to-MLD proxy state (enabled or disabled) and the name or names of the configured upstream interfaces.

```
user@host# run show multicast pim-to-mld-proxy
Proxy state: enabled
ge-0/5/0.1
ge-0/5/0.2
```

See Also

- Understanding PIM-to-IGMP and PIM-to-MLD Message Translation on page 384
- pim-to-mld-proxy on page 1207
- upstream-interface on page 1365

Related Documentation

- Configuring IGMP on page 23
- Examples: Configuring MLD on page 50

CHAPTER 15

Verifying PIM Configurations

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From the CLI, enter the **show pim interfaces** command.

Sample Output

```
user@host> show pim interfaces
Instance: PIM.master
Name                 Stat Mode       IP V State Count DR address
lo0.0                Up   Sparse      4 2 DR        0 127.0.0.1
pime.32769           Up   Sparse      4 2 P2P       0
```

Meaning The output shows a list of the interfaces that are configured for PIM. Verify the following information:

- Each interface on which PIM is enabled is listed.
- The network management interface, either **ge–0/0/0** or **fe–0/0/0**, is *not* listed.
- Under **Mode**, the word **Sparse** appears.

Verifying the PIM RP Configuration

Purpose Verify that the PIM RP is statically configured with the correct IP address.

Action From the CLI, enter the **show pim rps** command.

Sample Output

```
user@host> show pim rps
```

```
Instance: PIM.master
Address family INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
192.168.14.27   static           0    None             2 224.0.0.0/4
```

Meaning The output shows a list of the RP addresses that are configured for PIM. At least one RP must be configured. Verify the following information:

- The configured RP is listed with the proper IP address.
- Under **Type**, the word **static** appears.

Verifying the RPF Routing Table Configuration

Purpose Verify that the PIM RPF routing table is configured correctly.

Action From the CLI, enter the **show multicast rpf** command.

Sample Output

```
user@host> show multicast rpf
Multicast RPF table: inet.0 , 2 entries...
```

Meaning The output shows the multicast RPF table that is configured for PIM. If no multicast RPF routing table is configured, RPF checks use **inet.0**. Verify the following information:

- The configured multicast RPF routing table is **inet.0**.
- The **inet.0** table contains entries.

PART 4

Configuring Multicast Routing Protocols

CHAPTER 16

Connecting Routing Domains Using MSDP

Examples: Configuring MSDP

Understanding MSDP

The Multicast Source Discovery Protocol (MSDP) is used to connect multicast routing domains. It typically runs on the same router as the Protocol Independent Multicast (PIM) sparse-mode rendezvous point (RP). Each MSDP router establishes adjacencies with internal and external MSDP peers similar to the way BGP establishes peers. These peer routers inform each other about active sources within the domain. When they detect active sources, the routers can send PIM sparse-mode explicit join messages to the active source.

The peer with the higher IP address passively listens to a well-known port number and waits for the side with the lower IP address to establish a Transmission Control Protocol (TCP) connection. When a PIM sparse-mode RP that is running MSDP becomes aware of a new local source, it sends source-active type, length, and values (TLVs) to its MSDP peers. When a source-active TLV is received, a peer-reverse-path-forwarding (peer-RPF) check (not the same as a multicast RPF check) is done to make sure that this peer is in the path that leads back to the originating RP. If not, the source-active TLV is dropped. This TLV is counted as a “rejected” source-active message.

The MSDP peer-RPF check is different from the normal RPF checks done by non-MSDP multicast routers. The goal of the peer-RPF check is to stop source-active messages

from looping. Router R accepts source-active messages originated by Router S only from neighbor Router N or an MSDP mesh group member.

S ----------> N ----------> R

Router R (the router that accepts or rejects active-source messages) locates its MSDP peer-RPF neighbor (Router N) deterministically. A series of rules is applied in a particular order to received source-active messages, and the first rule that applies determines the peer-RPF neighbor. All source-active messages from other routers are rejected.

The six rules applied to source-active messages originating at Router S received at Router R from Router N are as follows:

1. If Router N originated the source-active message (Router N is Router S), then Router N is also the peer-RPF neighbor, and its source-active messages are accepted.
2. If Router N is a member of the Router R mesh group, or is the configured peer, then Router N is the peer-RPF neighbor, and its source-active messages are accepted.
3. If Router N is the BGP next hop of the active multicast RPF route toward Router S (Router N installed the route on Router R), then Router N is the peer-RPF neighbor, and its source-active messages are accepted.
4. If Router N is an external BGP (EBGP) or internal BGP (IBGP) peer of Router R, and the last autonomous system (AS) number in the BGP AS-path to Router S is the same as Router N's AS number, then Router N is the peer-RPF neighbor, and its source-active messages are accepted.
5. If Router N uses the same next hop as the next hop to Router S, then Router N is the peer-RPF neighbor, and its source-active messages are accepted.
6. If Router N fits none of these criteria, then Router N is not an MSDP peer-RPF neighbor, and its source-active messages are rejected.

The MSDP peers that receive source-active TLVs can be constrained by BGP reachability information. If the AS path of the network layer reachability information (NLRI) contains the receiving peer's AS number prepended second to last, the sending peer is using the receiving peer as a next hop for this source. If the split horizon information is not being received, the peer can be pruned from the source-active TLV distribution list.

For information about configuring MSDP mesh groups, see "Example: Configuring MSDP with Active Source Limits and Mesh Groups" on page 404.

See Also

- Configuring MSDP on page 394

Configuring MSDP

To configure the Multicast Source Discovery Protocol (MSDP), include the **msdp** statement:

```
msdp {
  disable;
  active-source-limit {
```

```
    maximum number;
    threshold number;
}
data-encapsulation (disable | enable);
export [ policy-names ];
group group-name {
    ... group-configuration ...
}
hold-time seconds;
import [ policy-names ];
local-address address;
keep-alive seconds;
peer address {
    ... peer-configuration ...
}
rib-group group-name;
source ip-prefix</prefix-length> {
    active-source-limit {
        maximum number;
        threshold number;
    }
}
sa-hold-time seconds;
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier > <disable>;
}
group group-name {
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    mode (mesh-group | standard);
    peer address {
        ... same statements as at the [edit protocols msdp peer address] hierarchy level shown
            just following ...
    }
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
peer address {
    disable;
    active-source-limit {
        maximum number;
        threshold number;
    }
    authentication-key peer-key;
    default-peer;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
```

```
    }
  }
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols]**
- **[edit routing-instances *routing-instance-name* protocols]**
- **[edit logical-systems *logical-system-name* protocols]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]**

By default, MSDP is disabled.

See Also
- Example: Configuring MSDP in a Routing Instance on page 396
- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

Example: Configuring MSDP in a Routing Instance

This example shows how to configure MSDP in a VRF instance.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Enable PIM. See "PIM Overview" on page 185.

Overview

You can configure MSDP in the following types of instances:

- Forwarding
- No forwarding
- Virtual router
- VPLS
- VRF

The main use of MSDP in a routing instance is to support anycast RPs in the network, which allows you to configure redundant RPs. Anycast RP addressing requires MSDP support to synchronize the active sources between RPs.

This example includes the following MSDP settings.

- **authentication-key**—By default, multicast routers accept and process any properly formatted MSDP messages from the configured peer address. This default behavior might violate the security policies in many organizations because MSDP messages by definition come from another routing domain beyond the control of the security practices of the multicast router's organization.

  The router can authenticate MSDP messages using the TCP message digest 5 (MD5) signature option for MSDP peering sessions. This authentication provides protection against spoofed packets being introduced into an MSDP peering session. Two organizations implementing MSDP authentication must decide on a human-readable key on both peers. This key is included in the MD5 signature computation for each MSDP segment sent between the two peers.

  You configure an MSDP authentication key on a per-peer basis, whether the MSDP peer is defined in a group or individually. If you configure different authentication keys for the same peer one in a group and one individually, the individual key is used.

  The peer key can be a text string up to 16 letters and digits long. Strings can include any ASCII characters with the exception of (,), &, and [. If you include spaces in an MSDP authentication key, enclose all characters in quotation marks (“ ”).

  Adding, removing, or changing an MSDP authentication key in a peering session resets the existing MSDP session and establishes a new session between the affected MSDP peers. This immediate session termination prevents excessive retransmissions and eventual session timeouts due to mismatched keys.

- **import** and **export**—All routing protocols use the routing table to store the routes that they learn and to determine which routes they advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in, and retrieve from, the routing table.

  You can configure routing policy globally, for a group, or for an individual peer. This example shows how to configure the policy for an individual peer.

  If you configure routing policy at the group level, each peer in a group inherits the group's routing policy.

  The **import** statement applies policies to source-active messages being imported into the source-active cache from MSDP. The **export** statement applies policies to source-active messages being exported from the source-active cache into MSDP. If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching policy is applied to the route. If no match is found for the import policy, MSDP shares with the routing table only those routes that were learned from MSDP routers. If no match is found for the export policy, the default MSDP export policy is applied to entries in the source-active cache. See Table 15 on page 398 for a list of match conditions.

Table 15: MSDP Source-Active Message Filter Match Conditions

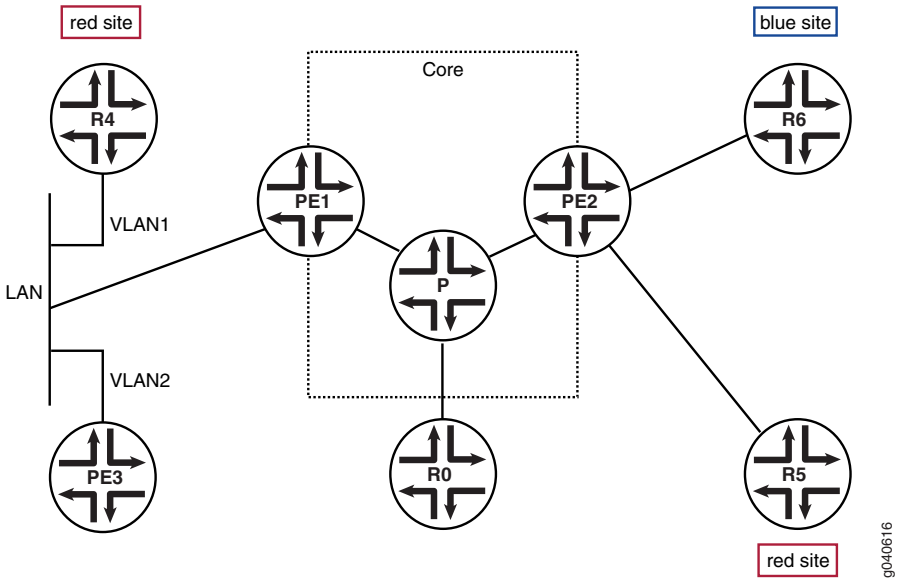
Match Condition	Matches On
interface	Router interface or interfaces specified by name or IP address
neighbor	Neighbor address (the source address in the IP header of the source-active message)
route-filter	Multicast group address embedded in the source-active message
source-address-filter	Multicast source address embedded in the source-active message

- **local-address**—Identifies the address of the router you are configuring as an MSDP router (the local router). When you configure MSDP, the **local-address** statement is required. The router must also be a Protocol Independent Multicast (PIM) sparse-mode rendezvous point (RP).
- **peer**—An MSDP router must know which routers are its peers. You define the peer relationships explicitly by configuring the neighboring routers that are the MSDP peers of the local router. After peer relationships are established, the MSDP peers exchange messages to advertise active multicast sources. You must configure at least one peer for MSDP to function. When you configure MSDP, the **peer** statement is required. The router must also be a Protocol Independent Multicast (PIM) sparse-mode rendezvous point (RP).

  You can arrange MSDP peers into groups. Each group must contain at least one peer. Arranging peers into groups is useful if you want to block sources from some peers and accept them from others, or set tracing options on one group and not others. This example shows how to configure the MSDP peers in groups. If you configure MSDP peers in a group, each peer in a group inherits all group-level options.

Figure 66 on page 399 shows the topology for this example.

Figure 66: MSDP in a VRF Instance Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set policy-options policy-statement bgp-to-ospf term 1 from protocol bgp
set policy-options policy-statement bgp-to-ospf term 1 then accept
set policy-options policy-statement sa-filter term bad-groups from route-filter 224.0.1.2/32
  exact
set policy-options policy-statement sa-filter term bad-groups from route-filter
  224.77.0.0/16 orlonger
set policy-options policy-statement sa-filter term bad-groups then reject
set policy-options policy-statement sa-filter term bad-sources from source-address-filter
  10.0.0.0/8 orlonger
set policy-options policy-statement sa-filter term bad-sources from source-address-filter
  127.0.0.0/8 orlonger
set policy-options policy-statement sa-filter term bad-sources then reject
set policy-options policy-statement sa-filter term accept-everything-else then accept
set routing-instances VPN-100 instance-type vrf
set routing-instances VPN-100 interface ge-0/0/0.100
set routing-instances VPN-100 interface lo0.100
set routing-instances VPN-100 route-distinguisher 10.255.120.36:100
set routing-instances VPN-100 vrf-target target:100:1
set routing-instances VPN-100 protocols ospf export bgp-to-ospf
set routing-instances VPN-100 protocols ospf area 0.0.0.0 interface lo0.100
set routing-instances VPN-100 protocols ospf area 0.0.0.0 interface ge-0/0/0.100
set routing-instances VPN-100 protocols pim rp static address 11.11.47.100
set routing-instances VPN-100 protocols pim interface lo0.100 mode sparse-dense
set routing-instances VPN-100 protocols pim interface lo0.100 version 2
set routing-instances VPN-100 protocols pim interface ge-0/0/0.100 mode sparse-dense
```

```
set routing-instances VPN-100 protocols pim interface ge-0/0/0.100 version 2
set routing-instances VPN-100 protocols msdp export sa-filter
set routing-instances VPN-100 protocols msdp import sa-filter
set routing-instances VPN-100 protocols msdp group 100 local-address 10.10.47.100
set routing-instances VPN-100 protocols msdp group 100 peer 10.255.120.39
  authentication-key “New York”
set routing-instances VPN-100 protocols msdp group to_pe local-address 10.10.47.100
set routing-instances VPN-100 protocols msdp group to_pe peer 11.11.47.100
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an MSDP routing instance:

1. Configure the BGP export policy.

```
[edit policy-options]
user@host# set policy-statement bgp-to-ospf term 1 from protocol bgp
user@host# set policy-statement bgp-to-ospf term 1 then accept
```

2. Configure a policy that filters out certain source and group addresses and accepts all other source and group addresses.

```
[edit policy-options]
user@host# set policy-statement sa-filter term bad-groups from route-filter
  224.0.1.2/32 exact
user@host# set policy-statement sa-filter term bad-groups from route-filter
  224.0.1.2/32 exact
user@host# set policy-statement sa-filter term bad-groups from route-filter
  224.77.0.0/16 orlonger
user@host# set policy-statement sa-filter term bad-groups then reject
user@host# set policy-statement sa-filter term bad-sources from
  source-address-filter 10.0.0.0/8 orlonger
user@host# set policy-statement sa-filter term bad-sources from
  source-address-filter 127.0.0.0/8 orlonger
user@host# set policy-statement sa-filter term bad-sources then reject
user@host# set policy-statement sa-filter term accept-everything-else then accept
```

3. Configure the routing instance type and interfaces.

```
[edit routing-instances]
user@host# set VPN-100 instance-type vrf
user@host# set VPN-100 interface ge-0/0/0.100
user@host# set VPN-100 interface lo0.100
```

4. Configure the routing instance route distinguisher and VRF target.

```
[edit routing-instances]
user@host# set VPN-100 route-distinguisher 10.255.120.36:100
user@host# set VPN-100 vrf-target target:100:1
```

5. Configure OSPF in the routing instance.

```
[edit routing-instances]
user@host# set VPN-100 protocols ospf export bgp-to-ospf
user@host# set VPN-100 protocols ospf area 0.0.0.0 interface lo0.100
user@host# set VPN-100 protocols ospf area 0.0.0.0 interface ge-0/0/0.100
```

6. Configure PIM in the routing instance.

```
[edit routing-instances]
user@host# set VPN-100 protocols pim rp static address 11.11.47.100
user@host# set VPN-100 protocols pim interface lo0.100 mode sparse-dense
user@host# set VPN-100 protocols pim interface lo0.100 version 2
user@host# set VPN-100 protocols pim interface ge-0/0/0.100 mode sparse-dense
user@host# set VPN-100 protocols pim interface ge-0/0/0.100 version 2
```

7. Configure MSDP in the routing instance.

```
[edit routing-instances]
user@host# set VPN-100 protocols msdp export sa-filter
user@host# set VPN-100 protocols msdp import sa-filter
user@host# set VPN-100 protocols msdp group 100 local-address 10.10.47.100
user@host# set VPN-100 protocols msdp group 100 peer 10.255.120.39
  authentication-key "New York"
[edit routing-instances]
user@host# set VPN-100 protocols msdp group to_pe local-address 10.10.47.100
[edit routing-instances]
user@host# set VPN-100 protocols msdp group to_pe peer 11.11.47.100
```

8. If you are done configuring the device, commit the configuration.

```
[edit routing-instances]
user@host# commit
```

Results

Confirm your configuration by entering the **show policy-options** command and the **show routing-instances** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement bgp-to-ospf {
  term 1 {
    from protocol bgp;
    then accept;
  }
}
  policy-statement sa-filter {
    term bad-groups {
      from {
        route-filter 224.0.1.2/32 exact;
        route-filter 224.77.0.0/16 orlonger;
      }
      then reject;
    }
```

```
    term bad-sources {
      from {
        source-address-filter 10.0.0.0/8 orlonger;
        source-address-filter 127.0.0.0/8 orlonger;
      }
      then reject;
    }
    term accept-everything-else {
      then accept;
    }
  }
user@host# show routing-instances
VPN-100 {
  instance-type vrf;
  interface ge-0/0/0.100; ## 'ge-0/0/0.100' is not defined
  interface lo0.100; ## 'lo0.100' is not defined
  route-distinguisher 10.255.120.36:100;
  vrf-target target:100:1;
  protocols {
    ospf {
      export bgp-to-ospf;
      area 0.0.0.0 {
        interface lo0.100;
        interface ge-0/0/0.100;
      }
    }
    pim {
      rp {
        static {
          address 11.11.47.100;
        }
      }
      interface lo0.100 {
        mode sparse-dense;
        version 2;
      }
      interface ge-0/0/0.100 {
        mode sparse-dense;
        version 2;
      }
    }
    msdp {
      export sa-filter;
      import sa-filter;
      group 100 {
        local-address 10.10.47.100;
        peer 10.255.120.39 {
          authentication-key "Hashed key found - Replaced with $ABC123abc123"; ##
            SECRET-DATA
        }
      }
      group to_pe {
        local-address 10.10.47.100;
        peer 11.11.47.100;
      }
```

```
    }
  }
}
```

Verification

To verify the configuration, run the following commands:

- **show msdp instance VPN-100**
- **show msdp source-active VPN-100**
- **show multicast usage instance VPN-100**
- **show route table VPN-100.inet.4**

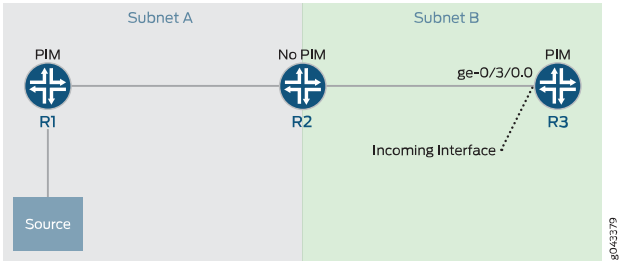
See Also

- Configuring Local PIM RPs on page 237
- Example: Configuring PIM Anycast With or Without MSDP on page 248

Configuring the Interface to Accept Traffic from a Remote Source

You can configure an incoming interface to accept multicast traffic from a remote source. A remote source is a source that is not on the same subnet as the incoming interface. Figure 67 on page 403 shows such a topology, where R2 connects to the R1 source on one subnet, and to the incoming interface on R3 (ge-1/3/0.0 in the figure) on another subnet.

Figure 67: Accepting Multicast Traffic from a Remote Source



In this topology R2 is a pass-through device not running PIM, so R3 is the first hop router for multicast packets sent from R1. Because R1 and R3 are in different subnets, the default behavior of R3 is to disregard R1 as a remote source. You can have R3 accept multicast traffic from R1, however, by enabling **accept-remote-source** on the target interface.

To accept traffic from a remote source:

1. Identify the router and physical interface that you want to receive multicast traffic from the remote source.

2. Configure the interface to accept traffic from the remote source.

   ```
   [edit protocols pim interface ge-1/3/0.0]
   user@host# set accept-remote-source
   ```

NOTE: If the interface you identified is not the only path from the remote source, you need to ensure that it is the best path. For example you can configure a static route on the receiver side PE router to the source, or you can prepend the AS path on the other possible routes:

```
[edit policy-options policy-statement as-path-prepend term prepend]
user@host# set from route-filter 192.168.0.0/16 orlonger
user@host# set from route-filter 172.16.0.0/16 orlonger
user@host# set then as-path-prepend "1111"
```

3. Commit the configuration changes.

4. Confirm that the interface you configured accepts traffic from the remote source.

   user@host# show pim statistics

See Also

- Example: Allowing MBGP MVPN Remote Sources on page 633
- *Understanding Prepending AS Numbers to BGP AS Paths*
- show pim statistics on page 1689

Example: Configuring MSDP with Active Source Limits and Mesh Groups

This example shows how to configure MSDP to filter source-active messages and limit the flooding of source-active messages.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Enable PIM sparse mode. See "PIM Overview" on page 185.
- Configure the router as a PIM sparse-mode RP. See "Configuring Local PIM RPs" on page 237.

Overview

A router interested in MSDP messages, such as an RP, might have to process a large number of MSDP messages, especially source-active messages, arriving from other

routers. Because of the potential need for a router to examine, process, and create state tables for many MSDP packets, there is a possibility of an MSDP-based denial-of-service (DoS) attack on a router running MSDP. To minimize this possibility, you can configure the router to limit the number of source active messages the router accepts. Also, you can configure a threshold for applying random early detection (RED) to drop some but not all MSDP active source messages.

By default, the router accepts 25,000 source active messages before ignoring the rest. The limit can be from 1 through 1,000,000. The limit is applied to both the number of messages and the number of MSDP peers.

By default, the router accepts 24,000 source-active messages before applying the RED profile to prevent a possible DoS attack. This number can also range from 1 through 1,000,000. The next 1000 messages are screened by the RED profile and the accepted messages processed. If you configure no drop profiles (as this example does not), RED is still in effect and functions as the primary mechanism for managing congestion. In the default RED drop profile, when the packet queue fill-level is 0 percent, the drop probability is 0 percent. When the fill-level is 100 percent, the drop probability is 100 percent.

NOTE: **The router ignores source-active messages with encapsulated TCP packets. Multicast does not use TCP; segments inside source-active messages are most likely the result of worm activity.**

The number configured for the threshold must be less than the number configured for the maximum number of active MSDP sources.

You can configure an active source limit globally, for a group, or for a peer. If active source limits are configured at multiple levels of the hierarchy (as shown in this example), all are applied.

You can configure an active source limit for an address range as well as for a specific peer. A per-source active source limit uses an IP prefix and prefix length instead of a specific address. You can configure more than one per-source active source limit. The longest match determines the limit.

Per-source active source limits can be combined with active source limits at the peer, group, and global (instance) hierarchy level. Per-source limits are applied before any other type of active source limit. Limits are tested in the following order:

- Per-source
- Per-peer or group
- Per-instance

An active source message must “pass” all limits established before being accepted. For example, if a source is configured with an active source limit of 10,000 active multicast groups and the instance is configured with a limit of 5000 (and there are no other sources or limits configured), only 5000 active source messages are accepted from this source.

MSDP mesh groups are groups of peers configured in a full-mesh topology that limits the flooding of source-active messages to neighboring peers. Every mesh group member must have a peer connection with every other mesh group member. When a source-active message is received from a mesh group member, the source-active message is always accepted but is not flooded to other members of the same mesh group. However, the source-active message is flooded to non-mesh group peers or members of other mesh groups. By default, standard flooding rules apply if **mesh-group** is not specified.

CAUTION: When configuring MSDP mesh groups, you must configure all members the same way. If you do not configure a full mesh, excessive flooding of source-active messages can occur.

A common application for MSDP mesh groups is peer-reverse-path-forwarding (peer-RPF) check bypass. For example, if there are two MSDP peers inside an autonomous system (AS), and only one of them has an external MSDP session to another AS, the internal MSDP peer often rejects incoming source-active messages relayed by the peer with the external link. Rejection occurs because the external MSDP peer must be reachable by the internal MSDP peer through the next hop toward the source in another AS, and this next-hop condition is not certain. To prevent rejections, configure an MSDP mesh group on the internal MSDP peer so it always accepts source-active messages.

NOTE: An alternative way to bypass the peer-RPF check is to configure a default peer. In networks with only one MSDP peer, especially stub networks, the source-active message always needs to be accepted. An MSDP default peer is an MSDP peer from which all source-active messages are accepted without performing the peer-RPF check. You can establish a default peer at the peer or group level by including the default-peer statement.

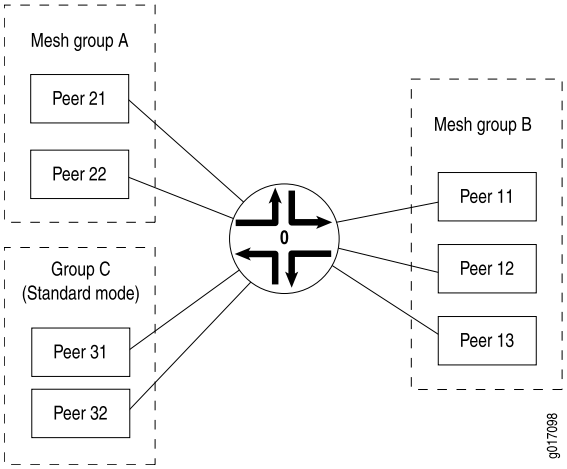
Table 16 on page 406 explains how flooding is handled by peers in this example. .

Table 16: Source-Active Message Flooding Explanation

Source-Active Message Received From	Source-Active Message Flooded To	Source-Active Message Not Flooded To
Peer 21	Peer 11, Peer 12, Peer 13, Peer 31, Peer 32	Peer 22
Peer 11	Peer 21, Peer 22, Peer 31, Peer 32	Peer 12, Peer 13
Peer 31	Peer 21, Peer 22, Peer 11, Peer 12, Peer 13, Peer 32	–

Figure 68 on page 407 illustrates source-active message flooding between different mesh groups and peers within the same mesh group.

Figure 68: Source-Active Message Flooding



This example includes the following settings:

- **active-source-limit maximum 10000**—Applies a limit of 10,000 active sources to all other peers.
- **data-encapsulation disable**—On an RP router using MSDP, disables the default encapsulation of multicast data received in MSDP register messages inside MSDP source-active messages.

  MSDP data encapsulation mainly concerns bursty sources of multicast traffic. Sources that send only one packet every few minutes have trouble with the timeout of state relationships between sources and their multicast groups (S,G). Routers lose data while they attempt to reestablish (S,G) state tables. As a result, multicast register messages contain data, and this data encapsulation in MSDP source-active messages can be turned on or off through configuration.

  By default, MSDP data encapsulation is enabled. An RP running MSDP takes the data packets arriving in the source's register message and encapsulates the data inside an MSDP source-active message.

  However, data encapsulation creates both a multicast forwarding cache entry in the **inet.1** table (this is also the forwarding table) and a routing table entry in the **inet.4** table. Without data encapsulation, MSDP creates only a routing table entry in the **inet.4** table. In some circumstances, such as the presence of Internet worms or other forms of DoS attack, the router's forwarding table might fill up with these entries. To prevent the forwarding table from filling up with MSDP entries, you can configure the router not to use MSDP data encapsulation. However, if you disable data encapsulation, the router ignores and discards the encapsulated data. Without data encapsulation, multicast applications with bursty sources having transmit intervals greater than about 3 minutes might not work well.
- **group MSDP-group local-address 10.1.2.3**—Specifies the address of the local router (this router).
- **group MSDP-group mode mesh-group**—Specifies that all peers belonging to the **MSDP-group** group are mesh group members.

- **group MSDP-group peer 10.10.10.10**—Prevents the sending of source-active messages to neighboring peer 10.10.10.10.
- **group MSDP-group peer 10.10.10.10 active-source-limit maximum 7500**—Applies a limit of 7500 active sources to MSDP peer 10.10.10.10 in group **MSDP-group**.
- **peer 10.0.0.1 active-source-limit maximum 5000 threshold 4000**—Applies a threshhold of 4000 active sources and a limit of 5000 active sources to MSDP peer 10.0.0.1.
- **source 10.1.0.0/16 active-source-limit maximum 500**—Applies a limit of 500 active sources to any source on the 10.1.0.0/16 network.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols msdp data-encapsulation disable
set protocols msdp active-source-limit maximum 10000
set protocols msdp peer 10.0.0.1 active-source-limit maximum 5000
set protocols msdp peer 10.0.0.1 active-source-limit threshold 4000
set protocols msdp source 10.1.0.0/16 active-source-limit maximum 500
set protocols msdp group MSDP-group mode mesh-group
set protocols msdp group MSDP-group local-address 10.1.2.3
set protocols msdp group MSDP-group peer 10.10.10.10 active-source-limit maximum
  7500
```

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure MSDP source active routes and mesh groups:

1. (Optional) Disable data encapsulation.

   ```
   [edit protocols msdp]
   user@host# set data-encapsulation disable
   ```

2. Configure the active source limits.

   ```
   [edit protocols msdp]
   user@host# set peer 10.0.0.1 active-source-limit maximum 5000 threshold 4000
   user@host# set group MSDP-group peer 10.10.10.10 active-source-limit maximum
     7500
   user@host# set active-source-limit maximum 10000
   user@host# set source 10.1.0.0/16 active-source-limit maximum 500
   ```

3. (Optional) Configure the threshold at which warning messages are logged and the amount of time between log messages.

   ```
   [edit protocols msdp]
   ```

```
user@host# set active-source-limit log-warning 80
user@host# set active-source-limit log-interval 20
```

4. Configure the mesh group.

```
[edit protocols msdp]
user@host# set group MSDP-group mode mesh-group
user@host# set group MSDP-group peer 10.10.10.10
user@host# set group MSDP-group local-address 10.1.2.3
```

5. If you are done configuring the device, commit the configuration.

```
[edit routing-instances]
user@host# commit
```

Results

Confirm your configuration by entering the **show protocols** command.

```
user@host# show protocols
msdp {
    data-encapsulation disable;
    active-source-limit {
        maximum 10000;
    }
    peer 10.0.0.1 {
        active-source-limit {
            maximum 5000;
            threshold 4000;
        }
    }
    source 10.1.0.0/16 {
        active-source-limit {
            maximum 500;
        }
    }
    group MSDP-group {
        mode mesh-group;
        local-address 10.1.2.3;
        peer 10.10.10.10 {
            active-source-limit {
                maximum 7500;
            }
        }
    }
}
```

Verification

To verify the configuration, run the following commands:

- show msdp source-active
- show msdp statistics

See Also

- Examples: Configuring MSDP on page 393
- Filtering MSDP SA Messages on page 269
- Configuring Local PIM RPs on page 237

Tracing MSDP Protocol Traffic

Tracing operations record detailed messages about the operation of routing protocols, such as the various types of routing protocol packets sent and received, and routing policy actions. You can specify which trace operations are logged by including specific tracing flags. The following table describes the flags that you can include.

Flag	Description
all	Trace all operations.
general	Trace general events.
keepalive	Trace keepalive messages.
normal	Trace normal events.
packets	Trace all MSDP packets.
policy	Trace policy processing.
route	Trace MSDP changes to the routing table.
source-active	Trace source-active packets.
source-active-request	Trace source-active request packets.
source-active-response	Trace source-active response packets.
state	Trace state transitions.
task	Trace task processing.
timer	Trace timer processing.

You can configure MSDP tracing for all peers, for all peers in a particular group, or for a particular peer.

In the following example, tracing is enabled for all routing protocol packets. Then tracing is narrowed to focus only on MSDP peers in a particular group. To configure tracing operations for MSDP:

1. (Optional) Configure tracing by including the **traceoptions** statement at the **[edit routing-options]** hierarchy level and set the **all-packets-trace** and **all** flags to trace all protocol packets.

   ```
   [edit routing-options traceoptions]
   user@host# set file all-packets-trace
   user@host# set flag all
   ```

2. Configure the filename for the MSDP trace file.

   ```
   [edit protocols msdp group groupa traceoptions]
   user@host# set file msdp-trace
   ```

3. (Optional) Configure the maximum number of trace files.

   ```
   [edit protocols msdp group groupa traceoptions]
   user@host# set file files 5
   ```

4. (Optional) Configure the maximum size of each trace file.

   ```
   [edit protocols msdp group groupa traceoptions]
   user@host# set file size 1m
   ```

5. (Optional) Enable unrestricted file access.

   ```
   [edit protocols msdp group groupa traceoptions]
   user@host# set file world-readable
   ```

6. Configure tracing flags. Suppose you are troubleshooting issues with the source-active cache for **groupa**. The following example shows how to trace messages associated with the group address.

   ```
   [edit protocols msdp group groupa traceoptions]
   user@host# set flag source-active | match 230.0.0.3
   ```

7. View the trace file.

   ```
   user@host> file list /var/log
   user@host> file show /var/log/msdp-trace
   ```

See Also

- Understanding MSDP on page 393
- *Tracing and Logging Junos OS Operations* in the *Junos OS Administration Library*

Disabling MSDP

To disable MSDP on the router, include the **disable** statement:

```
disable;
```

You can disable MSDP globally for all peers, for all peers in a group, or for an individual peer.

- Globally for all MSDP peers at the following hierarchy levels:
  - **[edit protocols msdp]**
  - **[edit logical-systems *logical-system-name* protocols msdp]**
  - **[edit routing-instances *routing-instance-name* protocols msdp]**
  - **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp]**
- For all peers in a group at the following hierarchy levels:
  - **[edit protocols msdp group *group-name*]**
  - **[edit logical-systems *logical-system-name* protocols msdp group *group-name*]**
  - **[edit routing-instances *routing-instance-name* protocols msdp group *group-name*]**
  - **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*]**
- For an individual peer at the following hierarchy levels:
  - **[edit protocols msdp peer *address*]**
  - **[edit protocols msdp group *group-name* peer *address*]**
  - **[edit logical-systems *logical-system-name* protocols msdp peer *address*]**
  - **[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*]**
  - **[edit routing-instances *routing-instance-name* protocols msdp peer *address*]**
  - **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*]**
  - **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*]**

If you disable MSDP at the group level, each peer in the group is disabled.

See Also
- Example: Configuring MSDP in a Routing Instance on page 396

Example: Configuring MSDP

Configure a router to act as a PIM sparse-mode rendezvous point and an MSDP peer:

```
[edit]
routing-options {
  interface-routes {
    rib-group ifrg;
  }
  rib-groups {
    ifrg {
      import-rib [inet.0 inet.2];
    }
    mcrg {
      export-rib inet.2;
      import-rib inet.2;
    }
  }
}
protocols {
  bgp {
    group lab {
      type internal;
      family any;
      neighbor 192.168.6.18 {
        local-address 192.168.6.17;
      }
    }
  }
  pim {
    dense-groups {
      224.0.1.39/32;
      224.0.1.40/32;
    }
    rib-group mcrg;
    rp {
      local {
        address 192.168.1.1;
      }
    }
    interface all {
      mode sparse-dense;
      version 1;
    }
  }
  msdp {
    rib-group mcrg;
    group lab {
      peer 192.168.6.18 {
        local-address 192.168.6.17;
      }
    }
  }
}
```

Related Documentation

- Understanding MSDP on page 393

Configuring Multiple Instances of MSDP

MSDP instances are supported for VRF instance types. For QFX5100, QFX5110, QFX5200, and EX9200 switches, MSDP instances are also supported for default and virtual router instance types. You can configure multiple instances of MSDP to support multicast over VPNs.

To configure multiple instances of MSDP, include the following statements:

```
routing-instances {
  routing-instance-name {
    interface interface-name;
    instance-type vrf;
    route-distinguisher (as-number:number | ip-address:number);
    vrf-import [ policy-names ];
    vrf-export [ policy-names ];
    protocols {
      msdp {
        ... msdp-configuration ...
      }
    }
  }
}
```

You can include the statements at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name* protocols]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]**

Related Documentation

- Example: Configuring MSDP in a Routing Instance on page 396
- *Junos OS MPLS Applications Library for Routing Devices*
- *Junos OS VPNs Library for Routing Devices*

CHAPTER 17

Handling Session Announcements with SAP and SDP

Configuring the Session Announcement Protocol

Understanding SAP and SDP

Session announcements are handled by two protocols: the Session Announcement Protocol (SAP) and the Session Description Protocol (SDP). These two protocols display multicast session names and correlate the names with multicast traffic.

SDP is a session directory protocol that is used for multimedia sessions. It helps advertise multimedia conference sessions and communicates setup information to participants who want to join the session. SDP simply formats the session description. It does not incorporate a transport protocol. A client commonly uses SDP to announce a conference session by periodically multicasting an announcement packet to a well-known multicast address and port using SAP.

SAP is a session directory announcement protocol that SDP uses as its transport protocol.

For information about supported standards for SAP and SDP, see "Supported IP Multicast Protocol Standards" on page 19.

Configuring the Session Announcement Protocol

The SAP and SDP protocols associate multicast session names with multicast traffic addresses. Only SAP has configuration parameters that users can change. Enabling SAP allows the router to receive announcements about multimedia and other multicast sessions.

Junos OS supports the following SAP and SDP standards:

- RFC 2327, *SDP Session Description Protocol*

- RFC 2974, *Session Announcement Protocol*

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.
3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its own RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.

To enable SAP and the receipt of session announcements, include the **sap** statement:

```
sap {
  disable;
  listen address <port port>;
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**

By default, SAP listens to the address and port 224.2.127.254:9875 for session advertisements. To add other addresses or pairs of address and port, include one or more **listen** statements.

Sessions established by SDP, SAP's higher-layer protocol, time out after 60 minutes.

To monitor the operation, use the show sap listen command.

See Also
- show sap listen on page 1775

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From the CLI, enter the **show sap listen** command.

Sample Output

```
user@host> show sap listen
```

```
Group Address   Port
224.2.127.254   9875
```

Meaning The output shows a list of the group addresses and ports that SAP and SDP listen on. Verify the following information:

- Each group address configured, especially the default **224.2.127.254**, is listed.
- Each port configured, especially the default **9875**, is listed.

CHAPTER 18

Facilitating Multicast Delivery Across Unicast-Only Networks with AMT

Example: Configuring Automatic IP Multicast Without Explicit Tunnels

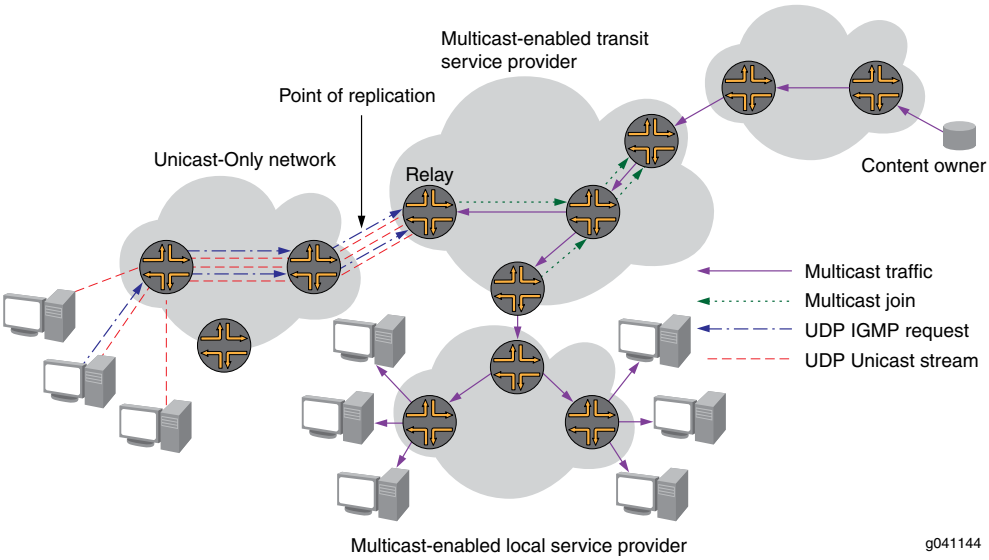
Understanding AMT

Automatic Multicast Tunneling (AMT) facilitates dynamic multicast connectivity between multicast-enabled networks across islands of unicast-only networks. Such connectivity enables service providers, content providers, and their customers to participate in delivering multicast traffic even if they lack end-to-end multicast connectivity.

AMT is supported on MX Series Ethernet Services Routers with Modular Port Concentrators (MPCs) that are running Junos 13.2 or later. AMT is also supported on i-chip based MPCs. AMT supports graceful restart (GR) but does not support graceful Routing Engine switchover (GRES).

AMT dynamically establishes unicast-encapsulated tunnels between well-known multicast-enabled relay points (AMT relays) and network points reachable only through unicast (AMT gateways). Figure 69 on page 420 shows the Automatic Multicast Tunneling Connectivity.

Figure 69: Automatic Multicast Tunneling Connectivity



The AMT protocol provides discovery and handshaking between relays and gateways to establish tunnels dynamically without requiring explicit per-tunnel configuration.

AMT relays are typically routers with native IP multicast connectivity that aggregate a potentially large number of AMT tunnels.

The Junos OS implementation supports the following AMT relay functions:

- IPv4 multicast traffic and IPv4 encapsulation
- Well-known sources located on the multicast network
- Prevention of denial-of-service attacks by quickly discarding multicast packets that are sourced through a gateway.
- Per-route replication to the full fan-out of all AMT tunnels desired
- The ability to collect normal interface statistics on AMT tunnels

Multicast sources located behind AMT gateways are not supported."Example: Configuring the AMT Protocol" on page 428"Example: Configuring the AMT Protocol" on page 428

AMT supports PIM sparse mode. AMT does not support dense mode operation.

See Also
- AMT Applications on page 420

AMT Applications

Transit service providers have a challenge in the Internet because many local service providers are not multicast-enabled. The challenge is how to entice content owners to transmit video and other multicast traffic across their backbones. The cost model for the content owners might be prohibitively high if they have to pay for unicast streams for the majority of their subscribers.

Until more local providers are multicast-enabled, there is a transition strategy proposed by the Internet Engineering Task Force (IETF) and implemented in open source software. This strategy is called Automatic IP Multicast Without Explicit Tunnels (AMT). AMT involves setting up relays at peering points in multicast networks that can be reached from gateways installed on hosts connected to unicast networks.

Without AMT, when a user who is connected to a unicast-only network wants to receive multicast content, the content owner can allow the user to join through unicast. However, the content owner incurs an added cost because the owner needs extra bandwidth to support the unicast subscribers.

AMT allows any host to receive multicast. On the client end is an AMT gateway that is a single host. Once the gateway has located an AMT relay, which might be a host but is more typically a router, the gateway periodically sends Internet Group Management Protocol (IGMP) messages over a dynamically created UDP tunnel to the relay. AMT relays and gateways cooperate to transmit multicast traffic sourced within the multicast network to end-user sites. AMT relays receive the traffic natively and unicast-encapsulate it to gateways. This allows anyone on the Internet to create a dynamic tunnel to download multicast data streams.

With AMT, a multicast-enabled service provider can offer multicast services to a content owner. When a customer of the unicast-only local provider wants to receive the content and subscribes using an AMT join, the multicast-enabled transit provider can then efficiently transport the content to the unicast-only local provider, which sends it on to the end user.

AMT is an excellent way for transit service providers (who can get access to the content, but do not have many end users) to provide multicast service to content owners, where it would not otherwise be economically feasible. It is also a useful transition strategy for local service providers who do not yet have multicast support on all downstream equipment.

AMT is also useful for connecting two multicast-enabled service providers that are separated by a unicast-only service provider.

Similarly, AMT can be used by local service providers whose networks are multicast-enabled to tunnel multicast traffic over legacy edge devices such as digital subscriber line access multiplexers (DSLAMs) that have limited multicast capabilities.

Technical details of the implementation of AMT are as follows:

- A three-way handshake is used to join groups from unicast receivers to prevent spoofing and denial-of-service (DoS) attacks.
- An AMT relay acting as a replication server joins the multicast group and translates multicast traffic into multiple unicast streams.
- The discovery mechanism uses anycast, enabling the discovery of the relay that is closest to the gateway in the network topology.

- An AMT gateway acting as a client is a host that joins the multicast group.
- Tunnel count limits on relays can limit bandwidth usage and avoid degradation of service.

AMT is described in detail in Internet draft draft-ietf-mboned-auto-multicast-10.txt, *Automatic IP Multicast Without Explicit Tunnels (AMT)*.

See Also
- Example: Configuring the AMT Protocol on page 428

AMT Operation

AMT is used to create multicast tunnels dynamically between multicast-enabled networks across islands of unicast-only networks. To do this, several steps occur sequentially.

1. The AMT relay (typically a router) advertises an anycast address prefix and route into the unicast routing infrastructure.
2. The AMT gateway (a host) sends AMT relay discovery messages to the nearest AMT relay reachable across the unicast-only infrastructure. To reduce the possibility of replay attacks or dictionary attacks, the relay discovery messages contain a cryptographic nonce. A cryptographic nonce is a random number used only once.
3. The closest relay in the topology receives the AMT relay discovery message and returns the nonce from the discovery message in an AMT relay advertisement message. This enables the gateway to learn the relay's unique IP address. The AMT relay now has an address to use for all subsequent (S,G), entries it will join.
4. The AMT gateway sends an AMT request message to the AMT relay's unique IP address to begin the process of joining the (S,G).
5. The AMT relay sends an AMT membership query back to the gateway.
6. The AMT gateway receives the AMT query message and sends an AMT membership update message containing the IGMP join messages.
7. The AMT relay sends a join message toward the source to build a native multicast tree in the native multicast infrastructure.
8. As packets are received from the source, the AMT relay replicates the packets to all interfaces in the outgoing interface list, including the AMT tunnel. The multicast traffic is then encapsulated in unicast AMT multicast data messages.
9. To maintain state in the AMT relay, the AMT gateway sends periodic AMT membership updates.
10. After the tunnel is established, the AMT tunnel state is refreshed with each membership update message sent. The timeout for the refresh messages is 240 seconds.
11. When the AMT gateway leaves the group, the AMT relay can free resources associated with the tunnel.

Note the following operational details:

- The AMT relay creates an AMT pseudo interface (tunnel interface). AMT tunnel interfaces are implemented as generic UDP encapsulation (**ud**) logical interfaces. These logical interfaces have the identifier format **ud-fpc/pic/port.unit**.
- All multicast packets (data and control) are encapsulated in unicast packets. UDP encapsulation is used for all AMT control and data packets using the IANA reserved UDP port number (2268) for AMT.
- The AMT relay maintains a receiver list for each multicast session. The relay maintains the multicast state for each gateway that has joined a particular group or (S,G) pair.

See Also
- AMT Applications on page 420
- Example: Configuring the AMT Protocol on page 428

Configuring the AMT Protocol

To configure the AMT protocol, include the **amt** statement:

```
amt {
  relay {
    accounting;
    family {
      inet {
        anycast-prefix ip-prefix</prefix-length>;
        local-address ip-address;
      }
    }
    secret-key-timeout minutes;
    tunnel-limit number;
  }
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**
- **[edit routing-instances *routing-instance-name* protocols]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols]**

NOTE: In the following example, only the **[edit protocols]** hierarchy is identified.

The minimum configuration to enable AMT is to specify the AMT local address and the AMT anycast prefix.

1. To enable the MX Series router to create the UDP encapsulation (**ud**) logical interfaces, include the **bandwidth** statement and specify the bandwidth in gigabits per second.

   ```
   [edit chassis fpc 0 pic 1]
   user@host# set tunnel-services bandwidth 1g
   ```

2. Specify the local address by including the **local-address** statement at the **[edit protocols amt relay family inet]** hierarchy level.

   ```
   [edit protocols amt relay family inet]
   user@host# set local-address 192.168.7.1
   ```

   The local address is used as the IP source of AMT control messages and the source of AMT data tunnel encapsulation. The local address can be configured on any active interface. Typically, the IP address of the router's **lo0.0** loopback interface is used for configuring the AMT local address in the default routing instance, and the IP address of the router's **lo0.*n*** loopback interface is used for configuring the AMT local address in VPN routing instances.

3. Specify the AMT anycast address by including the **anycast-prefix** statement at the **[edit protocols amt relay family inet]** hierarchy level.

   ```
   [edit protocols amt relay family inet]
   user@host# set anycast-prefix 192.168.0.0/16
   ```

   The AMT anycast prefix is advertised by unicast routing protocols to route AMT discovery messages to the router from nearby AMT gateways. Typically, the router's **lo0.0** interface loopback address is used for configuring the AMT anycast prefix in the default routing instance, and the router's **lo0.*n*** loopback address is used for configuring the AMT anycast prefix in VPN routing instances. However, the anycast address can be either the primary or secondary **lo0.0** loopback address.

   Ensure that your unicast routing protocol advertises the AMT anycast prefix in the route advertisements. If the AMT anycast prefix is advertised by BGP, ensure that the local autonomous system (AS) number for the AMT relay router is in the AS path leading to the AMT anycast prefix.

4. (Optional) Enable AMT accounting.

   ```
   [edit protocols amt relay]
   user@host# set accounting
   ```

5. (Optional) Specify the AMT secret key timeout by including the **secret-key-timeout** statement at the **[edit protocols amt relay]** hierarchy level. In the following example, the secret key timeout is configured to be 120 minutes.

```
[edit protocols amt relay]
user@host# set secret-key-timeout 120
```

The secret key is used to generate the AMT Message Authentication Code (MAC). Setting the secret key timeout shorter might improve security, but it consumes more CPU resources. The default is 60 minutes.

6. (Optional) Specify an AMT tunnel device by including the **tunnel-devices** statement at the **[edit protocols amt relay]** hierarchy level.

```
[edit protocols amt relay]
user@host# set tunnel-device 1
```

7. (Optional) Specify an AMT tunnel limit by including the **tunnel-limit** statement at the **[edit protocols amt relay]** hierarchy level. In the following example, the AMT tunnel limit is 12.

```
[edit protocols amt relay]
user@host# set tunnel-limit 12
```

The tunnel limit configures the static upper limit to the number of AMT tunnels that can be established. When the limit is reached, new AMT relay discovery messages are ignored.

8. Trace AMT protocol traffic by specifying options to the **traceoptions** statement at the **[edit protocols amt]** hierarchy level. Options applied at the AMT protocol level trace only AMT traffic. In the following example, all AMT packets are logged to the file **amt-log**.

```
[edit protocols amt]
user@host# set traceoptions file amt-log
user@host# set traceoptions flag packets
```

NOTE: For AMT operation, configure the PIM rendezvous point address as the primary loopback address of the AMT relay.

See Also

- AMT Applications on page 420
- Example: Configuring the AMT Protocol on page 428
- **mtrace** in the CLI Explorer

Configuring Default IGMP Parameters for AMT Interfaces

You can optionally configure default IGMP parameters for all AMT tunnel interfaces. Although, typically you do not need to change the values. To configure default IGMP attributes of all AMT relay tunnels, include the **amt** statement:

```
amt {
  relay {
    defaults {
```

```
        (accounting | no-accounting);
        group-policy [ policy-names ];
        query-interval seconds;
        query-response-interval seconds;
        robust-count number;
        ssm-map ssm-map-name;
        version version;
      }
    }
  }
```

You can include this statement at the following hierarchy levels:

- **[edit protocols igmp]**
- **[edit logical-systems *logical-system-name* protocols igmp]**
- **[edit routing-instances *routing-instance-name* protocols igmp]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp]**

The IGMP statements included at the **[edit protocols igmp amt relay defaults]** hierarchy level have the same syntax and purpose as IGMP statements included at the **[edit protocols igmp]** or **[edit protocols igmp interface *interface-name*]** hierarchy levels. These statements are as follows:

- You can collect IGMP join and leave event statistics. To enable the collection of IGMP join and leave event statistics for all AMT interfaces, include the **accounting** statement:

  ```
  user@host# set protocols igmp amt relay defaults accounting
  ```

- After enabling IGMP accounting, you must configure the router to filter the recorded information to a file or display it to a terminal. You can archive the events file.
- To disable the collection of IGMP join and leave event statistics for all AMT interfaces, include the **no-accounting** statement:

  ```
  user@host# set protocols igmp amt relay defaults no-accounting
  ```

- You can filter unwanted IGMP reports at the interface level. To filter unwanted IGMP reports, define a policy to match only IGMP group addresses (for IGMPv2) by using the policy's **route-filter** statement to match the group address. Define the policy to match IGMP (S,G) addresses (for IGMPv3) by using the policy's **route-filter** statement to match the group address and the policy's **source-address-filter** statement to match the source address. In the following example, the **amt_reject** policy is created to match both the group and source addresses.

  ```
  user@host# set policy-options policy-statement amt_reject from route-filter 224.1.1.1/32
    exact
  user@host# set policy-options policy-statement amt_reject from source-address-filter
    192.168.0.0/16 orlonger
  user@host# set policy-options policy-statement amt_reject then reject
  ```

- To apply the IGMP report filtering on the interface where you prefer not to receive specific group or (S,G) reports, include the **group-policy** statement. The following example applies the **amt_reject** policy to all AMT interfaces.

```
user@host# set protocols igmp amt relay defaults group-policy amt_reject
```

- You can change the IGMP query interval for all AMT interfaces to reduce or increase the number of host query messages sent. In AMT, host query messages are sent in response to membership request messages from the gateway. The query interval configured on the relay must be compatible with the membership request timer configured on the gateway. To modify this interval, include the **query-interval** statement. The following example sets the host query interval to 250 seconds.

  ```
  user@host# set protocols igmp amt relay defaults query-interval 250
  ```

  The IGMP querier router periodically sends general host-query messages. These messages solicit group membership information and are sent to the all-systems multicast group address, 224.0.0.1.

- You can change the IGMP query response interval. The query response interval multiplied by the robust count is the maximum amount of time that can elapse between the sending of a host query message by the querier router and the receipt of a response from a host. Varying this interval allows you to adjust the number of IGMP messages on the AMT interfaces. To modify this interval, include the **query-response-interval** statement. The following example configures the query response interval to 20 seconds.

  ```
  user@host# set protocols igmp amt relay defaults query-response-interval 20
  ```

- You can change the IGMP robust count. The robust count is used to adjust for the expected packet loss on the AMT interfaces. Increasing the robust count allows for more packet loss but increases the leave latency of the subnetwork. To modify the robust count, include the **robust-count** statement. The following example configures the robust count to **3**.

  ```
  user@host# set protocols igmp amt relay defaults robust-count 3
  ```

  The robust count automatically changes certain IGMP message intervals for IGMPv2 and IGMPv3.

  - On a shared network running IGMPv2, when the query router receives an IGMP leave message, it must send an IGMP group query message for a specified number of times. The number of IGMP group query messages sent is determined by the robust count. The interval between query messages is determined by the last member query interval. Also, the IGMPv2 query response interval is multiplied by the robust count to determine the maximum amount of time between the sending of a host query message and receipt of a response from a host.

    For more information about the IGMPv2 robust count, see RFC 2236, *Internet Group Management Protocol, Version 2*.

  - In IGMPv3 a change of interface state causes the system to immediately transmit a state-change report from that interface. If the state-change report is missed by one or more multicast routers, it is retransmitted. The number of times it is retransmitted is the robust count minus one. In IGMPv3 the robust count is also a factor in determining the group membership interval, the older version querier interval, and the other querier present interval.

    For more information about the IGMPv3 robust count, see RFC 3376, *Internet Group Management Protocol, Version 3*.

- You can apply a source-specific multicast (SSM) map to an AMT interface. SSM mapping translates IGMPv1 or IGMPv2 membership reports to an IGMPv3 report, which allows hosts running IGMPv1 or IGMPv2 to participate in SSM until the hosts transition to IGMPv3.

  SSM mapping applies to all group addresses that match the policy, not just those that conform to SSM addressing conventions (232/8 for IPv4).

  In this example, you create a policy to match the 232.1.1.1/32 group address for translation to IGMPv3. Then you define the SSM map that associates the policy with the 192.168.43.66 source address where these group addresses are found. Finally, you apply the SSM map to all AMT interfaces.

  ```
  user@host# set policy-options policy-statement ssm-policy-example term A from
    route-filter 232.1.1.1/32 exact
  user@host# set policy-options policy-statement ssm-policy-example term A then
    accept
  user@host# set routing-options multicast ssm-map ssm-map-example policy
    ssm-policy-example
  user@host# set routing-options multicast ssm-map ssm-map-example source
    192.168.43.66
  user@host# set protocols igmp amt relay defaults ssm-map ssm-map-example
  ```

See Also

- AMT Applications on page 420
- Example: Configuring the AMT Protocol on page 428
- *Specifying Log File Size, Number, and Archiving Properties* in the *Junos OS Administration Library*

Example: Configuring the AMT Protocol

This example shows how to configure the Automatic Multicast Tunneling (AMT) Protocol to facilitate dynamic multicast connectivity between multicast-enabled networks across islands of unicast-only networks.

Requirements

Before you begin:

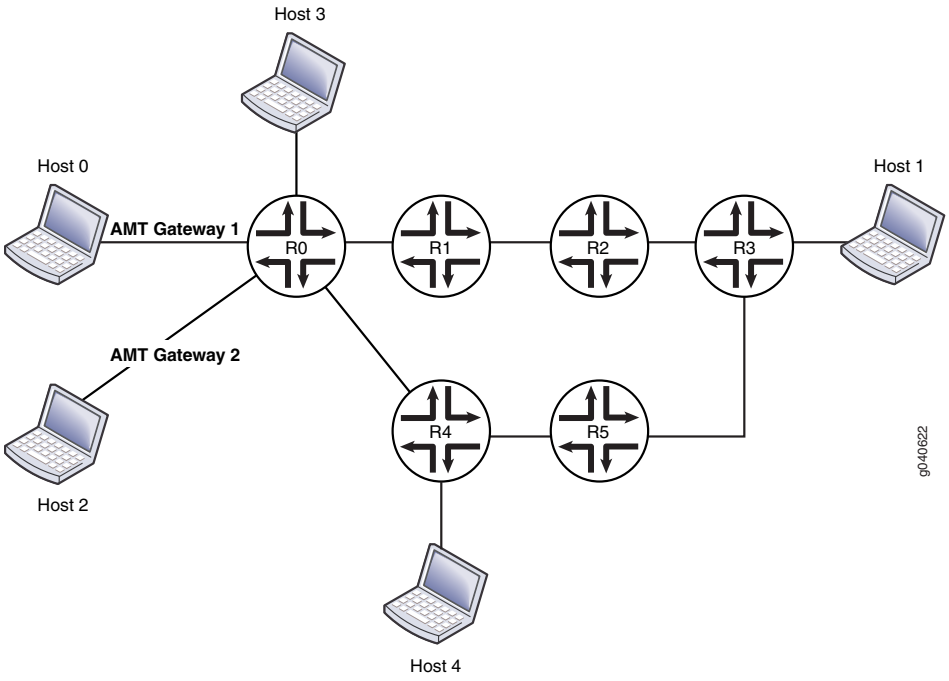
- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure a multicast group membership protocol (IGMP or MLD). See "Understanding IGMP" on page 25 and "Understanding MLD" on page 51.

Overview

In this example, Host 0 and Host 2 are multicast receivers in a unicast cloud. Their default gateway devices are AMT gateways. R0 and R4 are configured with unicast protocols only. R1, R2, R3, and R5 are configured with PIM multicast. Host 1 is a source in a multicast cloud. R0 and R5 are configured to perform AMT relay. Host 3 and Host 4 are multicast receivers (or sources that are directly connected to receivers). This example shows R1 configured with an AMT relay local address and an anycast prefix as its own loopback address. The example also shows R0 configured with tunnel services enabled.

Figure 70 on page 429 shows the topology used in this example.

Figure 70: AMT Gateway Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set protocols amt traceoptions file amt.log
set protocols amt traceoptions flag errors
set protocols amt traceoptions flag packets detail
set protocols amt traceoptions flag route detail
set protocols amt traceoptions flag state detail
set protocols amt traceoptions flag tunnels detail
set protocols amt relay family inet anycast-prefix 10.10.10.10/32
set protocols amt relay family inet local-address 10.255.112.201
set protocols amt relay tunnel-limit 10
```

```
set protocols pim interface all mode sparse-dense
set protocols pim interface all version 2
set protocols pim interface fxp0.0 disable
set chassis fpc 0 pic 0 tunnel-services bandwidth 1g
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the AMT protocol on R1:

1. Configure AMT tracing operations.

```
[edit protocols amt traceoptions]
user@host# set file amt.log
user@host# set flag errors
user@host# set flag packets detail
user@host# set flag route detail
user@host# set flag state detail
user@host# set flag tunnels detail
```

2. Configure the AMT relay settings.

```
[edit protocols amt relay]
user@host# set relay family inet anycast-prefix 10.10.10.10/32
user@host# set family inet local-address 10.255.112.201
user@host# set tunnel-limit 10
```

3. Configure PIM on R1's interfaces.

```
[edit protocols pim]
set interface all mode sparse-dense
set interface all version 2
set interface fxp0.0 disable
```

4. Enable tunnel functionality.

```
[edit chassis]
set fpc 0 pic 0 tunnel-services bandwidth 1g
```

5. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show chassis** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show chassis
fpc 0 {
```

```
    pic 0 {
      tunnel-services {
        bandwidth 1g;
      }
    }
  }
```

```
user@host# show protocols
amt {
  traceoptions {
    file amt.log;
    flag errors;
    flag packets detail;
    flag route detail;
    flag state detail;
    flag tunnels detail;
  }
  relay {
    family {
      inet {
        anycast-prefix 10.10.10.10/32;
        local-address 10.255.112.201;
      }
    }
    tunnel-limit 10;
  }
}
pim {
  interface all {
    mode sparse-dense;
    version 2;
  }
  interface fxp0.0 {
    disable;
  }
}
```

Verification

To verify the configuration, run the following commands:

- show amt statistics
- show amt summary
- show amt tunnel

See Also

- Configuring the AMT Protocol on page 423
- Configuring Default IGMP Parameters for AMT Interfaces on page 425
- AMT Applications on page 420

Related Documentation

- Understanding AMT on page 419

CHAPTER 19

Routing Content to Densely Clustered Receivers with DVMRP

Examples: Configuring DVMRP

Understanding DVMRP

Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

The Distance Vector Multicast Routing Protocol (DVMRP) is a distance-vector routing protocol that provides connectionless datagram delivery to a group of hosts across an internetwork. DVMRP is a distributed protocol that dynamically generates IP multicast delivery trees by using a technique called reverse-path multicasting (RPM) to forward multicast traffic to downstream interfaces. These mechanisms allow the formation of shortest-path trees, which are used to reach all group members from each network source of multicast traffic.

DVMRP is designed to be used as an interior gateway protocol (IGP) within a multicast domain.

Because not all IP routers support native multicast routing, DVMRP includes direct support for tunneling IP multicast datagrams through routers. The IP multicast datagrams are encapsulated in unicast IP packets and addressed to the routers that do support native multicast routing. DVMRP treats tunnel interfaces and physical network interfaces the same way.

DVMRP routers dynamically discover their neighbors by sending neighbor probe messages periodically to an IP multicast group address that is reserved for all DVMRP routers.

See Also
- Configuring DVMRP on page 434

Configuring DVMRP

Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Distance Vector Multicast Routing Protocol (DVMRP) is the first of the multicast routing protocols and has a number of limitations that make this method unattractive for large-scale Internet use. DVMRP is a dense-mode-only protocol, and uses the flood-and-prune or implicit join method to deliver traffic everywhere and then determine where the uninterested receivers are. DVMRP uses source-based distribution trees in the form (S,G).

To configure the Distance Vector Multicast Routing Protocol (DVMRP), include the **dvmrp** statement:

```
dvmrp {
  disable;
  export [ policy-names ];
  import [ policy-names ];
  interface interface-name {
    disable;
    hold-time seconds;
    metric metric;
    mode (forwarding | unicast-routing);
  }
  rib-group group-name;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

You can include this statement at the following hierarchy levels:

- **[edit protocols]**
- **[edit logical-systems *logical-system-name* protocols]**

By default, DVMRP is disabled.

See Also
- Example: Configuring DVMRP on page 434
- Example: Configuring DVMRP to Announce Unicast Routes on page 438
- Tracing DVMRP Protocol Traffic on page 442

Example: Configuring DVMRP

This example shows how to use DVMRP to announce routes used for multicast routing as well as multicast data forwarding.

Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.

Overview

DVMRP is a distance vector protocol for multicast. It is similar to RIP, in that both RIP and DVMRP have issues with scalability and robustness. PIM domains are more commonly used than DVMRP domains. In some environments, you might need to configure interoperability with DVMRP.

This example includes the following DVMRP settings:

- **protocols dvmrp rib-group**—Associates the **dvmrp-rib** routing table group with the DVMRP protocol to enable multicast RPF lookup.
- **protocols dvmrp interface**—Configures the DVMRP interface. The interface of a DVMRP router can be either a physical interface to a directly attached subnetwork or a tunnel interface to another multicast-capable area of the Multicast Backbone (*MBone*). The DVMRP hold-time period is the amount of time that a neighbor is to consider the sending router (this router) to be operative (up). The default hold-time period is 35 seconds.
- **protocols dvmrp interface hold-time**—The DVMRP hold-time period is the amount of time that a neighbor is to consider the sending router (this router) to be operative (up). The default hold-time period is 35 seconds.
- **protocols dvmrp interface metric**—All interfaces can be configured with a metric specifying cost for receiving packets on a given interface. The default metric is 1.

  For each source network reported, a route metric is associated with the unicast route being reported. The metric is the sum of the interface metrics between the router originating the report and the source network. A metric of 32 marks the source network as unreachable, thus limiting the breadth of the DVMRP network and placing an upper bound on the DVMRP convergence time.
- **routing-options rib-groups**—Enables DVMRP to access route information from the unicast routing table, **inet.0**, and from a separate routing table that is reserved for DVMRP. In this example, the first routing table group named **ifrg** contains local interface

routes. This ensures that local interface routes get added to both the **inet.0** table for use by unicast protocols and the **inet.2** table for multicast RPF check. The second routing table group named **dvmrp-rib** contains **inet.2** routes.

DVMRP needs to access route information from the unicast routing table, **inet.0**, and from a separate routing table that is reserved for DVMRP. You need to create the routing table for DVMRP and to create groups of routing tables so that the routing protocol process imports and exports routes properly. We recommend that you use routing table **inet.2** for DVMRP routing information.

- **routing-options interface-routes**— After defining the **ifrg** routing table group, use the **interface-routes** statement to insert interface routes into the **ifrg** group—in other words, into both **inet.0** and **inet.2**. By default, interface routes are imported into routing table **inet.0** only.
- **sap**—Enables the Session Directory Announcement Protocol (SAP) and the Session Directory Protocol (SDP). Enabling SAP allows the router to receive announcements about multimedia and other multicast sessions.

  SAP always listens to the address and port 224.2.127.254:9875 for session advertisements. To add other addresses or pairs of address and port, include one or more **listen** statements.

  Sessions learned by SDP, SAP's higher-layer protocol, time out after 60 minutes.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options interface-routes rib-group inet ifrg
set routing-options rib-groups ifrg import-rib inet.0
set routing-options rib-groups ifrg import-rib inet.2
set routing-options rib-groups dvmrp-rib export-rib inet.2
set routing-options rib-groups dvmrp-rib import-rib inet.2
set protocols sap
set protocols dvmrp rib-group dvmrp-rib
set protocols dvmrp interface ip-0/0/0.0 metric 5
set protocols dvmrp interface ip-0/0/0.0 hold-time 40
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an MSDP routing instance:

1. Create the routing tables for DVMRP routes.

```
[edit routing-options]
user@host# set interface-routes rib-group inet ifrg
user@host# set rib-groups ifrg import-rib [ inet.0 inet.2 ]
user@host# set rib-groups dvmrp-rib import-rib inet.2
```

```
user@host# set rib-groups dvmrp-rib export-rib inet.2
```

2. Configure SAP and SDP.

```
[edit protocols]
user@host# set sap
```

3. Enable DVMRP on the router and associate the **dvmrp-rib** routing table group with DVMRP to enable multicast RPF checks.

```
[edit protocols]
user@host# set dvmrp rib-group dvmrp-rib
```

4. Configure the DVMRP interface with a hold-time value and a metric. This example shows an IP-over-IP encapsulation tunnel interface.

```
[edit protocols]
user@host# set dvmrp interface ip–0/0/0.0
user@host# set dvmrp interface ip–0/0/0.0 hold-time 40
user@host# set dvmrp interface ip–0/0/0.0 metric 5
```

5. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show routing-options** command and the **show protocols** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-options
interface-routes {
   rib-group inet ifrg;
}
   rib-groups {
      ifrg {
         import-rib [ inet.0 inet.2 ];
      }
      dvmrp-rib {
         export-rib inet.2;
         import-rib inet.2;
      }
   }
user@host# show protocols
sap;
dvmrp {
   rib-group dvmrp-rib;
   interface ip-0/0/0.0 {
      metric 5;
      hold-time 40;
   }
```

}

Verification

To verify the configuration, run the following commands:

- show dvmrp interfaces
- show dvmrp neighbors

See Also

- Understanding DVMRP on page 433
- Example: Configuring DVMRP to Announce Unicast Routes on page 438

Example: Configuring DVMRP to Announce Unicast Routes

Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

This example shows how to use DVMRP to announce unicast routes used solely for multicast reverse-path forwarding (RPF) to set up the multicast control plane.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.

Overview

DVMRP has two modes. Forwarding mode is the default mode. In forwarding mode, DVMRP is responsible for the multicast control plane and multicast data forwarding. In the nondefault mode (which is shown in this example), DVMRP does not forward multicast data traffic. This mode is called unicast routing mode because in this mode DVMRP is only responsible for announcing unicast routes used for multicast RPF—in other words, for establishing the control plane. To forward multicast data, enable Protocol Independent Multicast (PIM) on the interface. If you have configured PIM on the interface, as shown in this example, you can configure DVMRP in unicast-routing mode only. You cannot configure PIM and DVMRP in forwarding mode at the same time.

This example includes the following settings:

- **policy-statement dvmrp-export**—Accepts static default routes.
- **protocols dvmrp export dvmrp-export**—Associates the **dvmrp-export** policy with the DVMRP protocol.

  All routing protocols use the routing table to store the routes that they learn and to determine which routes they advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in and retrieve from the routing table. Import and export policies are always from the point of view of the routing table. So the **dvmrp-export** policy exports static default routes from the routing table and accepts them into DVMRP.
- **protocols dvmrp interface all mode unicast-routing**—Enables all interfaces to announce unicast routes used solely for multicast RPF.
- **protocols dvmrp rib-group inet dvmrp-rg**—Associates the **dvmrp-rib** routing table group with the DVMRP protocol to enable multicast RPF checks.
- **protocols pim rib-group inet pim-rg**—Associates the **pim-rg** routing table group with the PIM protocol to enable multicast RPF checks.
- **routing-options rib inet.2 static route 0.0.0.0/0 discard**—Redistributes static routes to all DVMRP neighbors. The **inet.2** routing table stores unicast IPv4 routes for multicast RPF lookup. The **discard** statement silently drops packets without notice.
- **routing-options rib-groups dvmrp-rg import-rib inet.2**—Creates the routing table for DVMRP to ensure that the routing protocol process imports routes properly.
- **routing-options rib-groups dvmrp-rg export-rib inet.2**—Creates the routing table for DVMRP to ensure that the routing protocol process exports routes properly.
- **routing-options rib-groups pim-rg import-rib inet.2**—Enables access to route information from the routing table that stores unicast IPv4 routes for multicast RPF lookup. In this example, the first routing table group named **pim-rg** contains local interface routes. This ensures that local interface routes get added to the **inet.2** table.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set policy-options policy-statement dvmrp-export term 10 from protocol static
set policy-options policy-statement dvmrp-export term 10 from route-filter 0.0.0.0/0
  exact
set policy-options policy-statement dvmrp-export term 10 then accept
set protocols dvmrp rib-group inet
set protocols dvmrp rib-group dvmrp-rg
set protocols dvmrp export dvmrp-export
set protocols dvmrp interface all mode unicast-routing
set protocols dvmrp interface fxp0.0 disable
set protocols pim rib-group inet pim-rg
```

```
set protocols pim interface all
set routing-options rib inet.2 static route 0.0.0.0/0 discard
set routing-options rib-groups pim-rg import-rib inet.2
set routing-options rib-groups dvmrp-rg export-rib inet.2
set routing-options rib-groups dvmrp-rg import-rib inet.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an MSDP routing instance:

1. Configure the routing options.

```
[edit routing-options]
[edit routing -options]
user@host# set rib inet.2 static route 0.0.0.0/0 discard
user@host# set rib-groups pim-rg import-rib inet.2
user@host# set rib-groups dvmrp-rg import-rib inet.2
user@host# set rib-groups dvmrp-rg export-rib inet.2
```

2. Configure DVMRP.

```
[edit protocols]
user@host# set dvmrp rib-group inet dvmrp-rg
user@host# set dvmrp export dvmrp-export
user@host# set dvmrp interface all mode unicast-routing
user@host# set dvmrp interface fxp0 disable
```

3. Configure PIM so that PIM performs multicast data forwarding.

```
[edit protocols]
user@host# set pim rib-group inet pim-rg
user@host# set pim interface all
```

4. Configure the DVMRP routing policy.

```
[edit policy-options policy-statement dvmrp-export term 10]
user@host# set from protocol static
user@host# set from route-filter 0.0.0.0/0 exact
user@host# set then accept
```

5. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show policy-options** command, the **show protocols** command, and the **show routing-options** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement dvmrp-export {
   term 10 {
      from {
         protocol static;
         route-filter 0.0.0.0/0 exact;
      }
      then accept;
   }
}

user@host# show protocols
dvmrp {
   rib-group inet dvmrp-rg;
   export dvmrp-export;
   interface all {
      mode unicast-routing;
   }
   interface fxp0.0 {
      disable;
   }
}
pim {
   rib-group inet pim-rg;
   interface all;
}

user@host# show routing-options
rib inet.2 {
   static {
      route 0.0.0.0/0 discard;
   }
}
   rib-groups {
      pim-rg {
         import-rib inet.2;
      }
      dvmrp-rg {
         export-rib inet.2;
         import-rib inet.2;
      }
   }
```

Verification

To verify the configuration, run the following commands:

- show dvmrp interfaces
- show pim statistics

See Also

- Understanding DVMRP on page 433
- Example: Configuring DVMRP on page 434

Tracing DVMRP Protocol Traffic

Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Tracing operations record detailed messages about the operation of routing protocols, such as the various types of routing protocol packets sent and received, and routing policy actions. You can specify which trace operations are logged by including specific tracing flags. The following table describes the flags that you can include.

Flag	Description
all	Trace all operations.
general	Trace general flow.
graft	Trace graft messages.
neighbor	Trace neighbor probe packets.
normal	Trace normal events.
packets	Trace all DVMRP packets.
poison	Trace poison-route-reverse packets.
policy	Trace policy processing.
probe	Trace probe packets.
prune	Trace prune messages.
report	Trace membership report messages.
route	Trace routing information.
state	Trace state transitions.
task	Trace task processing.
timer	Trace timer processing.

In the following example, tracing is enabled for all routing protocol packets. Then tracing is narrowed to focus only on DVMRP packets of a particular type. To configure tracing operations for DVMRP:

1. (Optional) Configure tracing at the routing options level to trace all protocol packets.

   [edit routing-options traceoptions]

```
user@host# set file all-packets-trace
user@host# set flag all
```

2. Configure the filename for the DVMRP trace file.

```
[edit protocols dvmrp traceoptions]
user@host# set file dvmrp-trace
```

3. (Optional) Configure the maximum number of trace files.

```
[edit protocols dvmrp traceoptions]
user@host# set file files 5
```

4. (Optional) Configure the maximum size of each trace file.

```
[edit protocols dvmrp traceoptions]
user@host# set file size 1m
```

5. (Optional) Enable unrestricted file access.

```
[edit protocols dvmrp traceoptions]
user@host# set file world-readable
```

6. Configure tracing flags. Suppose you are troubleshooting issues with a particular DVMRP neighbor. The following example shows how to trace neighbor probe packets that match the neighbor's IP address.

```
[edit protocols dvmrp traceoptions]
user@host# set flag neighbor | match 192.168.1.1
```

7. View the trace file.

```
user@host> file list /var/log
user@host> file show /var/log/dvmrp-trace
```

See Also

- Understanding DVMRP on page 433
- *Tracing and Logging Junos OS Operations* in the *Junos OS Administration Library*

Release History Table

Release	Description
16.1	Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Related Documentation

- Understanding DVMRP on page 433

PART 5

Configuring Multicast VPNs

CHAPTER 20

Configuring Draft-Rosen Multicast VPNs

Draft-Rosen Multicast VPNs Overview

The Junos OS provides two types of draft-rosen multicast VPNs:

- Draft-rosen multicast VPNs with service provider tunnels operating in any-source multicast (ASM) mode (also referred to as *rosen 6* Layer 3 VPN multicast)—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on Section 2 of the IETF Internet draft **draft-rosen-vpn-mcast-06.txt**, *Multicast in MPLS/BGP VPNs* (expired April 2004).
- Draft-rosen multicast VPNs with service provider tunnels operating in source-specific multicast (SSM) mode (also referred to as *rosen 7* Layer 3 VPN multicast)—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on the IETF Internet draft **draft-rosen-vpn-mcast-07.txt**, *Multicast in MPLS/BGP IP VPNs*. Draft-rosen multicast VPNs with service provider tunnels operating in SSM mode do not require that the provider (P) routers maintain any VPN-specific Protocol-Independent Multicast (PIM) information.

NOTE: **Draft-rosen multicast VPNs are not supported in a logical system environment even though the configuration statements can be configured under the logical-systems hierarchy.**

In a draft-rosen Layer 3 multicast virtual private network (MVPN) configured with service provider tunnels, the VPN is multicast-enabled and configured to use the Protocol Independent Multicast (PIM) protocol within the VPN and within the service provider (SP) network. A multicast-enabled VPN routing and forwarding (VRF) instance

corresponds to a multicast domain (MD), and a PE router attached to a particular VRF instance is said to belong to the corresponding MD. For each MD there is a *default multicast distribution tree (MDT)* through the SP backbone, which connects all of the PE routers belonging to that MD. Any PE router configured with a default MDT group address can be the multicast source of one default MDT.

Draft-rosen MVPNs with service provider tunnels start by sending all multicast traffic over a default MDT, as described in section 2 of the IETF Internet draft **draft-rosen-vpn-mcast-06.txt** and section 7 of the IETF Internet draft **draft-rosen-vpn-mcast-07.txt**. This default mapping results in the delivery of packets to each provider edge (PE) router attached to the provider router even if the PE router has no receivers for the multicast group in that VPN. Each PE router processes the encapsulated VPN traffic even if the multicast packets are then discarded.

Related Documentation

- *Junos OS VPNs Library for Routing Devices*

Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs

Understanding Any-Source Multicast

Any-source multicast (ASM) is the form of multicast in which you can have multiple senders on the same group, as opposed to source-specific multicast where a single particular source is specified. The original multicast specification, RFC 1112, supports both the ASM many-to-many model and the SSM one-to-many model. For ASM, the (S,G) source, group pair is instead specified as (*,G), meaning that the multicast group traffic can be provided by multiple sources.

An ASM network must be able to determine the locations of all sources for a particular multicast group whenever there are interested listeners, no matter where the sources might be located in the network. In ASM, the key function of *source discovery* is a required function of the network itself.

In an environment where many sources come and go, such as for a videoconferencing service, ASM is appropriate. Multicast source discovery appears to be an easy process, but in sparse mode it is not. In dense mode, it is simple enough to flood traffic to every router in the network so that every router learns the source address of the content for that multicast group.

However, in PIM sparse mode, the flooding presents scalability and network resource use issues and is not a viable option.

See Also

- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316

- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512
- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 449

Example: Configuring Any-Source Multicast for Draft-Rosen VPNs

This example shows how to configure an any-source multicast VPN (MVPN) using dual PIM configuration with a customer RP and provider RP and mapping the multicast routes from customer to provider (known as *draft-rosen*). The Junos OS complies with RFC 4364 and Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP VPNs*.

- Requirements on page 449
- Overview on page 449
- Configuration on page 451
- Verification on page 458

Requirements

Before you begin:

- Configure the router interfaces. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure the VPN. See the *Junos OS VPNs Library for Routing Devices*.
- Configure the VPN import and VPN export policies. See *Configuring an Import Policy for the PE Router's VRF Table* in the *Junos OS VPNs Library for Routing Devices*.
- Make sure that the routing devices support multicast tunnel (**mt**) interfaces for encapsulating and de-encapsulating data packets into tunnels. See "Tunnel Services PICs and Multicast" on page 216 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 459.

  For multicast to work on draft-rosen Layer 3 VPNs, each of the following routers must have tunnel interfaces:

  - Each provider edge (PE) router.
  - Any provider (P) router acting as the RP.
  - Any customer edge (CE) router that is acting as a source's DR or as an RP. A receiver's designated router does not need a Tunnel Services PIC.

Overview

Draft-rosen multicast virtual private networks (MVPNs) can be configured to support service provider tunnels operating in any-source multicast (ASM) mode or source-specific multicast (SSM) mode.

In this example, the term *multicast Layer 3 VPNs* is used to refer to draft-rosen MVPNs.

This example includes the following settings.

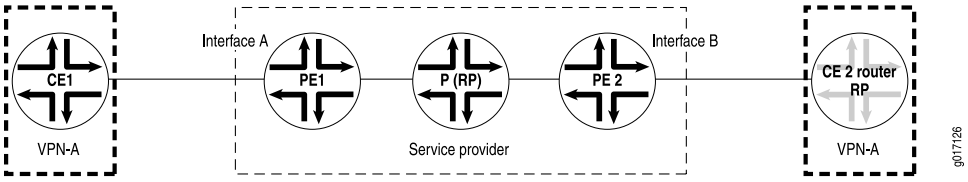
- **interface lo0.1**—Configures an additional unit on the loopback interface of the PE router. For the **lo0.1** interface, assign an address from the VPN address space. Add the **lo0.1** interface to the following places in the configuration:
  - VRF routing instance
  - PIM in the VRF routing instance
  - IGP and BGP policies to advertise the interface in the VPN address space

  In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector and the next hop is configured as **self**, Layer 3 multicast over VPN will not work, because PIM cannot transmit upstream interface information for multicast sources behind remote PEs into the network core. Multicast Layer 3 VPNs require that the BGP next-hop address of the VPN route match the BGP next-hop address of the loopback VRF instance address.
- **protocols pim interface**—Configures the interfaces between each provider router and the PE routers. On all CE routers, include this statement on the interfaces facing toward the provider router acting as the RP.
- **protocols pim mode sparse**—Enables PIM sparse mode on the **lo0** interface of all PE routers. You can either configure that specific interface or configure all interfaces with the **interface all** statement. On CE routers, you can configure sparse mode or sparse-dense mode.
- **protocols pim rp local**—On all routers acting as the RP, configure the address of the local **lo0** interface. The P router acts as the RP router in this example.
- **protocols pim rp static**—On all PE and CE routers, configure the address of the router acting as the RP.

  It is possible for a PE router to be configured as the VPN customer RP (C-RP) router. A PE router can also act as the DR. This type of PE configuration can simplify configuration of customer DRs and VPN C-RPs for multicast VPNs. This example does not discuss the use of the PE as the VPN C-RP.

  Figure 71 on page 450 shows multicast connectivity on the customer edge. In the figure, CE2 is the RP router. However, the RP router can be anywhere in the customer network.

Figure 71: Multicast Connectivity on the CE Routers



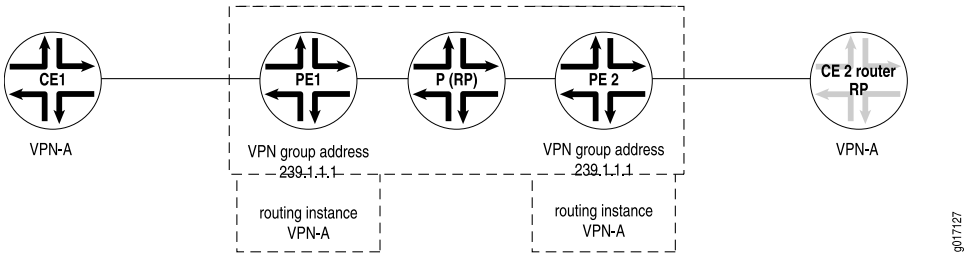
- **protocols pim version 2**—Enables PIM version 2 on the **lo0** interface of all PE routers and CE routers. You can either configure that specific interface or configure all interfaces with the **interface all** statement.

- **group-address**—In a routing instance, configure multicast connectivity for the VPN on the PE routers. Configure a VPN group address on the interfaces facing toward the router acting as the RP.

  The PIM configuration in the VPN routing and forwarding (VRF) instance on the PE routers needs to match the master PIM instance on the CE router. Therefore, the PE router contains both a master PIM instance (to communicate with the provider core) and the VRF instance (to communicate with the CE routers).

  VRF instances that are part of the same VPN share the same VPN group address. For example, all PE routers containing multicast-enabled routing instance VPN-A share the same VPN group address configuration. In Figure 72 on page 451, the shared VPN group address configuration is 239.1.1.1.

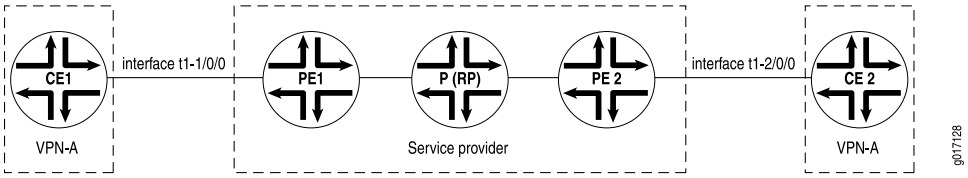
Figure 72: Multicast Connectivity for the VPN



- **routing-instances *instance-name* protocols pim rib-group**—Adds the routing group to the VPN's VRF instance.
- **routing-options rib-groups**—Configures the multicast routing group.

This example describes how to configure multicast in PIM sparse mode for a range of multicast addresses for VPN-A as shown in Figure 73 on page 451.

Figure 73: Customer Edge and Service Provider Networks



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

PE1

```
set interfaces lo0 unit 0 family inet address 192.168.27.13/32 primary
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 1 family inet address 10.10.47.101/32
set protocols pim rp static address 10.255.71.47
set protocols pim interface fxp0.0 disable
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

```
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface t1-1/0/0:0.0
set routing-instances VPN-A interface lo0.1
set routing-instances VPN-A route-distinguisher 10.255.71.46:100
set routing-instances VPN-A vrf-import VPNA-import
set routing-instances VPN-A vrf-export VPNA-export
set routing-instances VPN-A protocols ospf export bgp-to-ospf
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface t1-1/0/0:0.0
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface lo0.1
set routing-instances VPN-A protocols pim rib-group inet VPNA-mcast-rib
set routing-instances VPN-A protocols pim rp static address 10.255.245.91
set routing-instances VPN-A protocols pim interface t1-1/0/0:0.0 mode sparse
set routing-instances VPN-A protocols pim interface t1-1/0/0:0.0 version 2
set routing-instances VPN-A protocols pim interface lo0.1 mode sparse
set routing-instances VPN-A protocols pim interface lo0.1 version 2
set routing-instances VPN-A provider-tunnel pim-asm group-address 239.1.1.1
set routing-instances VPN-A protocols pim mvpn
set routing-options interface-routes rib-group inet VPNA-mcast-rib
set routing-options rib-groups VPNA-mcast-rib export-rib VPN-A.inet.2
set routing-options rib-groups VPNA-mcast-rib import-rib VPN-A.inet.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure multicast for draft-rosen VPNs:

1. Configure PIM on the P router.

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set dense-groups 224.0.1.39/32
[edit protocols pim]
user@host# set dense-groups 224.0.1.40/32
[edit protocols pim]
user@host# set rp local address 10.255.71.47
[edit protocols pim]
user@host# set interface all mode sparse
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
```

2. Configure PIM on the PE1 and PE2 routers. Specify a static route to the service provider RP—the P router (10.255.71.47).

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp static address 10.255.71.47
[edit protocols pim]
user@host#  set interface interface all mode sparse
[edit protocols pim]
user@host# set interface interface all version 2
```

```
[edit protocols pim]
user@host# set interface fxp0.0 disable
[edit protocols pim]
user@host# exit
```

3. Configure PIM on CE1. Specify the RP address for the VPN RP—Router CE2 (10.255.245.91).

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp static address 10.255.245.91
[edit protocols pim]
user@host# set interface all mode sparse
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
[edit protocols pim]
user@host# exit
```

4. Configure PIM on CE2, which acts as the VPN RP. Specify CE2's address (10.255.245.91).

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp local address 10.255.245.91
[edit protocols pim]
user@host# set interface all mode sparse
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
[edit protocols pim]
user@host# exit
```

5. On PE1, configure the routing instance (VPN-A) for the Layer 3 VPN.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set instance-type vrf
[edit routing-instances VPN-A]
user@host# set interface t1-1/0/0:0.0
[edit routing-instances VPN-A]
user@host# set interface lo0.1
[edit routing-instances VPN-A]
user@host# set route-distinguisher 10.255.71.46:100
[edit routing-instances VPN-A]
user@host# set vrf-import VPNA-import
[edit routing-instances VPN-A]
user@host# set vrf-export VPNA-export
```

6. On PE1, configure the IGP policy to advertise the interfaces in the VPN address space.

```
[edit routing-instances VPN-A]
user@host# set protocols ospf export bgp-to-ospf
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface t1-1/0/0:0.0
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface lo0.1
```

7. On PE1, set the RP configuration for the VRF instance. The RP configuration within the VRF instance provides explicit knowledge of the RP address, so that the (*,G) state can be forwarded.

```
[edit routing-instances VPN-A]
user@host# set protocols pim mvpn
[edit routing-instances VPN-A]
user@host# set protocols provider-tunnel pim-asm group-address 239.1.1.1
[edit routing-instances VPN-A]
user@host# set protocols pim rp static address 10.255.245.91
[edit routing-instances VPN-A]
user@host# set protocols pim interface t1-1/0/0:0.0 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface t1-1/0/0:0.0 version 2
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 version 2
[edit routing-instances VPN-A]
user@host# exit
```

8. On PE1, configure the loopback interfaces.

```
[edit]
user@host# edit interface lo0
[edit interface lo0]
user@host# set unit 0 family inet address 192.168.27.13/32 primary
[edit interface lo0]
user@host# set unit 0 family inet address 127.0.0.1/32
[edit interface lo0]
user@host# set unit 1 family inet address 10.10.47.101/32
[edit interface lo0]
user@host# exit
```

9. As you did for the PE1 router, configure the PE2 router.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set instance-type vrf
[edit routing-instances VPN-A]
user@host# set interface t1-2/0/0:0.0
[edit routing-instances VPN-A]
user@host# set interface lo0.1
[edit routing-instances VPN-A]
```

```
user@host# set route-distinguisher 10.255.71.51:100
[edit routing-instances VPN-A]
user@host# set vrf-import VPNA-import
[edit routing-instances VPN-A]
user@host# set vrf-export VPNA-export
[edit routing-instances VPN-A]
user@host# set protocols ospf export bgp-to-ospf
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface t1-2/0/0:0.0
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface lo0.1
[edit routing-instances VPN-A]
user@host# set protocols pim rp static address 10.255.245.91
[edit routing-instances VPN-A]
user@host# set protocols pim mvpn
[edit routing-instances VPN-A]
user@host# set protocols pim interface t1-2/0/0:0.0 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 version 2
[edit routing-instances VPN-A]
user@host# set provider-tunnel pim-asm group-address 239.1.1.1
user@host# exit
[edit]
user@host# edit interface lo0
[edit interface lo0]
user@host# set unit 0 family inet address 192.168.27.14/32 primary
[edit interface lo0]
user@host# set unit 0 family inet address 127.0.0.1/32
[edit interface lo0]
user@host# set unit 1 family inet address 10.10.47.102/32
```

10. When one of the PE routers is running Cisco Systems IOS software, you must configure the Juniper Networks PE router to support this multicast interoperability requirement. The Juniper Networks PE router must have the **lo0.0** interface in the master routing instance and the **lo0.1** interface assigned to the VPN routing instance. You must configure the **lo0.1** interface with the same IP address that the **lo0.0** interface uses for BGP peering in the provider core in the master routing instance.

    Configure the same IP address on the **lo0.0** and **lo0.1** loopback interfaces of the Juniper Networks PE router at the **[edit interfaces lo0]** hierarchy level, and assign the address used for BGP peering in the provider core in the master routing instance. In this alternate example, unit 0 and unit 1 are configured for Cisco IOS interoperability.

    ```
    [edit interface lo0]
    user@host# set unit 0 family inet address 192.168.27.14/32 primary
    [edit interface lo0]
    user@host# set unit 0 family inet address 127.0.0.1/32
    [edit interface lo0]
    user@host# set unit 1 family inet address 192.168.27.14/32
    [edit interface lo0]
    user@host# exit
    ```

11. Configure the multicast routing table group. This group accesses **inet.2** when doing RPF checks. However, if you are using **inet.0** for multicast RPF checks, this step will prevent your multicast configuration from working.

```
[edit]
user@host# edit routing-options
[edit routing-options]
user@host# set interface-routes rib-group inet VPNA-mcast-rib
[edit routing-options]
user@host# set rib-groups VPNA-mcast-rib export-rib VPN-A.inet.2
[edit routing-options]
user@host# set rib-groups VPNA-mcast-rib import-rib VPN-A.inet.2
[edit routing-options]
user@host# exit
```

12. Activate the multicast routing table group in the VPN's VRF instance.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set protocols pim rib-group inet VPNA-mcast-rib
```

13. If you are done configuring the device, commit the configuration.

```
[edit routing-instances VPN-A]
user@host# commit
```

Results

Confirm your configuration by entering the **show interfaces**, **show protocols**, **show routing-instances**, and **show routing-options** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration. This output shows the configuration on PE1.

```
user@host# show interfaces
lo0 {
    unit 0 {
        family inet {
            address 192.168.27.13/32 {
                primary;
            }
            address 127.0.0.1/32;
        }
    }
    unit 1 {
        family inet {
            address 10.10.47.101/32;
        }
    }
}

user@host# show protocols
pim {
    rp {
```

```
      static {
        address 10.255.71.47;
      }
    }
    interface fxp0.0 {
      disable;
    }
    interface all {
      mode sparse;
      version 2;
    }
}
```

```
user@host# show routing-instances
VPN-A {
    instance-type vrf;
    interface t1-1/0/0:0.0;
    interface lo0.1;
    route-distinguisher 10.255.71.46:100;
    vrf-import VPNA-import;
    vrf-export VPNA-export;
    provider-tunnel {
        pim-asm {
            group-address 239.1.1.1;
        }
    }
    protocols {
        ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
                interface t1-1/0/0:0.0;
                interface lo0.1;
            }
        }
        pim {
            mvpn;
            rib-group inet VPNA-mcast-rib;
            rp {
                static {
                    address 10.255.245.91;
                }
            }
            interface t1-1/0/0:0.0 {
                mode sparse;
                version 2;
            }
            interface lo0.1 {
                mode sparse;
                version 2;
            }
        }
    }
}
```

```
user@host# show routing-options
interface-routes {
    rib-group inet VPNA-mcast-rib;
```

```
}
rib-groups {
  VPNA-mcast-rib {
    export-rib VPN-A.inet.2;
    import-rib VPN-A.inet.2;
  }
}
```

Verification

To verify the configuration, run the following commands:

1. Display multicast tunnel information and the number of neighbors by using the show pim interfaces **instance** ***instance-name*** command from the PE1 or PE2 router. When issued from the PE1 router, the output display is:

   ```
   user@host> show pim interfaces instance VPN-A
   Instance: PIM.VPN-A
   Name                 Stat Mode       IP V State Count DR address
   lo0.1                Up   Sparse      4 2 DR        0 10.10.47.101
   mt-1/1/0.32769       Up   Sparse      4 2 DR        1
   mt-1/1/0.1081346       Up   Sparse      4 2 DR        0
   pe-1/1/0.32769       Up   Sparse      4 1 P2P       0
   t1-2/1/0:0.0         Up   Sparse      4 2 P2P       1
   ```

   You can also display all PE tunnel interfaces by using the show pim join command from the provider router acting as the RP.

2. Display multicast tunnel interface information, DR information, and the PIM neighbor status between VRF instances on the PE1 and PE2 routers by using the show pim neighbors **instance** ***instance-name*** command from either PE router. When issued from the PE1 router, the output is as follows:

   ```
   user@host> show pim neighbors instance VPN-A
   Instance: PIM.VPN-A
   Interface           IP V Mode        Option      Uptime Neighbor addr
   mt-1/1/0.32769       4 2             HPL       01:40:46 10.10.47.102
   t1-1/0/0:0.0         4 2             HPL       01:41:41 192.168.196.178
   ```

See Also

- Example: Configuring PIM RPF Selection on page 816

Load Balancing Multicast Tunnel Interfaces Among Available PICs

When you configure multicast on draft-rosen Layer 3 VPNs, multicast tunnel interfaces are automatically generated to encapsulate and de-encapsulate control and data traffic.

To generate multicast tunnel interfaces, a routing device must have one or more of the following tunnel-capable PICs:

- Adaptive Services PIC
- Multiservices PIC or Multiservices DPC
- Tunnel Services PIC
- On MX Series routers, a PIC created with the **tunnel-services** statement at the **[edit chassis fpc *slot-number* pic *number*]** hierarchy level

NOTE: A *routing device* is a router or an EX Series switch that is functioning as a router.

If a routing device has multiple such PICs, it might be important in your implementation to load balance the tunnel interfaces across the available tunnel-capable PICs.

The multicast tunnel interface that is used for encapsulation, **mt-[*xxxxx*]**, is in the range from 32,768 through 49,151. The interface **mt-[*yyyyy*]**, used for de-encapsulation, is in the range from 1,081,344 through 1,107,827. PIM runs only on the encapsulation interface. The de-encapsulation interface populates downstream interface information. For the default MDT, an instance's de-encapsulation and encapsulation interfaces are always created on the same PIC.

For each VPN, the PE routers build a multicast distribution tree within the service provider core network. After the tree is created, each PE router encapsulates all multicast traffic (data and control messages) from the attached VPN and sends the encapsulated traffic to the VPN group address. Because all the PE routers are members of the outgoing interface list in the multicast distribution tree for the VPN group address, they all receive the encapsulated traffic. When the PE routers receive the encapsulated traffic, they de-encapsulate the messages and send the data and control messages to the CE routers.

If a routing device has multiple tunnel-capable PICs (for example, two Tunnel Services PICs), the routing device load balances the creation of tunnel interfaces among the available PICs. However, in some cases (for example, after a reboot), a single PIC might be selected for all of the tunnel interfaces. This causes one PIC to have a heavy load, while other available PICs are underutilized. To prevent this, you can manually configure load balancing. Thus, you can configure and distribute the load uniformly across the available PICs.

The definition of a balanced state is determined by you and by the requirements of your Layer 3 VPN implementation. You might want all of the instances to be evenly distributed across the available PICs or across a configured list of PICs. You might want all of the encapsulation interfaces from all of the instances to be evenly distributed across the

available PICs or across a configured list of PICs. If the bandwidth of each tunnel encapsulation interface is considered, you might choose a different distribution. You can design your load-balancing configuration based on each instance or on each routing device.

NOTE: In a Layer 3 VPN, each of the following routing devices must have at least one tunnel-capable PIC:

- Each provider edge (PE) router.
- Any provider (P) router acting as the RP.
- Any customer edge (CE) router that is acting as a source's DR or as an RP. A receiver's designated router does not need a tunnel-capable PIC.

To configure load balancing:

1. On an M Series or T Series router or on an EX Series switch, install more than one tunnel-capable PIC. (In some implementations, only one PIC is required. Load balancing is based on the assumption that a routing device has more than one tunnel-capable PIC.)

2. On an MX Series router, configure more than one tunnel-capable PIC.

   ```
   [edit chassis fpc 0]
   user@host# set pic 0 tunnel-services bandwidth 10g
   user@host# set pic 1 tunnel-services bandwidth 10g
   ```

3. Configure Layer 3 VPNs as described in "Example: Configuring Any-Source Multicast for Draft-Rosen VPNs" on page 449.

   ```
   [edit routing-instances vpn1]
   user@host# set provider-tunnel pim-asm group-address 234.1.1.1
   user@host# set protocols pim rp static address 10.255.72.48
   user@host# set protocols pim interface fe-1/0/0.0
   user@host# set protocols pim interface lo0.1
   user@host# set protocols pim mvpn
   ```

4. For each VPN, specify a PIC list.

   ```
   [edit routing-instances vpn1 protocols pim]
   user@host# set tunnel-devices [ mt-1/1/0 mt-1/2/0 mt-2/0/0 ]
   ```

   The physical position of the PIC in the routing device determines the multicast tunnel interface name. For example, if you have an Adaptive Services PIC installed in FPC slot 0 and PIC slot 0, the corresponding multicast tunnel interface name is **mt-0/0/0**. The same is true for Tunnel Services PICs, Multiservices PICs, and Multiservices DPCs.

   In the **tunnel-devices** statement, the order of the PIC list that you specify does not impact how the interfaces are allocated. An instance uses all of the listed PICs to create default encapsulation and de-encapsulation interfaces, and data MDT encapsulation interfaces. The instance uses a round-robin approach to distributing

the tunnel interfaces (default and data MDT) across the PIC list (or across the available PICs, in the absence of a PIC list).

For the first tunnel, the round-robin algorithm starts with the lowest-numbered PIC. The second tunnel is created on the next-lowest-numbered PIC, and so on, round and round. The selection algorithm works routing device-wide. The round robin does not restart at the lowest-numbered PIC for each new instance. This applies to both the default and data MDT tunnel interfaces.

If one PIC in the list fails, new tunnel interfaces are created on the remaining PICs in the list using the round-robin algorithm. If all the PICs in the list go down, all tunnel interfaces are deleted and no new tunnel interfaces are created. If a PIC in the list comes up from the down state and the restored PIC is the only PIC that is up, the interfaces are reassigned to the restored PIC. If a PIC in the list comes up from the down state and other PICs are already up, an interface reassignment is not done. However, when a new tunnel interface needs to be created, the restored PIC is available for the selection process. If you include in the PIC list a PIC that is not installed on the routing device, the PIC is treated as if it is present but in the down state.

To balance the interfaces among the instances, you can assign one PIC to each instance. For example, if you have vpn1-10 and you have three PICs—for example, **mt-1/1/0**, **mt-1/2/0**, **mt-2/0/0**—you can configure vpn1-4 to only use **mt-1/1/0**, vpn5-7 to use **mt-1/2/0**, and vpn8-10 to use **mt-2/0/0**.

5. Commit the configuration.

   ```
   user@host# commit
   ```

   When you commit a new PIC list configuration, all the multicast tunnel interfaces for the routing instance are deleted and re-created using the new PIC list.

6. If you reboot the routing device, some PICs come up faster than others. The difference can be minutes. Therefore, when the tunnel interfaces are created, the known PIC list might not be the same as when the routing device is fully rebooted. This causes the tunnel interfaces to be created on some but not all available and configured PICs. To remedy this situation, you can manually rebalance the PIC load.

   Check to determine if a load rebalance is necessary.

   ```
   user@host> show interfaces terse | match mt-
   mt-1/1/0        up     up
   mt-1/1/0.32768  up     up      inet
   mt-1/1/0.1081344  up     up      inet
   mt-1/2/0        up     up
   mt-1/2/0.32769  up     up      inet
   mt-1/2/0.32770  up     up      inet
   mt-1/2/0.32771  up     up      inet
   ```

   The output shows that **mt-1/1/0** has only one tunnel encapsulation interface, while **mt-1/2/0** has three tunnel encapsulation interfaces. In a case like this, you might decide to rebalance the interfaces. As stated previously, encapsulation interfaces are in the range from 32,768 through 49,151. In determining whether a rebalance is necessary,

look at the encapsulation interfaces only, because the default MDT de-encapsulation interface always resides on the same PIC with the default MDT encapsulation interface.

7. (Optional) Rebalance the PIC load.

   user@host#> **request pim multicast-tunnel rebalance instance vpn1**

   This command re-creates and rebalances all tunnel interfaces for a specific instance.

   user@host#> **request pim multicast-tunnel rebalance**

   This command re-creates and rebalances all tunnel interfaces for all routing instances.

8. Verify that the PIC load is balanced.

```
user@host#> show interfaces terse | match mt-
mt-1/1/0        up     up
mt-1/1/0.32770  up     up      inet
mt-1/1/0.32768  up     up      inet
mt-1/1/0.1081344  up     up      inet
mt-1/2/0        up     up
mt-1/2/0.32769  up     up      inet
mt-1/2/0.32771  up     up      inet
```

   The output shows that **mt-1/1/0** has two encapsulation interfaces, and **mt-1/2/0** also has two encapsulation interfaces.

See Also

- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 449
- **request pim multicast-tunnel rebalance on page 1440** command in the CLI Explorer

Related Documentation

- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 489

Example: Configuring a Specific Tunnel for IPv4 Multicast VPN Traffic (Using Draft-Rosen MVPNs)

This example shows how to configure different provider tunnels to carry IPv4 customer traffic in a multicast VPN network.

Requirements

This example uses the following hardware and software components:

- Four Juniper Networks devices: Two PE routers and two CE devices.
- Junos OS Release 11.4 or later running on the PE routers.
- The PE routers can be M Series Multiservice Edge Routers, MX Series Ethernet Services Routers, or T Series Core Routers.
- The CE devices can be switches (such as EX Series Ethernet Switches), or they can be routers (such as M Series, MX Series, or T Series platforms).

Overview

A multicast tunnel is a mechanism to deliver control and data traffic across the provider core in a multicast VPN. Control and data packets are transmitted over the multicast distribution tree in the provider core. When a service provider carries both IPv4 and IPv6 traffic from a single customer, it is sometimes useful to separate the IPv4 and IPv6 traffic onto different multicast tunnels within the customer VRF routing instance. Putting customer IPv4 and IPv6 traffic on two different tunnels provides flexibility and control. For example, it helps the service provider to charge appropriately, to manage and measure traffic patterns, and to have an improved capability to make decisions when deploying new services.

A draft-rosen 7 multicast VPN control plane is configured in this example. The control plane is configured to use source-specific multicast (SSM) mode. The provider tunnel is used for the draft-rosen 7 control traffic and IPv4 customer traffic.

This example uses the following statements to configure the draft-rosen 7 control plane and specify IPv4 traffic to be carried in the provider tunnel:

- provider-tunnel pim-ssm **family inet group-address 232.1.1.1**
- **pim** mvpn **family inet** autodiscovery inet-mdt
- **pim mvpn family inet6 disable**
- **mvpn family inet** autodiscovery-only intra-as inclusive
- **family inet-mdt** signaling

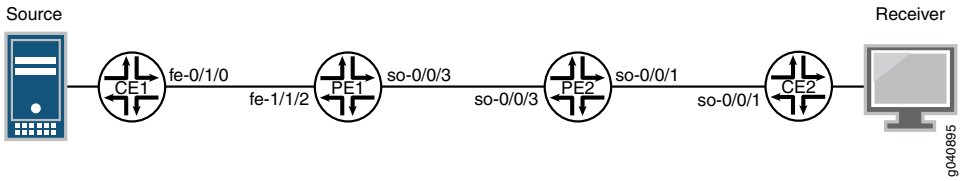
Note the following limitations:

- Junos OS does not currently support IPv6 with draft-rosen 6 or draft-rosen 7.
- Junos OS does not support more than two provider tunnels in a routing instance. For example, you cannot configure an RSVP-TE provider tunnel plus two MVPN provider tunnels.
- In a routing instance, you cannot configure both an any-source multicast (ASM) tunnel and an SSM tunnel.

Topology Diagram

Figure 74 on page 464 shows the topology used in this example.

Figure 74: Different Provider Tunnels for IPv4 Multicast VPN Traffic



PE Router Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router PE1

```
set interfaces so-0/0/3 unit 0 family inet address 10.111.10.1/30
set interfaces so-0/0/3 unit 0 family mpls
set interfaces fe-1/1/2 unit 0 family inet address 10.10.10.1/30
set interfaces lo0 unit 0 family inet address 10.255.182.133/32 primary
set interfaces lo0 unit 1 family inet address 10.10.47.100/32
set routing-options router-id 10.255.182.133
set routing-options route-distinguisher-id 10.255.182.133
set routing-options autonomous-system 100
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface fe-1/1/2.0
set routing-instances VPN-A interface lo0.1
set routing-instances VPN-A provider-tunnel pim-ssm family inet group-address 232.1.1.1
set routing-instances VPN-A provider-tunnel mdt threshold group 224.1.1.0/24 source
  10.240.0.242/32 rate 10
set routing-instances VPN-A provider-tunnel mdt tunnel-limit 20
set routing-instances VPN-A provider-tunnel mdt group-range 232.1.1.3/32
set routing-instances VPN-A vrf-target target:100:10
set routing-instances VPN-A vrf-table-label
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface all
set routing-instances VPN-A protocols ospf export bgp-to-ospf
set routing-instances VPN-A protocols pim mvpn family inet autodiscovery inet-mdt
set routing-instances VPN-A protocols pim mvpn family inet6 disable
set routing-instances VPN-A protocols pim rp static address 10.255.182.144
set routing-instances VPN-A protocols pim interface lo0.1 mode sparse-dense
set routing-instances VPN-A protocols pim interface fe-1/1/2.0 mode sparse-dense
set routing-instances VPN-A protocols mvpn family inet autodiscovery-only intra-as
  inclusive
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.182.133
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp family inet-mdt signaling
set protocols bgp group ibgp neighbor 10.255.182.142
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
```

```
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols pim rp local address 10.255.182.133
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim interface fxp0.0 disable
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf then accept
```

Router PE2

```
set interfaces so-0/0/1 unit 0 family inet address 10.10.20.1/30
set interfaces so-0/0/3 unit 0 family inet address 10.111.10.2/30
set interfaces so-0/0/3 unit 0 family iso
set interfaces so-0/0/3 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.182.142/32 primary
set interfaces lo0 unit 1 family inet address 10.10.47.101/32
set routing-options router-id 10.255.182.142
set routing-options route-distinguisher-id 10.255.182.142
set routing-options autonomous-system 100
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface so-0/0/1.0
set routing-instances VPN-A interface lo0.1
set routing-instances VPN-A provider-tunnel pim-ssm family inet group-address 232.1.1.1
set routing-instances VPN-A provider-tunnel mdt threshold group 224.1.1.0/24 source
  10.240.0.242/32 rate 10
set routing-instances VPN-A provider-tunnel mdt tunnel-limit 20
set routing-instances VPN-A provider-tunnel mdt group-range 232.1.1.3/32
set routing-instances VPN-A vrf-target target:100:10
set routing-instances VPN-A vrf-table-label
set routing-instances VPN-A routing-options graceful-restart
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface all
set routing-instances VPN-A protocols ospf export bgp-to-ospf
set routing-instances VPN-A protocols pim mvpn family inet autodiscovery inet-mdt
set routing-instances VPN-A protocols pim mvpn family inet6 disable
set routing-instances VPN-A protocols pim rp static address 10.255.182.144
set routing-instances VPN-A protocols pim interface lo0.1 mode sparse-dense
set routing-instances VPN-A protocols pim interface so-0/0/1.0 mode sparse-dense
set routing-instances VPN-A protocols mvpn family inet autodiscovery-only intra-as
  inclusive
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.182.142
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp family inet-mdt signaling
set protocols bgp group ibgp neighbor 10.255.182.133
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols pim rp static address 10.255.182.133
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim interface fxp0.0 disable
set policy-options policy-statement bgp-to-ospf from protocol bgp
set policy-options policy-statement bgp-to-ospf then accept
```

Router PE1

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the *CLI User Guide*.

To configure Router PE1:

1. Configure the router interfaces, enabling IPv4 traffic.

   Also enable MPLS on the interface facing Router PE2.

   The **lo0.1** interface is for the **VPN-A** routing instance.

   ```
   [edit interfaces]
   user@PE1# set so-0/0/3 unit 0 family inet address 10.111.10.1/30
   user@PE1# set so-0/0/3 unit 0 family mpls
   user@PE1# set fe-1/1/2 unit 0 family inet address 10.10.10.1/30
   user@PE1# set lo0 unit 0 family inet address 10.255.182.133/32 primary
   user@PE1# set lo0 unit 1 family inet address 10.10.47.100/32
   ```

2. Configure a routing policy to export BGP routes from the routing table into OSPF.

   ```
   [edit policy-options policy-statement bgp-to-ospf]
   user@PE1# set from protocol bgp
   user@PE1# set then accept
   ```

3. Configure the router ID, route distinguisher, and autonomous system number.

   ```
   [edit routing-options]
   user@PE1# set router-id 10.255.182.133
   user@PE1# set route-distinguisher-id 10.255.182.133
   user@PE1# set autonomous-system 100
   ```

4. Configure the protocols that need to run in the main routing instance to enable MPLS, BGP, the IGP, VPNs, and PIM sparse mode.

   ```
   [edit protocols ]
   user@PE1# set mpls interface all
   user@PE1# set mpls interface fxp0.0 disable
   user@PE1# set bgp group ibgp type internal
   user@PE1# set bgp group ibgp local-address 10.255.182.133
   user@PE1# set bgp group ibgp family inet-vpn unicast
   user@PE1# set bgp group ibgp neighbor 10.255.182.142
   user@PE1# set ospf traffic-engineering
   user@PE1# set ospf area 0.0.0.0 interface all
   user@PE1# set ospf area 0.0.0.0 interface fxp0.0 disable
   user@PE1# set ldp interface all
   user@PE1# set pim rp local address 10.255.182.133
   user@PE1# set pim interface all mode sparse
   user@PE1# set pim interface all version 2
   user@PE1# set pim interface fxp0.0 disable
   ```

5. Create the customer VRF routing instance.

```
[edit routing-instances VPN-A]
user@PE1# set instance-type vrf
user@PE1# set interface fe-1/1/2.0
user@PE1# set interface lo0.1
user@PE1# set vrf-target target:100:10
user@PE1# set vrf-table-label
user@PE1# set protocols ospf area 0.0.0.0 interface all
user@PE1# set protocols ospf export bgp-to-ospf
user@PE1# set protocols pim rp static address 10.255.182.144
user@PE1# set protocols pim interface lo0.1 mode sparse-dense
user@PE1# set protocols pim interface fe-1/1/2.0 mode sparse-dense
```

6. Configure the draft-rosen 7 control plane, and specify IPv4 traffic to be carried in the provider tunnel.

```
[edit routing-instances VPN-A]
user@PE1# set provider-tunnel pim-ssm family inet group-address 232.1.1.1
user@PE1# set protocols pim mvpn family inet autodiscovery inet-mdt
user@PE1# set protocols pim mvpn family inet6 disable
user@PE1# set protocols mvpn family inet autodiscovery-only intra-as inclusive

[edit protocols bgp group ibgp]
user@PE1# set family inet-mdt signaling
```

7. (Optional) Configure a data MDT tunnel.

```
[edit routing-instances VPN-A]
user@PE1# set provider-tunnel mdt threshold group 224.1.1.0/24 source
  10.240.0.242/32 rate 10
user@PE1# set provider-tunnel mdt tunnel-limit 20
user@PE1# set provider-tunnel mdt group-range 232.1.1.3/32
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
lo0 {
  unit 0 {
    family inet {
      address 10.255.182.133/32 {
        primary;
      }
    }
  }
  unit 1 {
    family inet {
      address 10.10.47.100/32;
```

```
    }
  }
}
so-0/0/3 {
  unit 0 {
    family inet {
      address 10.111.10.1/30;
    }
    family mpls;
  }
}
fe-1/1/2 {
  unit 0 {
    family inet {
      address 10.10.10.1/30;
    }
  }
}
```

```
user@PE1# show policy-options
policy-statement bgp-to-ospf {
  from protocol bgp;
  then accept;
}
```

```
user@PE1# show protocols
mpls {
  ipv6-tunneling;
  interface all;
  interface fxp0.0 {
    disable;
  }
}
bgp {
  group ibgp {
    type internal;
    local-address 10.255.182.133;
    family inet-vpn {
      unicast;
    }
    family inet-mdt {
      signaling;
    }
    neighbor 10.255.182.142;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
ldp {
  interface all;
```

```
}
pim {
    rp {
        local {
            address 10.255.182.133;
        }
    }
    interface all {
        mode sparse;
        version 2;
    }
    interface fxp0.0 {
        disable;
    }
}
```

```
user@PE1# show routing-instances
VPN-A {
    instance-type vrf;
    interface fe-1/1/2.0;
    interface lo0.1;
    provider-tunnel {
        pim-ssm {
            family {
                inet {
                    group-address 232.1.1.1;
                }
            }
        }
        mdt {
            threshold {
                group 224.1.1.0/24 {
                    source 10.240.0.242/32 {
                        rate 10;
                    }
                }
            }
            tunnel-limit 20;
            group-range 232.1.1.3/32;
        }
    }
    vrf-target target:100:10;
    vrf-table-label;
    protocols {
        ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
                interface all;
            }
        }
        pim {
            mvpn {
                family {
                    inet {
                        autodiscovery {
                            inet-mdt;
```

```
            }
          }
          inet6 {
            disable;
          }
        }
      }
      rp {
        static {
          address 10.255.182.144;
        }
      }
      interface lo0.1 {
        mode sparse-dense;
      }
      interface fe-1/1/2.0 {
        mode sparse-dense;
      }
    }
    mvpn {
      family {
        inet {
          autodiscovery-only {
            intra-as {
              inclusive;
            }
          }
        }
      }
    }
  }
}
```

```
user@PE1# show routing-options
route-distinguisher-id 10.255.182.133;
autonomous-system 100;
router-id 10.255.182.133;
```

If you are done configuring the router, enter **commit** from configuration mode.

Repeat the procedure for Router PE2, using the appropriate interface names and IP addresses.

CE Device Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device CE1

```
set interfaces fe-0/1/0 unit 0 family inet address 10.10.10.2/30
set interfaces lo0 unit 0 family inet address 10.255.182.144/32 primary
```

```
set routing-options router-id 10.255.182.144
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols pim rp local address 10.255.182.144
set protocols pim interface all mode sparse-dense
set protocols pim interface fxp0.0 disable
```

Device CE2

```
set interfaces so-0/0/1 unit 0 family inet address 10.10.20.2/30
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 0 family inet address 10.255.182.140/32 primary
set routing-options router-id 10.255.182.140
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols pim rp static address 10.255.182.144
set protocols pim interface all mode sparse-dense
set protocols pim interface fxp0.0 disable
```

Device CE1

Step-by-Step Procedure

To configure Device CE1:

1. Configure the router interfaces, enabling IPv4 and IPv6 traffic.

   ```
   [edit interfaces]
   user@CE1# set fe-0/1/0 unit 0 family inet address 10.10.10.2/30
   user@CE1# set lo0 unit 0 family inet address 10.255.182.144/32 primary
   ```

2. Configure the router ID.

   ```
   [edit routing-options]
   user@CE1# set router-id 10.255.182.144
   ```

3. Configure the protocols that need to run on the CE device to enable OSPF (for IPv4) and PIM sparse-dense mode.

   ```
   [edit protocols]
   user@CE1# set ospf area 0.0.0.0 interface all
   user@CE1# set ospf area 0.0.0.0 interface fxp0.0 disable
   user@CE1# set pim rp local address 10.255.182.144
   user@CE1# set pim interface all mode sparse-dense
   user@CE1# set pim interface fxp0.0 disable
   ```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@CE1# show interfaces
fe-0/1/0 {
    unit 0 {
```

```
        family inet {
          address 10.10.10.2/30;
        }
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.255.182.144/32 {
            primary;
          }
        }
      }
    }
```

```
user@CE1# show protocols
ospf {
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
pim {
  rp {
    local {
      address 10.255.182.144;
    }
  }
  interface all {
    mode sparse-dense;
  }
  interface fxp0.0 {
    disable;
  }
}
```

```
user@CE1# show routing-options
router-id 10.255.182.144;
```

If you are done configuring the router, enter **commit** from configuration mode.

Repeat the procedure for Device CE2, using the appropriate interface names and IP addresses.

Verification

Confirm that the configuration is working properly.

- Verifying Tunnel Encapsulation on page 473
- Verifying PIM Neighbors on page 473
- Verifying the Provider Tunnel and Control Plane on page 473

- Checking Routes on page 474
- Verifying MDT Tunnels on page 474

Verifying Tunnel Encapsulation

Purpose Verify that PIM multicast tunnel (**mt**) encapsulation and deencapsulation interfaces come up.

Action

```
user@PE1> show pim interfaces instance VPN-A
Instance: PIM.VPN-A

Name               Stat Mode       IP V State NbrCnt JoinCnt(sg) JoinCnt(*g) DR
address
fe-1/1/2.0         Up   SparseDense 4 2 NotDR      1           1            1
10.10.10.2
lo0.1              Up   SparseDense 4 2 DR         0           0            0
10.10.47.100
lsi.2304           Up   SparseDense 4 2 P2P        0           0            0
mt-0/3/0.32769     Up   SparseDense 4 2 P2P        0           0            0
mt-1/2/0.1081344   Up   SparseDense 4 2 P2P        0           0            0
mt-1/2/0.32768     Up   SparseDense 4 2 P2P        1           0            0
pe-0/3/0.32770     Up   Sparse      4 2 P2P        0           0            0
```

Meaning The multicast tunnel interface that is used for encapsulation, **mt-[*xxxxx*]**, is in the range from 32,768 through 49,151. The interface **mt-[*yyyyy*]**, used for de-encapsulation, is in the range from 1,081,344 through 1,107,827. PIM runs only on the encapsulation interface. The de-encapsulation interface populates downstream interface information.

Verifying PIM Neighbors

Purpose Verify that PIM neighborship is established over the multicast tunnel interface.

Action

```
user@PE1> show pim neighbors instance VPN-A
Instance: PIM.VPN-A
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking Bit

Interface           IP V Mode        Option      Uptime Neighbor addr
fe-1/1/2.0           4 2             HPLGT      00:29:35 10.10.10.2
mt-1/2/0.32768       4 2             HPLGT      00:28:32 10.10.47.101
```

Meaning When the neighbor address is listed and the uptime is incrementing, it means that PIM neighborship is established over the multicast tunnel interface.

Verifying the Provider Tunnel and Control Plane

Purpose Confirm that the provider tunnel and control-plane protocols are correct.

Action

```
user@PE1> show pim mvpn
Instance          Family          VPN-Group          Mode          Tunnel
PIM.VPN-A         INET            225.1.1.1          PIM-MVPN      PIM-SSM
```

Meaning For draft-rosen, the MVPN mode appears in the output as **PIM-MVPN**.

Checking Routes

Purpose Verify that traffic flows as expected.

Action

```
user@R1> show multicast route extensive instance VPN-A
Family: INET

Group: 224.1.1.1
    Source: 10.240.0.242/32
    Upstream interface: fe-1/1/2.0
    Downstream interface list:
        mt-1/2/0.32768
    Session description: NOB Cross media facilities
    Statistics: 92 kBps, 1001 pps, 1869820 packets
    Next-hop ID: 1048581
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
```

Meaning For draft-rosen, the upstream protocol appears in the output as **PIM**.

Verifying MDT Tunnels

Purpose Verify that both default and data MDT tunnels are correct.

Action

```
user@PE1> show pim mdt instance VPN-A
Instance: PIM.VPN-A
Tunnel direction: Outgoing
Tunnel mode: PIM-SSM
Default group address: 232.1.1.1
Default source address: 10.255.182.133
Default tunnel interface: mt-1/2/0.32769
Default tunnel source: 0.0.0.0

C-group address   C-source address   P-group address    Data tunnel interface
224.1.1.1         10.240.0.242       232.1.1.3          mt-0/3/0.32771

Instance: PIM.VPN-A
Tunnel direction: Incoming
Tunnel mode: PIM-SSM
Default group address: 232.1.1.1
Default source address: 10.255.182.142
Default tunnel interface: mt-1/2/0.1081345
Default tunnel source: 0.0.0.0
```

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502

Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs

- Understanding Any-Source Multicast on page 475
- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 476
- Load Balancing Multicast Tunnel Interfaces Among Available PICs on page 486

Understanding Any-Source Multicast

Any-source multicast (ASM) is the form of multicast in which you can have multiple senders on the same group, as opposed to source-specific multicast where a single particular source is specified. The original multicast specification, RFC 1112, supports both the ASM many-to-many model and the SSM one-to-many model. For ASM, the (S,G) source, group pair is instead specified as (*,G), meaning that the multicast group traffic can be provided by multiple sources.

An ASM network must be able to determine the locations of all sources for a particular multicast group whenever there are interested listeners, no matter where the sources might be located in the network. In ASM, the key function of *source discovery* is a required function of the network itself.

In an environment where many sources come and go, such as for a videoconferencing service, ASM is appropriate. Multicast source discovery appears to be an easy process, but in sparse mode it is not. In dense mode, it is simple enough to flood traffic to every

router in the network so that every router learns the source address of the content for that multicast group.

However, in PIM sparse mode, the flooding presents scalability and network resource use issues and is not a viable option.

See Also

- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512
- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 449

Example: Configuring Any-Source Multicast for Draft-Rosen VPNs

This example shows how to configure an any-source multicast VPN (MVPN) using dual PIM configuration with a customer RP and provider RP and mapping the multicast routes from customer to provider (known as *draft-rosen*). The Junos OS complies with RFC 4364 and Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP VPNs*.

Requirements

Before you begin:

- Configure the router interfaces. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure the VPN. See the *Junos OS VPNs Library for Routing Devices*.
- Configure the VPN import and VPN export policies. See *Configuring an Import Policy for the PE Router's VRF Table* in the *Junos OS VPNs Library for Routing Devices*.
- Make sure that the routing devices support multicast tunnel (**mt**) interfaces for encapsulating and de-encapsulating data packets into tunnels. See "Tunnel Services PICs and Multicast" on page 216 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 459.

  For multicast to work on draft-rosen Layer 3 VPNs, each of the following routers must have tunnel interfaces:

  - Each provider edge (PE) router.
  - Any provider (P) router acting as the RP.

- Any customer edge (CE) router that is acting as a source's DR or as an RP. A receiver's designated router does not need a Tunnel Services PIC.

Overview

Draft-rosen multicast virtual private networks (MVPNs) can be configured to support service provider tunnels operating in any-source multicast (ASM) mode or source-specific multicast (SSM) mode.

In this example, the term *multicast Layer 3 VPNs* is used to refer to draft-rosen MVPNs.

This example includes the following settings.

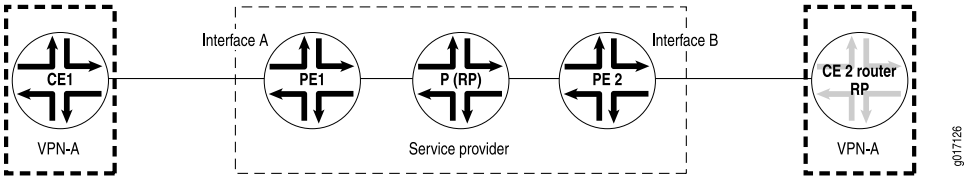
- **interface lo0.1**—Configures an additional unit on the loopback interface of the PE router. For the **lo0.1** interface, assign an address from the VPN address space. Add the **lo0.1** interface to the following places in the configuration:
  - VRF routing instance
  - PIM in the VRF routing instance
  - IGP and BGP policies to advertise the interface in the VPN address space

  In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector and the next hop is configured as **self**, Layer 3 multicast over VPN will not work, because PIM cannot transmit upstream interface information for multicast sources behind remote PEs into the network core. Multicast Layer 3 VPNs require that the BGP next-hop address of the VPN route match the BGP next-hop address of the loopback VRF instance address.
- **protocols pim interface**—Configures the interfaces between each provider router and the PE routers. On all CE routers, include this statement on the interfaces facing toward the provider router acting as the RP.
- **protocols pim mode sparse**—Enables PIM sparse mode on the **lo0** interface of all PE routers. You can either configure that specific interface or configure all interfaces with the **interface all** statement. On CE routers, you can configure sparse mode or sparse-dense mode.
- **protocols pim rp local**—On all routers acting as the RP, configure the address of the local **lo0** interface. The P router acts as the RP router in this example.
- **protocols pim rp static**—On all PE and CE routers, configure the address of the router acting as the RP.

  It is possible for a PE router to be configured as the VPN customer RP (C-RP) router. A PE router can also act as the DR. This type of PE configuration can simplify configuration of customer DRs and VPN C-RPs for multicast VPNs. This example does not discuss the use of the PE as the VPN C-RP.

  Figure 71 on page 450 shows multicast connectivity on the customer edge. In the figure, CE2 is the RP router. However, the RP router can be anywhere in the customer network.

Figure 75: Multicast Connectivity on the CE Routers

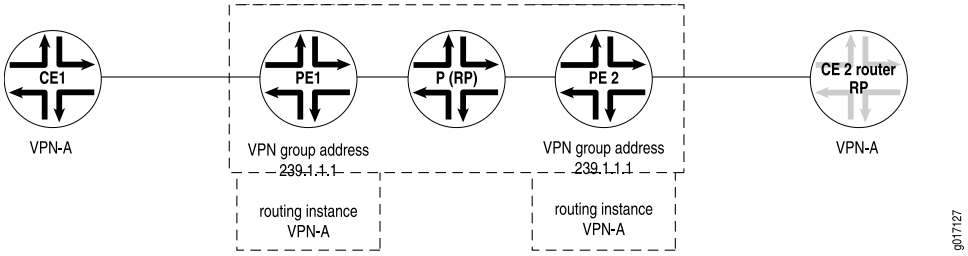


- **protocols pim version 2**—Enables PIM version 2 on the **lo0** interface of all PE routers and CE routers. You can either configure that specific interface or configure all interfaces with the **interface all** statement.
- **group-address**—In a routing instance, configure multicast connectivity for the VPN on the PE routers. Configure a VPN group address on the interfaces facing toward the router acting as the RP.

  The PIM configuration in the VPN routing and forwarding (VRF) instance on the PE routers needs to match the master PIM instance on the CE router. Therefore, the PE router contains both a master PIM instance (to communicate with the provider core) and the VRF instance (to communicate with the CE routers).

  VRF instances that are part of the same VPN share the same VPN group address. For example, all PE routers containing multicast-enabled routing instance VPN-A share the same VPN group address configuration. In Figure 72 on page 451, the shared VPN group address configuration is 239.1.1.1.

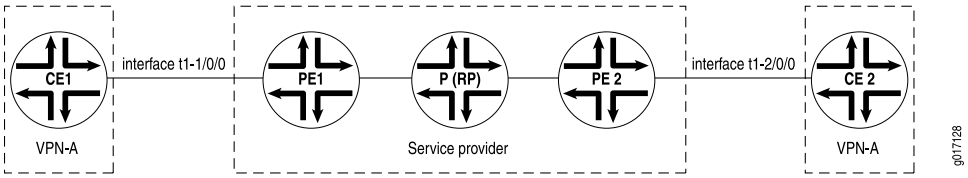
Figure 76: Multicast Connectivity for the VPN



- **routing-instances** ***instance-name*** **protocols pim rib-group**—Adds the routing group to the VPN's VRF instance.
- **routing-options rib-groups**—Configures the multicast routing group.

This example describes how to configure multicast in PIM sparse mode for a range of multicast addresses for VPN-A as shown in Figure 73 on page 451.

Figure 77: Customer Edge and Service Provider Networks



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

PE1

```
set interfaces lo0 unit 0 family inet address 192.168.27.13/32 primary
set interfaces lo0 unit 0 family inet address 127.0.0.1/32
set interfaces lo0 unit 1 family inet address 10.10.47.101/32
set protocols pim rp static address 10.255.71.47
set protocols pim interface fxp0.0 disable
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface t1-1/0/0:0.0
set routing-instances VPN-A interface lo0.1
set routing-instances VPN-A route-distinguisher 10.255.71.46:100
set routing-instances VPN-A vrf-import VPNA-import
set routing-instances VPN-A vrf-export VPNA-export
set routing-instances VPN-A protocols ospf export bgp-to-ospf
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface t1-1/0/0:0.0
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface lo0.1
set routing-instances VPN-A protocols pim rib-group inet VPNA-mcast-rib
set routing-instances VPN-A protocols pim rp static address 10.255.245.91
set routing-instances VPN-A protocols pim interface t1-1/0/0:0.0 mode sparse
set routing-instances VPN-A protocols pim interface t1-1/0/0:0.0 version 2
set routing-instances VPN-A protocols pim interface lo0.1 mode sparse
set routing-instances VPN-A protocols pim interface lo0.1 version 2
set routing-instances VPN-A provider-tunnel pim-asm group-address 239.1.1.1
set routing-instances VPN-A protocols pim mvpn
set routing-options interface-routes rib-group inet VPNA-mcast-rib
set routing-options rib-groups VPNA-mcast-rib export-rib VPN-A.inet.2
set routing-options rib-groups VPNA-mcast-rib import-rib VPN-A.inet.2
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure multicast for draft-rosen VPNs:

1. Configure PIM on the P router.

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set dense-groups 224.0.1.39/32
[edit protocols pim]
user@host# set dense-groups 224.0.1.40/32
[edit protocols pim]
user@host# set rp local address 10.255.71.47
[edit protocols pim]
user@host# set interface all mode sparse
```

```
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
```

2. Configure PIM on the PE1 and PE2 routers. Specify a static route to the service provider RP—the P router (10.255.71.47).

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp static address 10.255.71.47
[edit protocols pim]
user@host#  set interface interface all mode sparse
[edit protocols pim]
user@host# set interface interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
[edit protocols pim]
user@host# exit
```

3. Configure PIM on CE1. Specify the RP address for the VPN RP—Router CE2 (10.255.245.91).

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp static address 10.255.245.91
[edit protocols pim]
user@host# set interface all mode sparse
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
[edit protocols pim]
user@host# exit
```

4. Configure PIM on CE2, which acts as the VPN RP. Specify CE2's address (10.255.245.91).

```
[edit]
user@host# edit protocols pim
[edit protocols pim]
user@host# set rp local address 10.255.245.91
[edit protocols pim]
user@host# set interface all mode sparse
[edit protocols pim]
user@host# set interface all version 2
[edit protocols pim]
user@host# set interface fxp0.0 disable
[edit protocols pim]
user@host# exit
```

5. On PE1, configure the routing instance (VPN-A) for the Layer 3 VPN.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set instance-type vrf
[edit routing-instances VPN-A]
user@host# set interface t1-1/0/0:0.0
[edit routing-instances VPN-A]
user@host# set interface lo0.1
[edit routing-instances VPN-A]
user@host# set route-distinguisher 10.255.71.46:100
[edit routing-instances VPN-A]
user@host# set vrf-import VPNA-import
[edit routing-instances VPN-A]
user@host# set vrf-export VPNA-export
```

6. On PE1, configure the IGP policy to advertise the interfaces in the VPN address space.

```
[edit routing-instances VPN-A]
user@host# set protocols ospf export bgp-to-ospf
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface t1-1/0/0:0.0
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface lo0.1
```

7. On PE1, set the RP configuration for the VRF instance. The RP configuration within the VRF instance provides explicit knowledge of the RP address, so that the (*,G) state can be forwarded.

```
[edit routing-instances VPN-A]
user@host# set protocols pim mvpn
[edit routing-instances VPN-A]
user@host# set protocols provider-tunnel pim-asm group-address 239.1.1.1
[edit routing-instances VPN-A]
user@host# set protocols pim rp static address 10.255.245.91
[edit routing-instances VPN-A]
user@host# set protocols pim interface t1-1/0/0:0.0 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface t1-1/0/0:0.0 version 2
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 version 2
[edit routing-instances VPN-A]
user@host# exit
```

8. On PE1, configure the loopback interfaces.

```
[edit]
user@host# edit interface lo0
[edit interface lo0]
user@host# set unit 0 family inet address 192.168.27.13/32 primary
[edit interface lo0]
user@host# set unit 0 family inet address 127.0.0.1/32
[edit interface lo0]
```

```
user@host# set unit 1 family inet address 10.10.47.101/32
[edit interface lo0]
user@host# exit
```

9. As you did for the PE1 router, configure the PE2 router.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set instance-type vrf
[edit routing-instances VPN-A]
user@host# set interface t1-2/0/0:0.0
[edit routing-instances VPN-A]
user@host# set interface lo0.1
[edit routing-instances VPN-A]
user@host# set route-distinguisher 10.255.71.51:100
[edit routing-instances VPN-A]
user@host# set vrf-import VPNA-import
[edit routing-instances VPN-A]
user@host# set vrf-export VPNA-export
[edit routing-instances VPN-A]
user@host# set protocols ospf export bgp-to-ospf
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface t1-2/0/0:0.0
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface lo0.1
[edit routing-instances VPN-A]
user@host# set protocols pim rp static address 10.255.245.91
[edit routing-instances VPN-A]
user@host# set protocols pim mvpn
[edit routing-instances VPN-A]
user@host# set protocols pim interface t1-2/0/0:0.0 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 mode sparse
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 version 2
[edit routing-instances VPN-A]
user@host# set provider-tunnel pim-asm group-address 239.1.1.1
user@host# exit
[edit]
user@host# edit interface lo0
[edit interface lo0]
user@host# set unit 0 family inet address 192.168.27.14/32 primary
[edit interface lo0]
user@host# set unit 0 family inet address 127.0.0.1/32
[edit interface lo0]
user@host# set unit 1 family inet address 10.10.47.102/32
```

10. When one of the PE routers is running Cisco Systems IOS software, you must configure the Juniper Networks PE router to support this multicast interoperability requirement. The Juniper Networks PE router must have the **lo0.0** interface in the master routing instance and the **lo0.1** interface assigned to the VPN routing instance. You must configure the **lo0.1** interface with the same IP address that the **lo0.0** interface uses for BGP peering in the provider core in the master routing instance.

Configure the same IP address on the **lo0.0** and **lo0.1** loopback interfaces of the Juniper Networks PE router at the **[edit interfaces lo0]** hierarchy level, and assign the address used for BGP peering in the provider core in the master routing instance. In this alternate example, unit 0 and unit 1 are configured for Cisco IOS interoperability.

```
[edit interface lo0]
user@host# set unit 0 family inet address 192.168.27.14/32 primary
[edit interface lo0]
user@host# set unit 0 family inet address 127.0.0.1/32
[edit interface lo0]
user@host# set unit 1 family inet address 192.168.27.14/32
[edit interface lo0]
user@host# exit
```

11. Configure the multicast routing table group. This group accesses **inet.2** when doing RPF checks. However, if you are using **inet.0** for multicast RPF checks, this step will prevent your multicast configuration from working.

```
[edit]
user@host# edit routing-options
[edit routing-options]
user@host# set interface-routes rib-group inet VPNA-mcast-rib
[edit routing-options]
user@host# set rib-groups VPNA-mcast-rib export-rib VPN-A.inet.2
[edit routing-options]
user@host# set rib-groups VPNA-mcast-rib import-rib VPN-A.inet.2
[edit routing-options]
user@host# exit
```

12. Activate the multicast routing table group in the VPN's VRF instance.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set protocols pim rib-group inet VPNA-mcast-rib
```

13. If you are done configuring the device, commit the configuration.

```
[edit routing-instances VPN-A]
user@host# commit
```

Results

Confirm your configuration by entering the **show interfaces**, **show protocols**, **show routing-instances**, and **show routing-options** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration. This output shows the configuration on PE1.

```
user@host# show interfaces
lo0 {
   unit 0 {
      family inet {
```

```
        address 192.168.27.13/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
    unit 1 {
      family inet {
        address 10.10.47.101/32;
      }
    }
}
```

```
user@host# show protocols
pim {
    rp {
        static {
            address 10.255.71.47;
        }
    }
    interface fxp0.0 {
        disable;
    }
    interface all {
        mode sparse;
        version 2;
    }
}
```

```
user@host# show routing-instances
VPN-A {
    instance-type vrf;
    interface t1-1/0/0:0.0;
    interface lo0.1;
    route-distinguisher 10.255.71.46:100;
    vrf-import VPNA-import;
    vrf-export VPNA-export;
    provider-tunnel {
        pim-asm {
            group-address 239.1.1.1;
        }
    }
    protocols {
        ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
                interface t1-1/0/0:0.0;
                interface lo0.1;
            }
        }
        pim {
            mvpn;
            rib-group inet VPNA-mcast-rib;
            rp {
                static {
                    address 10.255.245.91;
                }
```

```
        }
        interface t1-1/0/0:0.0 {
          mode sparse;
          version 2;
        }
        interface lo0.1 {
          mode sparse;
          version 2;
        }
      }
    }
  }
```

```
user@host# show routing-options
interface-routes {
  rib-group inet VPNA-mcast-rib;
}
rib-groups {
  VPNA-mcast-rib {
    export-rib VPN-A.inet.2;
    import-rib VPN-A.inet.2;
  }
}
```

Verification

To verify the configuration, run the following commands:

1. Display multicast tunnel information and the number of neighbors by using the show pim interfaces **instance** ***instance-name*** command from the PE1 or PE2 router. When issued from the PE1 router, the output display is:

   ```
   user@host> show pim interfaces instance VPN-A
   Instance: PIM.VPN-A
   Name                 Stat Mode       IP V State Count DR address
   lo0.1                Up   Sparse      4 2 DR        0 10.10.47.101
   mt-1/1/0.32769       Up   Sparse      4 2 DR        1
   mt-1/1/0.1081346       Up   Sparse      4 2 DR        0
   pe-1/1/0.32769       Up   Sparse      4 1 P2P       0
   t1-2/1/0:0.0         Up   Sparse      4 2 P2P       1
   ```

   You can also display all PE tunnel interfaces by using the show pim join command from the provider router acting as the RP.

2. Display multicast tunnel interface information, DR information, and the PIM neighbor status between VRF instances on the PE1 and PE2 routers by using the show pim neighbors **instance** ***instance-name*** command from either PE router. When issued from the PE1 router, the output is as follows:

   ```
   user@host> show pim neighbors instance VPN-A
   Instance: PIM.VPN-A
   Interface           IP V Mode        Option      Uptime Neighbor addr
   mt-1/1/0.32769       4 2             HPL       01:40:46 10.10.47.102
   t1-1/0/0:0.0         4 2             HPL       01:41:41 192.168.196.178
   ```

See Also
- Example: Configuring PIM RPF Selection on page 816

Load Balancing Multicast Tunnel Interfaces Among Available PICs

When you configure multicast on draft-rosen Layer 3 VPNs, multicast tunnel interfaces are automatically generated to encapsulate and de-encapsulate control and data traffic.

To generate multicast tunnel interfaces, a routing device must have one or more of the following tunnel-capable PICs:

- Adaptive Services PIC
- Multiservices PIC or Multiservices DPC
- Tunnel Services PIC
- On MX Series routers, a PIC created with the **tunnel-services** statement at the **[edit chassis fpc *slot-number* pic *number*]** hierarchy level

NOTE: A *routing device* is a router or an EX Series switch that is functioning as a router.

If a routing device has multiple such PICs, it might be important in your implementation to load balance the tunnel interfaces across the available tunnel-capable PICs.

The multicast tunnel interface that is used for encapsulation, **mt-[*xxxxx*]**, is in the range from 32,768 through 49,151. The interface **mt-[*yyyyy*]**, used for de-encapsulation, is in the range from 1,081,344 through 1,107,827. PIM runs only on the encapsulation interface. The de-encapsulation interface populates downstream interface information. For the default MDT, an instance's de-encapsulation and encapsulation interfaces are always created on the same PIC.

For each VPN, the PE routers build a multicast distribution tree within the service provider core network. After the tree is created, each PE router encapsulates all multicast traffic (data and control messages) from the attached VPN and sends the encapsulated traffic to the VPN group address. Because all the PE routers are members of the outgoing interface list in the multicast distribution tree for the VPN group address, they all receive the encapsulated traffic. When the PE routers receive the encapsulated traffic, they de-encapsulate the messages and send the data and control messages to the CE routers.

If a routing device has multiple tunnel-capable PICs (for example, two Tunnel Services PICs), the routing device load balances the creation of tunnel interfaces among the available PICs. However, in some cases (for example, after a reboot), a single PIC might be selected for all of the tunnel interfaces. This causes one PIC to have a heavy load, while other available PICs are underutilized. To prevent this, you can manually configure load balancing. Thus, you can configure and distribute the load uniformly across the available PICs.

The definition of a balanced state is determined by you and by the requirements of your Layer 3 VPN implementation. You might want all of the instances to be evenly distributed

across the available PICs or across a configured list of PICs. You might want all of the encapsulation interfaces from all of the instances to be evenly distributed across the available PICs or across a configured list of PICs. If the bandwidth of each tunnel encapsulation interface is considered, you might choose a different distribution. You can design your load-balancing configuration based on each instance or on each routing device.

NOTE: In a Layer 3 VPN, each of the following routing devices must have at least one tunnel-capable PIC:

- Each provider edge (PE) router.
- Any provider (P) router acting as the RP.
- Any customer edge (CE) router that is acting as a source's DR or as an RP. A receiver's designated router does not need a tunnel-capable PIC.

To configure load balancing:

1. On an M Series or T Series router or on an EX Series switch, install more than one tunnel-capable PIC. (In some implementations, only one PIC is required. Load balancing is based on the assumption that a routing device has more than one tunnel-capable PIC.)

2. On an MX Series router, configure more than one tunnel-capable PIC.

   ```
   [edit chassis fpc 0]
   user@host# set pic 0 tunnel-services bandwidth 10g
   user@host# set pic 1 tunnel-services bandwidth 10g
   ```

3. Configure Layer 3 VPNs as described in "Example: Configuring Any-Source Multicast for Draft-Rosen VPNs" on page 449.

   ```
   [edit routing-instances vpn1]
   user@host# set provider-tunnel pim-asm group-address 234.1.1.1
   user@host# set protocols pim rp static address 10.255.72.48
   user@host# set protocols pim interface fe-1/0/0.0
   user@host# set protocols pim interface lo0.1
   user@host# set protocols pim mvpn
   ```

4. For each VPN, specify a PIC list.

   ```
   [edit routing-instances vpn1 protocols pim]
   user@host# set tunnel-devices [ mt-1/1/0 mt-1/2/0 mt-2/0/0 ]
   ```

   The physical position of the PIC in the routing device determines the multicast tunnel interface name. For example, if you have an Adaptive Services PIC installed in FPC slot 0 and PIC slot 0, the corresponding multicast tunnel interface name is **mt-0/0/0**. The same is true for Tunnel Services PICs, Multiservices PICs, and Multiservices DPCs.

   In the **tunnel-devices** statement, the order of the PIC list that you specify does not impact how the interfaces are allocated. An instance uses all of the listed PICs to

create default encapsulation and de-encapsulation interfaces, and data MDT encapsulation interfaces. The instance uses a round-robin approach to distributing the tunnel interfaces (default and data MDT) across the PIC list (or across the available PICs, in the absence of a PIC list).

For the first tunnel, the round-robin algorithm starts with the lowest-numbered PIC. The second tunnel is created on the next-lowest-numbered PIC, and so on, round and round. The selection algorithm works routing device-wide. The round robin does not restart at the lowest-numbered PIC for each new instance. This applies to both the default and data MDT tunnel interfaces.

If one PIC in the list fails, new tunnel interfaces are created on the remaining PICs in the list using the round-robin algorithm. If all the PICs in the list go down, all tunnel interfaces are deleted and no new tunnel interfaces are created. If a PIC in the list comes up from the down state and the restored PIC is the only PIC that is up, the interfaces are reassigned to the restored PIC. If a PIC in the list comes up from the down state and other PICs are already up, an interface reassignment is not done. However, when a new tunnel interface needs to be created, the restored PIC is available for the selection process. If you include in the PIC list a PIC that is not installed on the routing device, the PIC is treated as if it is present but in the down state.

To balance the interfaces among the instances, you can assign one PIC to each instance. For example, if you have vpn1-10 and you have three PICs—for example, **mt-1/1/0**, **mt-1/2/0**, **mt-2/0/0**—you can configure vpn1-4 to only use **mt-1/1/0**, vpn5-7 to use **mt-1/2/0**, and vpn8-10 to use **mt-2/0/0**.

5. Commit the configuration.

   ```
   user@host# commit
   ```

   When you commit a new PIC list configuration, all the multicast tunnel interfaces for the routing instance are deleted and re-created using the new PIC list.

6. If you reboot the routing device, some PICs come up faster than others. The difference can be minutes. Therefore, when the tunnel interfaces are created, the known PIC list might not be the same as when the routing device is fully rebooted. This causes the tunnel interfaces to be created on some but not all available and configured PICs. To remedy this situation, you can manually rebalance the PIC load.

   Check to determine if a load rebalance is necessary.

   ```
   user@host> show interfaces terse | match mt-
   mt-1/1/0        up     up
   mt-1/1/0.32768  up     up      inet
   mt-1/1/0.1081344  up     up      inet
   mt-1/2/0        up     up
   mt-1/2/0.32769  up     up      inet
   mt-1/2/0.32770  up     up      inet
   mt-1/2/0.32771  up     up      inet
   ```

   The output shows that **mt-1/1/0** has only one tunnel encapsulation interface, while **mt-1/2/0** has three tunnel encapsulation interfaces. In a case like this, you might decide to rebalance the interfaces. As stated previously, encapsulation interfaces are in the

range from 32,768 through 49,151. In determining whether a rebalance is necessary, look at the encapsulation interfaces only, because the default MDT de-encapsulation interface always resides on the same PIC with the default MDT encapsulation interface.

7. (Optional) Rebalance the PIC load.

   user@host#> **request pim multicast-tunnel rebalance instance vpn1**

   This command re-creates and rebalances all tunnel interfaces for a specific instance.

   user@host#> **request pim multicast-tunnel rebalance**

   This command re-creates and rebalances all tunnel interfaces for all routing instances.

8. Verify that the PIC load is balanced.

```
user@host#> show interfaces terse | match mt-
mt-1/1/0        up     up
mt-1/1/0.32770  up     up      inet
mt-1/1/0.32768  up     up      inet
mt-1/1/0.1081344  up     up      inet
mt-1/2/0        up     up
mt-1/2/0.32769  up     up      inet
mt-1/2/0.32771  up     up      inet
```

   The output shows that **mt-1/1/0** has two encapsulation interfaces, and **mt-1/2/0** also has two encapsulation interfaces.

See Also

- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 449
- **request pim multicast-tunnel rebalance on page 1440** command in the CLI Explorer

Related Documentation

- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 489

Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs

- Understanding Source-Specific Multicast VPNs on page 490
- Draft-Rosen 7 Multicast VPN Control Plane on page 490
- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

Understanding Source-Specific Multicast VPNs

A draft-rosen MVPN with service provider tunnels operating in SSM mode uses BGP signaling for autodiscovery of the PE routers. These MVPNs are also referred to as Draft Rosen 7.

Each PE sends an MDT subsequent address family identifier (MDT-SAFI) BGP network layer reachability information (NLRI) advertisement. The advertisement contains the following information:

- Route distinguisher
- Unicast address of the PE router to which the source site is attached (usually the loopback)
- Multicast group address
- Route target extended community attribute

Each remote PE router imports the MDT-SAFI advertisements from each of the other PE routers if the route target matches. Each PE router then joins the (S,G) tree rooted at each of the other PE routers.

After a PE router discovers the other PE routers, the source and group are bound to the VPN routing and forwarding (VRF) through the multicast tunnel de-encapsulation interface.

A draft-rosen MVPN with service provider tunnels operating in any-source multicast sparse-mode uses a shared tree and rendezvous point (RP) for autodiscovery of the PE routers. The PE that is the source of the multicast group encapsulates multicast data packets into a PIM register message and sends them by means of unicast to the RP router. The RP then builds a shortest-path tree (SPT) toward the source PE. The remote PE that acts as a receiver for the MDT multicast group sends (*,G) join messages toward the RP and joins the distribution tree for that group.

Draft-Rosen 7 Multicast VPN Control Plane

The control plane of a draft-rosen MVPN with service provider tunnels operating in SSM mode must be configured to support autodiscovery.

After the PE routers are discovered, PIM is notified of the multicast source and group addresses. PIM binds the (S,G) state to the multicast tunnel (**mt**) interface and sends a join message for that group.

Autodiscovery for a draft-rosen MVPN with service provider tunnels operating in SSM mode uses some of the facilities of the BGP-based MVPN control plane software module. Therefore, the BGP-based MVPN control plane must be enabled. The BGP-based MVPN control plane can be enabled for autodiscovery only.

See Also

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs

This example shows how to configure a draft-rosen Layer 3 VPN operating in source-specific multicast (SSM) mode. This example is based on the Junos OS implementation of the IETF Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP VPNs*.

Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.4 or later
- Make sure that the routing devices support multicast tunnel (**mt**) interfaces.

  A tunnel-capable PIC supports a maximum of 512 multicast tunnel interfaces. Both default and data MDTs contribute to this total. The default MDT uses two multicast tunnel interfaces (one for encapsulation and one for de-encapsulation). To enable an M Series or T Series router to support more than 512 multicast tunnel interfaces, another tunnel-capable PIC is required. See "Tunnel Services PICs and Multicast" on page 216 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 459.

NOTE: In Junos OS Release 17.3R1, the **pim-ssm** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7.

Overview

The IETF Internet draft draft-rosen-vpn-mcast-07.txt introduced the ability to configure the provider network to operate in SSM mode. When a draft-rosen multicast VPN is used over an SSM provider core, there are no PIM RPs to provide rendezvous and autodiscovery between PE routers. Therefore, draft-rosen-vpn-mcast-07 specifies the use of a BGP network layer reachability information (NLRI), called MDT subaddress family identifier information (MDT-SAFI) to facilitate autodiscovery of PEs by other PEs. MDT-SAFI updates are BGP messages distributed between intra-AS internal BGP peer PEs. Thus, receipt of an MDT-SAFI update enables a PE to autodiscover the identity of other PEs with sites for a given VPN and the default MDT (S,G) routes to join for each. Autodiscovery provides the next-hop address of each PE, and the VPN group address for the tunnel rooted at that PE for the given route distinguisher (RD) and route-target extended community attribute.

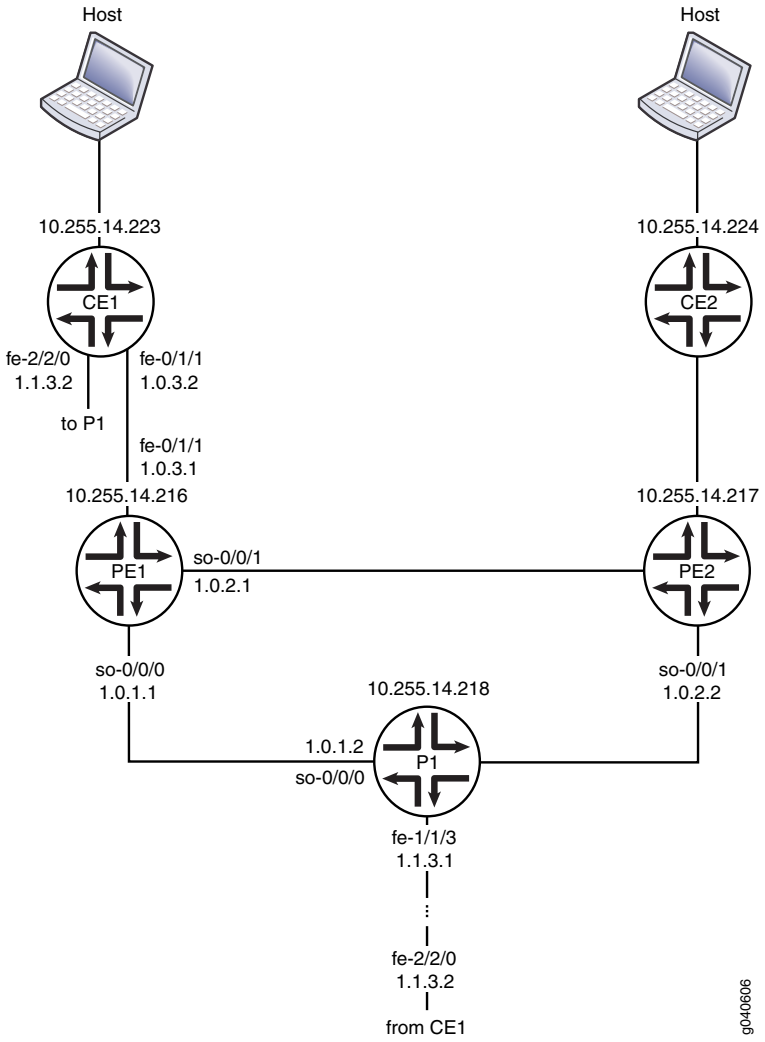
This example includes the following configuration options to enable draft-rosen SSM:

- **protocols bgp group *group-name* family inet-mdt signaling**—Enables MDT-SAFI signaling in BGP.
- **routing-instance *instance-name* protocols mvpn family inet autodiscovery-only intra-as inclusive**—Enables the multicast VPN to use the MDT-SAFI autodiscovery NLRI.
- **routing-instance *instance-name* protocols pim mvpn**—Specifies the SSM control plane. When **pim mvpn** is configured for a VRF, the VPN group address must be specified with the **provider-tunnel pim-ssm group-address** statement.
- **routing-instance *instance-name* protocols pim mvpn family inet autodiscovery inet-mdt**—Enables PIM to learn about neighbors from the MDT-SAFI autodiscovery NLRI.
- **routing-instance *instance-name* provider-tunnel family inet pim-ssm group-address *multicast-address***—Configures the provider tunnel that serves as the control plane and enables the provider tunnel to have a static group address. Unlike draft-rosen multicast VPNs with ASM provider cores, the SSM configuration does not require that each PE for a VPN use the same group address. This is because the rendezvous point assignment and autodiscovery are not accomplished over the default MDT tunnels for the group. Thus, you can configure some or all PEs in a VPN to use a different group, but the same group cannot be used in different VPNs on the same PE router.
- **routing-instances ce1 vrf-target target:100:1**—Configures the VRF export policy. When you configure draft-rosen multicast VPNs with provider tunnels operating in source-specific mode and using the **vrf-target** statement, the VRF export policy is automatically generated and automatically accepts routes from the **vrf-name.mdt.0** routing table.

NOTE: When you configure draft-rosen multicast VPNs with provider tunnels operating in source-specific mode and using the **vrf-export** statement to specify the export policy, the policy must have a term that accepts routes from the **vrf-name.mdt.0** routing table. This term ensures proper PE autodiscovery using the **inet-mdt** address family.

Figure 78 on page 493 shows the topology for this example.

Figure 78: SSM for Draft-Rosen Multicast VPNs Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set interfaces so-0/0/0 description "TO P1_P1"
set interfaces so-0/0/0 unit 0 description "to P1 (provider router) so-0/0/0.0"
set interfaces so-0/0/0 unit 0 family inet address 1.0.1.1/30
set interfaces so-0/0/0 unit 0 family iso
set interfaces so-0/0/0 unit 0 family mpls
set interfaces so-0/0/1 description "TO PE2"
set interfaces so-0/0/1 unit 0 description "to PE2 (PE router) so-0/0/1.0"
set interfaces so-0/0/1 unit 0 family inet address 1.0.2.1/30
set interfaces so-0/0/1 unit 0 family iso
```

```
set interfaces so-0/0/1 unit 0 family mpls
set interfaces fe-0/1/1 description "TO CE1"
set interfaces fe-0/1/1 unit 0 description "to CE router fe-0/1/1.0"
set interfaces fe-0/1/1 unit 0 family inet address 1.0.3.1/30
set interfaces lo0 unit 0 description "PE1 (this PE router) Loopback"
set interfaces lo0 unit 1 family inet address 1.1.1.0/32
set routing-options autonomous-system 200
set protocols igmp query-interval 2
set protocols igmp query-response-interval 1
set protocols igmp query-last-member-interval 1
set protocols igmp interface all immediate-leave
set protocols igmp interface fxp0.0 disable
set protocols rsvp interface all
set protocols rsvp interface so-0/0/0.0
set protocols rsvp interface so-0/0/1.0
set protocols mpls label-switched-path PE1-to-PE2 to 10.255.14.217
set protocols mpls label-switched-path PE1-to-PE2 primary PE1_PE2_prime
set protocols mpls label-switched-path PE1-to-P1 to 10.255.14.218
set protocols mpls label-switched-path PE1-to-P1 primary PE1_P1_prime
set protocols mpls path PE1_P1_prime 1.0.1.2
set protocols mpls path PE1_PE2_prime 1.0.2.2
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group int type internal
set protocols bgp group int local-address 10.255.14.216
set protocols bgp group int family inet unicast
set protocols bgp group int family inet-vpn unicast
set protocols bgp group int family inet-vpn multicast
set protocols bgp group int family inet-mdt signaling
set protocols bgp group int neighbor 10.255.14.218
set protocols bgp group int neighbor 10.255.14.217
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface so-0/0/0.0 metric 10
set protocols ospf area 0.0.0.0 interface so-0/0/1.0 metric 10
set protocols pim assert-timeout 5
set protocols pim join-prune-timeout 210
set protocols pim rp bootstrap-priority 10
set protocols pim rp local address 10.255.14.216
set protocols pim interface lo0.0
set protocols pim interface all hello-interval 1
set protocols pim interface fxp0.0 disable
set policy-options policy-statement bgp_ospf term 1 from protocol bgp
set policy-options policy-statement bgp_ospf term 1 then accept
set routing-instances ce1 instance-type vrf
set routing-instances ce1 interface fe-0/1/1.0
set routing-instances ce1 interface lo0.1
set routing-instances ce1 route-distinguisher 1:0
set routing-instances ce1 provider-tunnel pim-ssm group-address 232.1.1.1
set routing-instances ce1 vrf-target target:100:1
set routing-instances ce1 protocols ospf export bgp_ospf
set routing-instances ce1 protocols ospf sham-link local 1.1.1.0
set routing-instances ce1 protocols ospf area 0.0.0.0 sham-link-remote 1.1.1.1
set routing-instances ce1 protocols ospf area 0.0.0.0 sham-link-remote 1.1.1.2
set routing-instances ce1 protocols ospf area 0.0.0.0 interface lo0.1
set routing-instances ce1 protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 10
```

```
set routing-instances ce1 protocols pim mvpn family inet autodiscovery inet-mdt
set routing-instances ce1 protocols pim interface lo0.1
set routing-instances ce1 protocols pim interface fe-0/1/1.0 priority 100
set routing-instances ce1 protocols pim interface fe-0/1/1.0 hello-interval 1
set routing-instances ce1 protocols mvpn family inet autodiscovery-only intra-as inclusive
```

Interface Configuration

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the interfaces on one PE router:

1. Configure PE1's interface to the provider router.

```
[edit interfaces so-0/0/0]
user@host# set description "TO P1"
user@host# set unit 0 description "to P1 (provider router, 10.255.14.218 ) so-0/0/0.0"
user@host# set unit 0 family inet address 1.0.1.1/30
user@host# set unit 0 family iso
user@host# set unit 0 family mpls
```

2. Configure PE1's interface to PE2.

```
[edit interfaces so-0/0/1]
user@host# set description "TO PE2"
user@host# set unit 0 description "to PE2 (10.255.14.217) so-0/0/1.0"
user@host# set unit 0 family inet address 1.0.2.1/30
user@host# set unit 0 family iso
user@host# set unit 0 family mpls
```

3. Configure PE1's interface to CE1.

```
[edit interfaces fe-0/1/1]
user@host# set description "TO CE1"
user@host# set unit 0 description "to CE1 (10.255.14.223) fe-0/1/1.0"
user@host# set unit 0 family inet address 1.0.3.1/30
user@host# set unit 0 family iso
user@host# set unit 0 family mpls
```

4. Configure PE1's loopback interface.

```
[edit interfaces lo0]
user@host# set unit 0 description "PE1 (this PE router, 10.255.14.216) Loopback"
user@host# set unit 1 family inet address 1.1.1.0/32
```

Multicast Group Management

Step-by-Step Procedure

To configure multicast group management:

1. Configure the IGMP interfaces.

```
[edit protocols igmp]
user@host# set interface all immediate-leave
user@host# set interface fxp0.0 disable
```

2. Configure the IGMP settings.

```
[edit protocols igmp]
user@host# set query-interval 2
user@host# set query-response-interval 1
user@host# set query-last-member-interval 1
```

MPLS Signaling Protocol and MPLS LSPs

Step-by-Step Procedure

To configure the MPLS signaling protocol and MPLS LSPs:

1. Configure RSVP signaling among this PE router (PE1), the other PE router (PE2). and the provider router (P1).

```
[edit protocols rsvp]
user@host# set interface so-0/0/0.0
user@host# set interface so-0/0/1.0
```

2. Configure MPLS LSPs.

```
[edit protocols mpls]
user@host# set label-switched-path pe1-to-pe2 to 10.255.14.217
user@host# set label-switched-path pe1-to-pe2 primary pe1_pe2_prime
user@host# set label-switched-path pe1-to-p1 to 10.255.14.218
user@host# set label-switched-path pe1-to-p1 primary pe1_p1_prime
user@host# set path pe1_p1_prime 1.0.1.2
user@host# set path pe1_pe2_prime 1.0.2.2
user@host# set interface all
user@host# set interface fxp0.0 disable
```

BGP

Step-by-Step Procedure

To configure BGP:

1. Configure the AS number. In this example, both of the PE routers and the provider router are in AS 200.

```
[edit]
user@host# set routing-options autonomous-system 200
```

2. Configure the internal BGP full mesh with the PE2 and P1 routers.

```
[edit protocols bgp group int]
user@host# set type internal
user@host# set local-address 10.255.14.216
user@host# set family inet unicast
user@host# set neighbor 10.255.14.218
user@host# set neighbor 10.255.14.217
```

3. Enable MDT-SAFI NLRI control plane messages.

```
[edit protocols bgp group int]
user@host# set family inet-mdt signaling
```

4. Enable BGP to carry Layer 3 VPN NLRI for the IPv4 address family.

```
[edit protocols bgp group int]
user@host# set family inet-vpn unicast
user@host# set family inet-vpn multicast
```

5. Configure BGP export policy.

```
[edit policy-options]
user@host# set policy-statement bgp_ospf term 1 from protocol bgp
user@host# set policy-statement bgp_ospf term 1 then accept
```

Interior Gateway Protocol

Step-by-Step Procedure

To configure the interior gateway protocol:

1. Configure the OSPF interfaces.

```
[edit protocols ospf]
user@host# set area 0.0.0.0 interface lo0.0 passive
user@host# set area 0.0.0.0 interface so-0/0/0.0 metric 10
user@host# set area 0.0.0.0 interface so-0/0/1.0 metric 10
```

2. Enable traffic engineering.

```
[edit protocols ospf]
user@host# set traffic-engineering
```

PIM

Step-by-Step Procedure

To configure PIM:

1. Configure timeout periods and the RP. Local RP configuration makes PE1 a statically defined RP.

```
[edit protocols pim]
user@host# set assert-timeout 5
user@host# set join-prune-timeout 210
```

```
user@host# set rp bootstrap-priority 10
user@host# set rp local address 10.255.14.216
```

2. Configure the PIM interfaces.

```
[edit protocols pim]
user@host# set interface lo0.0
user@host# set interface all hello-interval 1
user@host# set interface fxp0.0 disable
```

Routing Instance

Step-by-Step Procedure

To configure the routing instance between PE1 and CE1:

1. Configure the basic routing instance.

```
[edit routing-instances ce1]
user@host# set instance-type vrf
user@host# set interface fe-0/1/1.0
user@host# set interface lo0.1
user@host# set route-distinguisher 1:0
user@host# set vrf-target target:100:1
```

2. Configure the SSM provider tunnel.

```
[edit routing-instances ce1]
user@host# set provider-tunnel family inet pim-ssm group-address (Routing
  Instances) 232.1.1.1
```

3. Configure OSPF in the routing instance.

```
[edit routing-instances ce1 protocols ospf]
user@host# set export bgp_ospf
user@host# set sham-link local 1.1.1.0
user@host# set area 0.0.0.0 sham-link-remote 1.1.1.1
user@host# set area 0.0.0.0 sham-link-remote 1.1.1.2
user@host# set area 0.0.0.0 interface lo0.1
user@host# set area 0.0.0.0 interface fe-0/1/1.0 metric 10
```

4. Configure PIM in the routing instance.

```
[edit routing-instances ce1 protocols pim]
user@host# set interface lo0.1
user@host# set interface fe-0/1/1.0 priority 100
user@host# set interface fe-0/1/1.0 hello-interval 1
```

5. Configure draft-rosen VPN autodiscovery for provider tunnels operating in SSM mode.

```
[edit routing-instances ce1 protocols pim ]
user@host# set mvpn family inet autodiscovery inet-mdt
```

6. Configure the BGP-based MVPN control plane to provide signaling only for autodiscovery and not for PIM operations.

```
[edit routing-instances ce1 protocols mvpn family inet]
user@host# set autodiscovery-only intra-as inclusive
```

Verification

You can monitor the operation of the routing instance by running the **show route table ce1.mdt.0** command.

You can manage the group-instance mapping for local SSM tunnel roots by running the **show pim mvpn** command.

The **show pim mdt** command shows the tunnel type and source PE address for each outgoing and incoming MDT. In addition, because each PE might have its own default MDT group address, one incoming entry is shown for each remote PE. Outgoing data MDTs are shown after the outgoing default MDT. Incoming data MDTs are shown after all incoming default MDTS.

For troubleshooting, you can configure tracing operations for all of the protocols.

See Also

- Draft-Rosen Multicast VPNs Overview on page 447
- Understanding Data MDTs on page 499
- Data MDT Characteristics on page 501
- Understanding Source-Specific Multicast VPNs on page 490
- Draft-Rosen 7 Multicast VPN Control Plane on page 490

Related Documentation

- Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs on page 448

Examples: Configuring Data MDTs

- Understanding Data MDTs on page 499
- Data MDT Characteristics on page 501
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512
- Example: Enabling Dynamic Reuse of Data MDT Group Addresses on page 517

Understanding Data MDTs

In a draft-rosen Layer 3 multicast virtual private network (MVPN) configured with service provider tunnels, the VPN is multicast-enabled and configured to use the Protocol Independent Multicast (PIM) protocol within the VPN and within the service provider

(SP) network. A multicast-enabled VPN routing and forwarding (VRF) instance corresponds to a multicast domain (MD), and a PE router attached to a particular VRF instance is said to belong to the corresponding MD. For each MD there is a default multicast distribution tree (MDT) through the SP backbone, which connects all of the PE routers belonging to that MD. Any PE router configured with a default MDT group address can be the multicast source of one default MDT.

To provide optimal multicast routing, you can configure the PE routers so that when the multicast source within a site exceeds a traffic rate threshold, the PE router to which the source site is attached creates a new data MDT and advertises the new MDT group address. An advertisement of a new MDT group address is sent in a User Datagram Protocol (UDP) type-length-value (TLV) packet called an *MDT join TLV*. The MDT join TLV identifies the source and group pair (S,G) in the VRF instance as well as the new data MDT group address used in the provider space. The PE router to which the source site is attached sends the MDT join TLV over the default MDT for that VRF instance every 60 seconds as long as the source is active.

All PE routers in the VRF instance receive the MDT join TLV because it is sent over the default MDT, but not all the PE routers join the new data MDT group:

- PE routers connected to receivers in the VRF instance for the current multicast group cache the contents of the MDT join TLV, adding a 180-second timeout value to the cache entry, and also join the new data MDT group.
- PE routers not connected to receivers listed in the VRF instance for the current multicast group also cache the contents of the MDT join TLV, adding a 180-second timeout value to the cache entry, but do not join the new data MDT group at this time.

After the source PE stops sending the multicast traffic stream over the default MDT and uses the new MDT instead, only the PE routers that join the new group receive the multicast traffic for that group.

When a remote PE router joins the new data MDT group, it sends a PIM join message for the new group directly to the source PE router from the remote PE routers by means of a PIM (S,G) join.

If a PE router that has not yet joined the new data MDT group receives a PIM join message for a new receiver for which (S,G) traffic is already flowing over the data MDT in the provider core, then that PE router can obtain the new group address from its cache and can join the data MDT immediately without waiting up to 59 seconds for the next data MDT advertisement.

When the PE router to which the source site is attached sends a subsequent MDT join TLV for the VRF instance over the default MDT, any existing cache entries for that VRF instance are simply refreshed with a timeout value of 180 seconds.

To display the information cached from MDT join TLV packets received by all PE routers in a PIM-enabled VRF instance, use the show pim mdt data-mdt-joins operational mode command.

The source PE router starts encapsulating the multicast traffic for the VRF instance using the new data MDT group after 3 seconds, allowing time for the remote PE routers to join

the new group. The source PE router then halts the flow of multicast packets over the default MDT, and the packet flow for the VRF instance source shifts to the newly created data MDT.

The PE router monitors the traffic rate during its periodic statistics-collection cycles. If the traffic rate drops below the threshold or the source stops sending multicast traffic, the PE router to which the source site is attached stops announcing the MDT join TLVs and switches back to sending on the default MDT for that VRF instance.

See Also

- show pim mdt data-mdt-joins on page 1706 in the CLI Explorer

Data MDT Characteristics

A data multicast distribution tree (MDT) solves the problem of routers flooding unnecessary multicast information to PE routers that have no interested receivers for a particular VPN multicast group.

The default MDT uses multicast tunnel (**mt-**) logical interfaces. Data MDTs also use multicast tunnel logical interfaces. If you administratively disable the physical interface that the multicast tunnel logical interfaces are configured on, the multicast tunnel logical interfaces are moved to a different physical interface that is up. In this case the traffic is sent over the default MDT until new data MDTs are created.

The maximum number of data MDTs for all VPNs on a PE router is 1024, and the maximum number of data MDTs for a VRF instance is 1024. The configuration of a VRF instance can limit the number of MDTs possible. No new MDTs can be created after the 1024 MDT limit is reached in the VRF instance, and all traffic for other sources that exceed the configured limit is sent on the default MDT.

Tear-down of data MDTs depends on the monitoring of the multicast source data rate. This rate is checked once per minute, so if the source data rate falls below the configured value, data MDT deletion can be delayed for up to 1 minute until the next statistics-monitoring collection cycle.

Changes to the configured data MDT limit value do not affect existing tunnels that exceed the new limit. Data MDTs that are already active remain in place until the threshold conditions are no longer met.

In a draft-rosen MVPN in which PE routers are already configured to create data MDTs in response to exceeded multicast source traffic rate thresholds, you can change the group range used for creating data MDTs in a VRF instance. To remove any active data MDTs created using the previous group range, you must restart the PIM routing process. This restart clears all remnants of the former group addresses but disrupts routing and therefore requires a maintenance window for the change.

CAUTION: **Never restart any of the software processes unless instructed to do so by a customer support engineer.**

Multicast tunnel (**mt**) interfaces created because of exceeded thresholds are not re-created if the routing process crashes. Therefore, graceful restart does not automatically reinstate the data MDT state. However, as soon as the periodic statistics collection reveals that the threshold condition is still exceeded, the tunnels are quickly re-created.

Data MDTs are supported for customer traffic with PIM sparse mode, dense mode, and sparse-dense mode. Note that the provider core does not support PIM dense mode.

Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode

This example shows how to configure data multicast distribution trees (MDTs) for a provider edge (PE) router attached to a VPN routing and forwarding (VRF) instance in a draft-rosen Layer 3 multicast VPN operating in source-specific multicast (SSM) mode. The example is based on the Junos OS implementation of RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and on section 7 of the IETF Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs*.

Requirements

Before you begin:

- Make sure that the routing devices support multicast tunnel (**mt**) interfaces.

  A tunnel-capable PIC supports a maximum of 512 multicast tunnel interfaces. Both default and data MDTs contribute to this total. The default MDT uses two multicast tunnel interfaces (one for encapsulation and one for de-encapsulation). To enable an M Series or T Series router to support more than 512 multicast tunnel interfaces, another tunnel-capable PIC is required. See ""Tunnel Services PICs and Multicast" on page 216" and ""Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 459" in the *Multicast Protocols Feature Guide* .

- Make sure that the PE router has been configured for a draft-rosen Layer 3 multicast VPN operating in SSM mode in the provider core.

  In this type of multicast VPN, PE routers discover one another by sending MDT subsequent address family identifier (MDT-SAFI) BGP network layer reachability information (NLRI) advertisements. Key configuration statements for the master instance are highlighted in Table 17 on page 503. Key configuration statements for the VRF instance to which your PE router is attached are highlighted in Table 18 on page 505. For complete configuration details, see ""Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs" on page 491" in the *Multicast Protocols Feature Guide* .

Overview

By using data MDTs in a Layer 3 VPN, you can prevent multicast packets from being flooded unnecessarily to specified provider edge (PE) routers within a VPN group. This option is primarily useful for PE routers in your Layer 3 VPN multicast network that have no receivers for the multicast traffic from a particular source.

- When a PE router that is directly connected to the multicast source (also called the *source PE*) receives Layer 3 VPN multicast traffic that exceeds a configured threshold, a new data MDT tunnel is established between the PE router connected to the source site and its remote PE router neighbors.
- The source PE advertises the new data MDT group as long as the source is active. The periodic announcement is sent over the default MDT for the VRF. Because the data MDT announcement is sent over the default tunnel, all the PE routers receive the announcement.
- Neighbors that do not have receivers for the multicast traffic cache the advertisement of the new data MDT group but ignore the new tunnel. Neighbors that do have receivers for the multicast traffic cache the advertisement of the new data MDT group and also send a PIM join message for the new group.
- The source PE encapsulates the VRF multicast traffic using the new data MDT group and stops the packet flow over the default multicast tree. If the multicast traffic level drops back below the threshold, the data MDT is torn down automatically and traffic flows back across the default multicast tree.
- If a PE router that has not yet joined the new data MDT group receives a PIM join message for a new receiver for which (S,G) traffic is already flowing over the data MDT in the provider core, then that PE router can obtain the new group address from its cache and can join the data-MDT immediately without waiting up to 59 seconds for the next data MDT advertisement.

By default, automatic creation of data MDTs is disabled.

The following sections summarize the data MDT configuration statements used in this example and in the prerequisite configuration for this example:

- In the master instance, the PE router's prerequisite draft-rosen PIM-SSM multicast configuration includes statements that directly support the data MDT configuration you will enable in this example. Table 17 on page 503 highlights some of these statements[†].

Table 17: Data MDTS—Key Prerequisites in the Master Instance

Statement	Description

[edit protocols]
pim {
  interface *interface-name* <*options*>;
}
	Enables the PIM protocol on PE router interfaces.

Table 17: Data MDTS—Key Prerequisites in the Master Instance (continued)

Statement	Description

[edit protocols]
bgp {
  group *name* {
  type internal;
  peer-as *autonomous-system*;
  neighbor *address*;
  family inet-mdt {
  signaling;
  }
  }
}
 	
[edit routing-options]
autonomous-system *autonomous-system*;
	In the internal BGP full mesh between PE routers in the VRF instance, enables the BGP protocol to carry MDT-SAFI NLRI signaling messages for IPv4 traffic in Layer 3 VPNs.

[edit routing-options]
multicast {
  ssm-groups [*ip-addresses*];
}
	(Optional) Configures one or more SSM groups to use inside the provider network in addition to the default SSM group address range of 232.0.0.0/8.
	
	NOTE: For this example, it is assumed that you previously specified an additional SSM group address range of 239.0.0.0/8.

† This table contains only a partial list of the PE router configuration statements for a draft-rosen multicast VPN operating in SSM mode in the provider core. For complete configuration information about this prerequisite, see "“Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs” on page 491" in the *Multicast Protocols Feature Guide* .

- In the VRF instance to which the PE router is attached—at the **[edit routing-instances *name*]** hierarchy level—the PE router's prerequisite draft-rosen PIM-SSM multicast configuration includes statements that directly support the data MDT configuration you will enable in this example. Table 18 on page 505 highlights some of these statements‡.

Table 18: Data MDTs—Key Prerequisites in the VRF Instance

Statement	Description
[edit routing-instances *name*] instance-type vrf; vrf-target *community*;	Creates a VRF table (***insta*** contains the routes origin for the Layer 3 VPN. Creates a VRF export poli accepts routes from the ***i*** routing table. ensures prop the **inet-mdt** address fam You must also configure t **route-distinguisher** staten routing instance.

[edit routing-instances *name*]
protocols {
  pim {
  mvpn {
  family {
  inet | *inet6* {
  autodiscovery {
  inet-mdt;
  }
  }
  }
  }
  }
}
	Configures the PE router i
	MDT-SAFI NLRI for autod
	routers:

[edit routing-instances *name*]
provider-tunnelfamily *inet* | *inet6*{
  pim-ssm {
  group-address (Routing Instances) *address*;
  }
}
	Configures the PIM-SSM
	MDT group address.
	
	NOTE: For this example,
	previously configured the
	default MDT for the VPN i
	address **239.1.1.1**.
	
	To verify the configuration
	for the VRF instance to w
	attached, use the **show pi**
	command.

‡ This table contains only a partial list of the PE router configuration statements for a draft-rosen multicast VPN operating in SSM mode in the provi
configuration information about this prerequisite, see ""Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs" on page 491"
Feature Guide .

- For a rosen 7 MVPN—a draft-rosen multicast VPN with provider tunnels operating in SSM mode—you configure data MDT creation for a tunnel multicast group by including statements under the PIM-SSM provider tunnel configuration for the VRF instance associated with the multicast group. Because data MDTs are specific to VPNs and VRF routing instances, you cannot configure MDT statements in the master routing instance. Table 19 on page 506 summarizes the data MDT configuration statements for PIM-SSM provider tunnels.

Table 19: Data MDTs for PIM-SSM Provider Tunnels in a Draft-Rosen MVPN

Statement	Description

[edit routing-instances *name*]
provider-tunnel family *inet* | *inet6*{{
mdt {
group-range *multicast-prefix*;
}
}
	Configures the IP group range used when a new data MDT needs to be created in the VRF instance on the PE router. This address range cannot overlap the default MDT addresses of any other VPNs on the router. If you configure overlapping group ranges, the configuration commit fails.
	
	This statement has no default value. If you do not set the ***multicast-prefix*** to a valid, nonreserved multicast address range, then no data MDTs are created for this VRF instance.
	
	NOTE: For this example, it is assumed that you previously configured the PE router to automatically select an address from the **239.10.10.0/24** range when a new data MDT needs to be initiated.

[edit routing-instances *name*]
provider-tunnel family *inet* | *inet6*{{
mdt {
tunnel-limit *limit*;
}
}
	Configures the maximum number of data MDTs that can be created for the VRF instance.
	
	The default value is 0. If you do not configure the ***limit*** to a non-zero value, then no data MDTs are created for this VRF instance.
	
	The valid range is from 0 through 1024 for a VRF instance. There is a limit of 8000 tunnels for all data MDTs in all VRF instances on a PE router.
	
	If the configured maximum number of data MDT tunnels is reached, then no new tunnels are created for the VRF instance, and traffic that exceeds the configured threshold is sent on the default MDT.
	
	NOTE: For this example, you limit the number of data MDTs for the VRF instance to 10.

Table 19: Data MDTs for PIM-SSM Provider Tunnels in a Draft-Rosen MVPN (continued)

Statement	Description

[edit routing-instances *name*]
provider-tunnel family *inet* | *inet6*{{
mdt {
threshold {
group *group-address* {
source *source-address* {
rate *threshold-rate*;
}
}
}
}
}
	Configures a data rate for the multicast source of a default MDT. When the source traffic in the VRF instance exceeds the configured data rate, a new tunnel is created.
	• **group *group-address***—Multicast group address of the default MDT that corresponds to a VRF instance to which the PE router is attached. The ***group-address*** explicit (all 32 bits of the address specified) or a prefix (network address and prefix length specified). This is typically a well-known address for a certain type of multicast traffic.
	• **source *source-address***—Unicast IP prefix of one or more multicast sources in the specified default MDT group.
	• **rate *threshold-rate***—Data rate for the multicast source to trigger the automatic creation of a data MDT. The data rate is specified in kilobits per second (Kbps).
	The default ***threshold-rate*** is 10 kilobits per second (Kbps).
	NOTE:
	For this example, you configure the following data MDT threshold:
	• Multicast group address or address range to which the threshold limits apply—224.0.9.0/32
	• Multicast source address or address range to which the threshold limits apply—10.1.1.2/32
	• Data rate—10 Kbps
	When the traffic stops or the rate falls below the threshold value, the source PE router switches back to the default MDT.

Topology

Figure 79 on page 508 shows a default MDT.

Figure 79: Default MDT

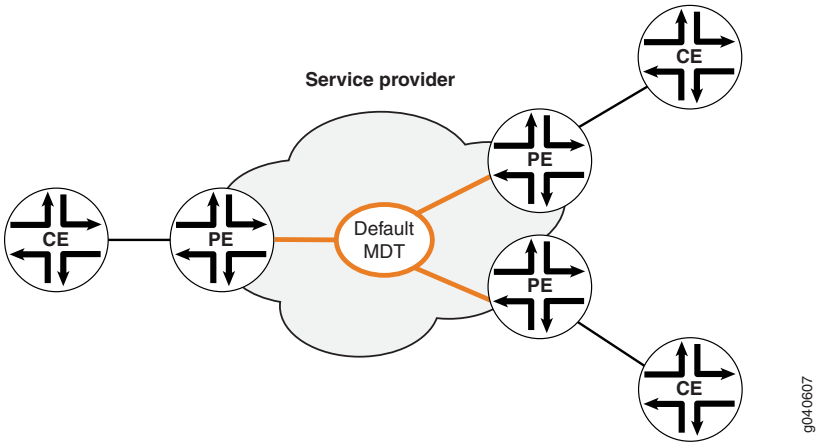
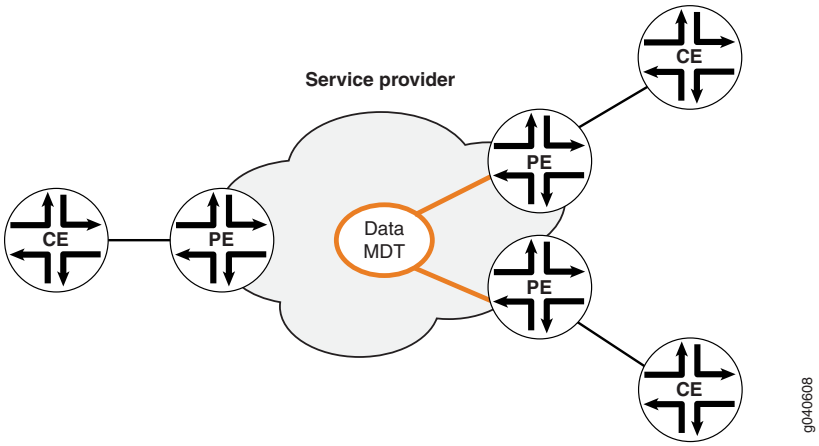


Figure 80 on page 508 shows a data MDT.

Figure 80: Data MDT



Configuration

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the *CLI User Guide*.

- Enabling Data MDTs and PIM-SSM Provider Tunnels on the Local PE Router Attached to a VRF on page 509
- (Optional) Enabling Logging of Detailed Trace Information for Multicast Tunnel Interfaces on the Local PE Router on page 510
- Results on page 511

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level and then enter **commit** from configuration mode.

```
set routing-instances ce1 provider-tunnel family inet mdt group-range 239.10.10.0/24
```

```
set routing-instances ce1 provider-tunnel family inet mdt tunnel-limit 10
set routing-instances ce1 provider-tunnel family inet mdt threshold group 224.0.9.0/32
  source 10.1.1.2/32 rate 10
set protocols pim traceoptions file trace-pim-mdt
set protocols pim traceoptions file files 5
set protocols pim traceoptions file size 1m
set protocols pim traceoptions file world-readable
set protocols pim traceoptions flag mdt detail
```

Enabling Data MDTs and PIM-SSM Provider Tunnels on the Local PE Router Attached to a VRF

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the local PE router attached to the VRF instance **ce1** in a PIM-SSM multicast VPN to initiate new data MDTs and provider tunnels for that VRF:

1. Enable configuration of provider tunnels operating in SSM mode.

   ```
   [edit]
   user@host# edit routing-instances ce1 provider-tunnel
   ```

2. Configure the range of multicast IP addresses for new data MDTs.

   ```
   [edit routing-instances ce1 provider-tunnel]
   user@host# set mdt group-range 239.10.10.0/24
   ```

3. Configure the maximum number of data MDTs for this VRF instance.

   ```
   [edit routing-instances ce1 provider-tunnel]
   user@host# set mdt tunnel-limit 10
   ```

4. Configure the data MDT-creation threshold for a multicast group and source.

   ```
   [edit routing-instances ce1 provider-tunnel]
   user@host# set mdt threshold group 224.0.9.0/32 source 10.1.1.2/32 rate 10
   ```

5. If you are done configuring the device, commit the configuration.

   ```
   [edit]
   user@host# commit
   ```

Results

Confirm the configuration of data MDTs for PIM-SSM provider tunnels by entering the **show routing-instances** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show routing-instances
ce1 {
```

```
        instance-type vrf;
        vrf-target target:100:1;
        ...
        provider-tunnel {
          pim-ssm {
            group-address 239.1.1.1;
          }
          mdt {
            threshold {
              group 224.0.9.0/32 {
                source 10.1.1.2/32 {
                  rate 10;
                }
              }
            }
            tunnel-limit 10;
            group-range 239.10.10.0/24;
          }
        }
        protocols {
          ...
          pim {
            mvpn {
                 family {
                   inet {
                     autodiscovery {
                       inet-mdt;
                     }
                   }
                 }
              }
            }
          }
        }
```

NOTE: The **show routing-instances** command output above does not show the complete configuration of a VRF instance in a draft-rosen MVPN operating in SSM mode in the provider core.

(Optional) Enabling Logging of Detailed Trace Information for Multicast Tunnel Interfaces on the Local PE Router

Step-by-Step Procedure

To enable logging of detailed trace information for all multicast tunnel interfaces on the local PE router:

1. Enable configuration of PIM tracing options.

   ```
   [edit]
   user@host# set protocols pim traceoptions
   ```

2. Configure the trace file name, maximum number of trace files, maximum size of each trace file, and file access type.

```
[edit protocols pim traceoptions]
set file trace-pim-mdt
set file files 5
set file size 1m
set file world-readable
```

3. Specify that messages related to multicast data tunnel operations are logged.

```
[edit protocols pim traceoptions]
set flag mdt detail
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@host# commit
```

Results

Confirm the configuration of multicast tunnel logging by entering the **show protocols** command from configuration mode. If the output does not display the intended configuration, repeat the instructions in this procedure to correct the configuration.

```
[edit]
user@host# show protocols
pim {
   traceoptions {
      file trace-pim-mdt size 1m files 5 world-readable;
      flag mdt detail;
   }
   interface lo0.0;
   ...
}
```

Verification

To verify that the local PE router is managing data MDTs and PIM-SSM provider tunnels properly, perform the following tasks:

- Monitor Data MDTs Initiated for the Multicast Group on page 511
- Monitor Data MDT Group Addresses Cached by All PE Routers in the Multicast Group on page 512
- (Optional) View the Trace Log for Multicast Tunnel Interfaces on page 512

Monitor Data MDTs Initiated for the Multicast Group

Purpose For the VRF instance **ce1**, check the incoming and outgoing tunnels established by the local PE router for the default MDT and monitor the data MDTs initiated by the local PE router.

Action Use the show pim mdt **instance ce1 detail** operational mode command.

For the default MDT, the command displays details about the incoming and outgoing tunnels established by the local PE router for specific multicast source addresses in the multicast group using the default MDT and identifies the tunnel mode as **PIM-SSM**.

For the data MDTs initiated by the local PE router, the command identifies the multicast source using the data MDT, the multicast tunnel logical interface set up for the data MDT tunnel, the configured threshold rate, and current statistics.

Monitor Data MDT Group Addresses Cached by All PE Routers in the Multicast Group

Purpose For the VRF instance **ce1**, check the data MDT group addresses cached by all PE routers that participate in the VRF.

Action Use the show pim mdt data-mdt-joins **instance ce1** operational mode command. The command output displays the information cached from MDT join TLV packets received by all PE routers participating in the specified VRF instance, including the current timeout value of each entry.

(Optional) View the Trace Log for Multicast Tunnel Interfaces

Purpose If you configured logging of trace Information for multicast tunnel interfaces, you can trace the creation and tear-down of data MDTs on the local router through the **mt** interface-related activity in the log.

Action To view the trace file, use the **file show /var/log/trace-pim-mdt** operational mode command.

See Also

- "Tunnel Services PICs and Multicast on page 216" in the *Multicast Protocols Feature Guide*
- "Load Balancing Multicast Tunnel Interfaces Among Available PICs on page 459" in the *Multicast Protocols Feature Guide*
- "Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491" in the *Multicast Protocols Feature Guide*

Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode

This example shows how to configure data multicast distribution trees (MDTs) in a draft-rosen Layer 3 VPN operating in any-source multicast (ASM) mode. This example is based on the Junos OS implementation of RFC 4364, *BGP/MPLS IP Virtual Private*

Networks (VPNs) and on section 2 of the IETF Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs* (expired April 2004).

Requirements

Before you begin:

- Configure the draft-rosen multicast over Layer 3 VPN scenario.
- Make sure that the routing devices support multicast tunnel (**mt**) interfaces.

  A tunnel-capable PIC supports a maximum of 512 multicast tunnel interfaces. Both default and data MDTs contribute to this total. The default MDT uses two multicast tunnel interfaces (one for encapsulation and one for de-encapsulation). To enable an M Series or T Series router to support more than 512 multicast tunnel interfaces, another tunnel-capable PIC is required. See "Tunnel Services PICs and Multicast" on page 216 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 459.

Overview

By using data multicast distribution trees (MDTs) in a Layer 3 VPN, you can prevent multicast packets from being flooded unnecessarily to specified provider edge (PE) routers within a VPN group. This option is primarily useful for PE routers in your Layer 3 VPN multicast network that have no receivers for the multicast traffic from a particular source.

When a PE router that is directly connected to the multicast source (also called the *source PE*) receives Layer 3 VPN multicast traffic that exceeds a configured threshold, a new data MDT tunnel is established between the PE router connected to the source site and its remote PE router neighbors.

The source PE advertises the new data MDT group as long as the source is active. The periodic announcement is sent over the default MDT for the VRF. Because the data MDT announcement is sent over the default tunnel, all the PE routers receive the announcement.

Neighbors that do not have receivers for the multicast traffic cache the advertisement of the new data MDT group but ignore the new tunnel. Neighbors that do have receivers for the multicast traffic cache the advertisement of the new data MDT group and also send a PIM join message for the new group.

The source PE encapsulates the VRF multicast traffic using the new data MDT group and stops the packet flow over the default multicast tree. If the multicast traffic level drops back below the threshold, the data MDT is torn down automatically and traffic flows back across the default multicast tree.

If a PE router that has not yet joined the new data MDT group receives a PIM join message for a new receiver for which (S,G) traffic is already flowing over the data MDT in the provider core, then that PE router can obtain the new group address from its cache and can join the data-MDT immediately without waiting up to 59 seconds for the next data MDT advertisement.

By default, automatic creation of data MDTs is disabled.

For a rosen 6 MVPN—a draft-rosen multicast VPN with provider tunnels operating in ASM mode—you configure data MDT creation for a tunnel multicast group by including statements under the PIM protocol configuration for the VRF instance associated with the multicast group. Because data MDTs apply to VPNs and VRF routing instances, you cannot configure MDT statements in the master routing instance.

This example includes the following configuration options:

- **group**—Specifies the multicast group address to which the threshold applies. This could be a well-known address for a certain type of multicast traffic.

  The group address can be explicit (all 32 bits of the address specified) or a prefix (network address and prefix length specified). Explicit and prefix address forms can be combined if they do not overlap. Overlapping configurations, in which prefix and more explicit address forms are used for the same source or group address, are not supported.

- **group-range**—Specifies the multicast group IP address range used when a new data MDT needs to be initiated on the PE router. For each new data MDT, one address is automatically selected from the configured group range.

  The PE router implementing data MDTs for a local multicast source must be configured with a range of multicast group addresses. Group addresses that fall within the configured range are used in the join messages for the data MDTs created in this VRF instance. Any multicast address range can be used as the multicast prefix. However, the group address range cannot overlap the default MDT group address configured for any VPN on the router. If you configure overlapping group addresses, the configuration commit operation fails.

- **pim**—Supports data MDTs for service provider tunnels operating in any-source multicast mode.

- **rate**—Specifies the data rate that initiates the creation of data MDTs. When the source traffic in the VRF exceeds the configured data rate, a new tunnel is created. The range is from 10 kilobits per second (Kbps), the default, to 1 gigabit per second (Gbps, equivalent to 1,000,000 Kbps).

- **source**—Specifies the unicast address of the source of the multicast traffic. It can be a source locally attached to or reached through the PE router. A group can have more than one source.

  The source address can be explicit (all 32 bits of the address specified) or a prefix (network address and prefix length specified). Explicit and prefix address forms can be combined if they do not overlap. Overlapping configurations, in which prefix and more explicit address forms are used for the same source or group address, are not supported.

- **threshold**—Associates a rate with a group and a source. The PE router implementing data MDTs for a local multicast source must establish a data MDT-creation threshold for a multicast group and source.

  When the traffic stops or the rate falls below the threshold value, the source PE router switches back to the default MDT.

- **tunnel-limit**—Specifies the maximum number of data MDTs that can be created for a single routing instance. The PE router implementing a data MDT for a local multicast source must establish a limit for the number of data MDTs created in this VRF instance. If the limit is 0 (the default), then no data MDTs are created for this VRF instance.

  If the number of data MDT tunnels exceeds the maximum configured tunnel limit for the VRF, then no new tunnels are created. Traffic that exceeds the configured threshold is sent on the default MDT.

  The valid range is from 0 through 1024 for a VRF instance. There is a limit of 8000 tunnels for all data MDTs in all VRF instances on a PE router.

Figure 81 on page 515 shows a default MDT.

Figure 81: Default MDT

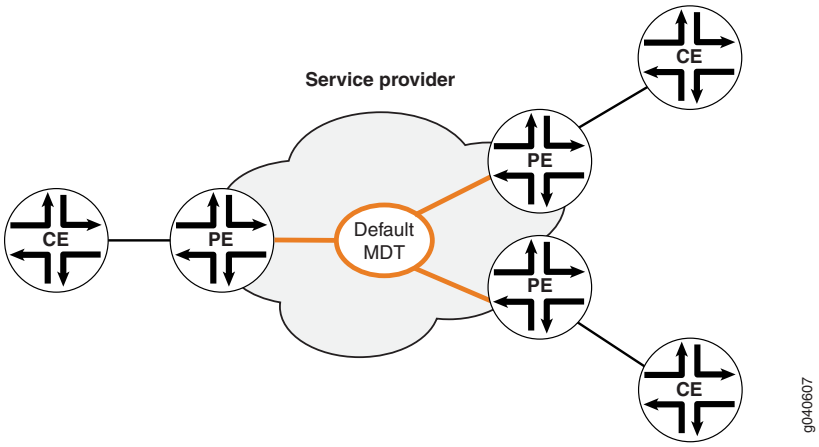
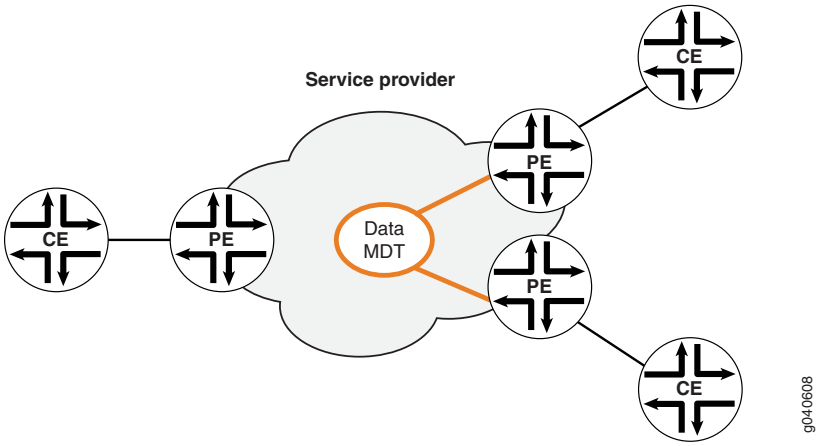


Figure 82 on page 516 shows a data MDT.

Figure 82: Data MDT



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
[edit]
set routing-instances vpn-A protocols pim mdt group-range 227.0.0.0/8
set routing-instances vpn-A protocols pim mdt threshold group 224.4.4.4/32 source
  10.10.20.43/32 rate 10
set routing-instances vpn-A protocols pim mdt tunnel-limit 10
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a PE router attached to the VRF instance **vpn-A** in a PIM-ASM multicast VPN to initiate new data MDTs and provider tunnels for that VRF:

1. Configure the group range.

   ```
   [edit]
   user@host# edit routing-instances vpn-A protocols pim mdt
   [edit routing-instances vpn-A protocols pim mdt]
   user@host# set group-range 227.0.0.0/8
   ```

2. Configure a data MDT-creation threshold for a multicast group and source.

   ```
   [edit routing-instances vpn-A protocols pim mdt]
   user@host# set threshold group 224.4.4.4 source 10.10.20.43 rate 10
   ```

3. Configure a tunnel limit.

   ```
   [edit routing-instances vpn-A protocols pim mdt]
   user@host# set tunnel-limit 10
   ```

4. If you are done configuring the device, commit the configuration.

```
[edit routing-instances vpn-A protocols pim mdt]
user@host# commit
```

Verification

To display information about the default MDT and any data MDTs for the VRF instance **vpn-A**, use the **show pim mdt instance ce1 detail** operational mode command. This command displays either the outgoing tunnels (the tunnels initiated by the local PE router), the incoming tunnels (tunnels initiated by the remote PE routers), or both.

To display the data MDT group addresses cached by PE routers that participate in the VRF instance **vpn-A**, use the **show pim mdt data-mdt-joins instance vpn-A** operational mode command. The command displays the information cached from MDT join TLV packets received by all PE routers participating in the specified VRF instance.

You can trace the operation of data MDTs by including the **mdt detail** flag in the **[edit protocols pim traceoptions]** configuration. When this flag is set, all the **mt** interface-related activity is logged in trace files.

See Also

- "*Introduction to Configuring Layer 3 VPNs*" in the *Junos OS VPNs Library for Routing Devices*

Example: Enabling Dynamic Reuse of Data MDT Group Addresses

This example describes how to enable dynamic reuse of data multicast distribution tree (MDT) group addresses.

Requirements

Before you begin:

- Configure the router interfaces. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 217.

Overview

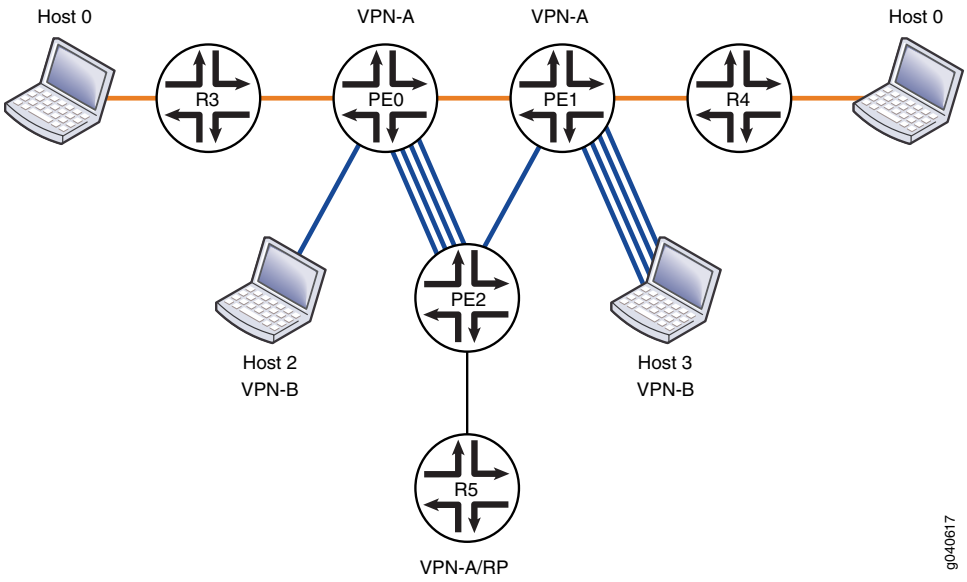
A limited number of multicast group addresses are available for use in data MDT tunnels. By default, when the available multicast group addresses are all used, no new data MDTs can be created.

You can enable dynamic reuse of data MDT group addresses. Dynamic reuse of data MDT group addresses allows multiple multicast streams to share a single MDT and multicast provider group address. For example, three streams can use the same provider group address and MDT tunnel.

The streams are assigned to a particular MDT in a round-robin fashion. Since a provider tunnel might be used by multiple customer streams, this can result in egress routers receiving customer traffic that is not destined for their attached customer sites. This example shows the plain PIM scenario, without the MVPN provider tunnel.

Figure 83 on page 518 shows the topology used in this example.

Figure 83: Dynamic Reuse of Data MDT Group Addresses



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement bgp-to-ospf term 1 from protocol bgp
set policy-options policy-statement bgp-to-ospf term 1 then accept
set protocols mpls interface all
set protocols bgp local-as 65520
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.38.17
```

```
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp neighbor 10.255.38.21
set protocols bgp group ibgp neighbor 10.255.38.15
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols pim rp static address 10.255.38.21
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim interface fxp0.0 disable
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface ge-1/1/2.0
set routing-instances VPN-A interface lo0.1
set routing-instances VPN-A route-distinguisher 10.0.0.10:04
set routing-instances VPN-A vrf-target target:100:10
set routing-instances VPN-A protocols ospf export bgp-to-ospf
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface all
set routing-instances VPN-A protocols pim traceoptions file pim-VPN-A.log
set routing-instances VPN-A protocols pim traceoptions file size 5m
set routing-instances VPN-A protocols pim traceoptions flag mdt detail
set routing-instances VPN-A protocols pim dense-groups 224.0.1.39/32
set routing-instances VPN-A protocols pim dense-groups 224.0.1.40/32
set routing-instances VPN-A protocols pim dense-groups 229.0.0.0/8
set routing-instances VPN-A protocols pim vpn-group-address 239.1.0.0
set routing-instances VPN-A protocols pim rp static address 10.255.38.15
set routing-instances VPN-A protocols pim interface lo0.1 mode sparse-dense
set routing-instances VPN-A protocols pim interface ge-1/1/2.0 mode sparse-dense
set routing-instances VPN-A protocols pim mdt threshold group 224.1.1.1/32 source
  192.168.255.245/32 rate 20
set routing-instances VPN-A protocols pim mdt threshold group 224.1.1.2/32 source
  192.168.255.245/32 rate 20
set routing-instances VPN-A protocols pim mdt threshold group 224.1.1.3/32 source
  192.168.255.245/32 rate 20
set routing-instances VPN-A protocols pim mdt data-mdt-reuse
set routing-instances VPN-A protocols pim mdt tunnel-limit 2
set routing-instances VPN-A protocols pim mdt group-range 239.1.1.0/30
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure dynamic reuse of data MDT group addresses:

1. Configure the **bgp-to-ospf** export policy.

   ```
   [edit policy-options policy-statement bgp-to-ospf]
   user@host# set term 1 from protocol bgp
   user@host# set term 1 then accept
   ```

2. Configure MPLS, LDP, BGP, OSPF, and PIM.

   ```
   [edit]
   user@host# edit protocols
   [edit protocols]
   ```

```
user@host# set mpls interface all
[edit protocols]
user@host# set ldp interface all
[edit protocols]
user@host# set bgp local-as 65520
[edit protocols]
user@host# set bgp group ibgp type internal
[edit protocols]
user@host# set bgp group ibgp local-address 10.255.38.17
[edit protocols]
user@host# set bgp group ibgp family inet-vpn unicast
[edit protocols]
user@host# set bgp group ibgp neighbor 10.255.38.21
[edit protocols]
user@host# set bgp group ibgp neighbor 10.255.38.15
[edit protocols]
user@host# set ospf traffic-engineering
[edit protocols]
user@host# set ospf area 0.0.0.0 interface all
[edit protocols]
user@host# set ospf area 0.0.0.0 interface fxp0.0 disable
[edit protocols]
user@host# set pim rp static address 10.255.38.21
[edit protocols]
user@host# set pim interface all mode sparse
[edit protocols]
user@host# set pim interface all version 2
[edit protocols]
user@host# set pim interface fxp0.0 disable
[edit protocols]
user@host# exit
```

3. Configure the routing instance, and apply the **bgp-to-ospf** export policy.

```
[edit]
user@host# edit routing-instances VPN-A
[edit routing-instances VPN-A]
user@host# set instance-type vrf
[edit routing-instances VPN-A]
user@host# set interface ge-1/1/2.0
[edit routing-instances VPN-A]
user@host# set interface lo0.1
[edit routing-instances VPN-A]
user@host# set route-distinguisher 10.0.0.10:04
[edit routing-instances VPN-A]
user@host# set vrf-target target:100:10
[edit routing-instances VPN-A]
user@host# set protocols ospf export bgp-to-ospf
[edit routing-instances VPN-A]
user@host# set protocols ospf area 0.0.0.0 interface all
```

4. Configure PIM trace operations for troubleshooting.

```
[edit routing-instances VPN-A]
user@host# set protocols pim traceoptions file pim-VPN-A.log
```

```
[edit routing-instances VPN-A]
user@host# set protocols pim traceoptions file size 5m
[edit routing-instances VPN-A]
user@host# set protocols pim traceoptions flag mdt detail
```

5. Configure the groups that operate in dense mode and the group address on which to encapsulate multicast traffic from the routing instance.

```
[edit routing-instances VPN-A]
user@host# set protocols pim dense-groups 224.0.1.39/32
[edit routing-instances VPN-A]
user@host# set protocols pim dense-groups 224.0.1.40/32
[edit routing-instances VPN-A]
user@host# set protocols pim dense-groups 229.0.0.0/8
[edit routing-instances VPN-A]
user@host# set protocols pim group-address 239.1.0.0
[edit routing-instances VPN-A]
```

6. Configure the address of the RP and the interfaces operating in sparse-dense mode.

```
[edit routing-instances VPN-A]
user@host# set protocols pim rp static address 10.255.38.15
[edit routing-instances VPN-A]
user@host# set protocols pim interface lo0.1 mode sparse-dense
[edit routing-instances VPN-A]
user@host# set protocols pim interface ge-1/1/2.0 mode sparse-dense
```

7. Configure the data MDT, including the **data-mdt-reuse** statement.

```
[edit routing-instances VPN-A]
user@host# set protocols pim mdt threshold group 224.1.1.1/32 source
  192.168.255.245/32 rate 20
[edit routing-instances VPN-A]
user@host# set protocols pim mdt threshold group 224.1.1.2/32 source
  192.168.255.245/32 rate 20
[edit routing-instances VPN-A]
user@host# set protocols pim mdt threshold group 224.1.1.3/32 source
  192.168.255.245/32 rate 20
[edit routing-instances VPN-A]
user@host# set protocols pim mdt data-mdt-reuse
[edit routing-instances VPN-A]
user@host# set protocols pim mdt tunnel-limit 2
[edit routing-instances VPN-A]
user@host# set protocols pim mdt group-range 239.1.1.0/30
```

8. If you are done configuring the device, commit the configuration.

```
[edit routing-instances VPN-A]
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show policy-options**, **show protocols**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
policy-statement bgp-to-ospf {
    term 1 {
        from protocol bgp;
        then accept;
    }
}

user@host# show protocols
mpls {
    interface all;
}
bgp {
    local-as 65520;
    group ibgp {
        type internal;
        local-address 10.255.38.17;
        family inet-vpn {
            unicast;
        }
        neighbor 10.255.38.21;
        neighbor 10.255.38.15;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all;
}
pim {
    rp {
        static {
            address 10.255.38.21;
        }
    }
    interface all {
        mode sparse;
        version 2;
    }
    interface fxp0.0 {
        disable;
```

```
    }
}

user@host# show routing-instances
VPN-A {
    instance-type vrf;
    interface ge-1/1/2.0;
    interface lo0.1;
    route-distinguisher 10.0.0.10:04;
    vrf-target target:100:10;
    protocols {
        ospf {
            export bgp-to-ospf;
            area 0.0.0.0 {
                interface all;
            }
        }
        pim {
            traceoptions {
                file pim-VPN-A.log size 5m;
                flag mdt detail;
            }
            dense-groups {
                224.0.1.39/32;
                224.0.1.40/32;
                229.0.0.0/8;
            }
            vpn-group-address 239.1.0.0;
            rp {
                static {
                    address 10.255.38.15;
                }
            }
            interface lo0.1 {
                mode sparse-dense;
            }
            interface ge-1/1/2.0 {
                mode sparse-dense;
            }
            mdt {
                threshold {
                    group 224.1.1.1/32 {
                        source 192.168.255.245/32 {
                            rate 20;
                        }
                    }
                    group 224.1.1.2/32 {
                        source 192.168.255.245/32 {
                            rate 20;
                        }
                    }
                    group 224.1.1.3/32 {
                        source 192.168.255.245/32 {
                            rate 20;
                        }
                    }
```

```
            }
            data-mdt-reuse;
            tunnel-limit 2;
            group-range 239.1.1.0/30;
        }
    }
  }
}
```

Verification

To verify the configuration, run the following commands:

- show pim join instance VPN-A extensive
- show multicast route instance VPN-A extensive
- show pim mdt instance VPN-A
- show pim mdt data-mdt-joins instance VPN-A

See Also

- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

Related Documentation

- Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs on page 448
- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 489

CHAPTER 21

Configuring Next-Generation Multicast VPNs

Multiprotocol BGP MVPNs Overview

Comparison of Draft Rosen Multicast VPNs and Next-Generation Multiprotocol BGP Multicast VPNs

There are several multicast applications driving the deployment of next-generation Layer 3 multicast VPNs (MVPNs). Some of the key emerging applications include the following:

- Layer 3 VPN multicast service offered by service providers to enterprise customers
- Video transport applications for wholesale IPTV and multiple content providers attached to the same network
- Distribution of media-rich financial services or enterprise multicast services
- Multicast backhaul over a metro network

There are two ways to implement Layer 3 MVPNs. They are often referred to as dual PIM MVPNs (also known as “draft-rosen”) and multiprotocol BGP (MBGP)-based MVPNs (the “next generation” method of MVPN configuration). Both methods are supported and equally effective. The main difference is that the MBGP-based MVPN method does not require multicast configuration on the service provider backbone. Multiprotocol BGP multicast VPNs employ the intra-autonomous system (AS) next-generation BGP control plane and PIM sparse mode as the data plane. The PIM state information is maintained between the PE routers using the same architecture that is used for unicast VPNs. The main advantage of deploying MVPNs with MBGP is simplicity of configuration and operation because multicast is not needed on the service provider VPN backbone connecting the PE routers.

Using the draft-rosen approach, service providers might experience control and data plane scaling issues associated with the maintenance of two routing and forwarding mechanisms: one for VPN unicast and one for VPN multicast. For more information on the limitations of Draft Rosen, see draft-rekhter-mboned-mvpn-deploy.

See Also

- MBGP Multicast VPN Sites on page 527

MBGP Multicast VPN Sites

The main characteristics of MBGP MVPNs are:

- They extend Layer 3 VPN service (RFC 4364) to support IP multicast for Layer 3 VPN service providers.
- They follow the same architecture as specified by RFC 4364 for unicast VPNs. Specifically, BGP is used as the provider edge (PE) router-to-PE router control plane for multicast VPN.
- They eliminate the requirement for the virtual router (VR) model (as specified in Internet draft draft-rosen-vpn-mcast, *Multicast in MPLS/BGP VPNs*) for multicast VPNs and the RFC 4364 model for unicast VPNs.
- They rely on RFC 4364-based unicast with extensions for intra-AS and inter-AS communication.

An MBGP MVPN defines two types of site sets, a sender site set and a receiver site set. These sites have the following properties:

- Hosts within the sender site set can originate multicast traffic for receivers in the receiver site set.
- Receivers outside the receiver site set should not be able to receive this traffic.
- Hosts within the receiver site set can receive multicast traffic originated by any host in the sender site set.
- Hosts within the receiver site set should not be able to receive multicast traffic originated by any host that is not in the sender site set.

A site can be in both the sender site set and the receiver site set, so hosts within such a site can both originate and receive multicast traffic. For example, the sender site set could be the same as the receiver site set, in which case all sites could both originate and receive multicast traffic from one another.

Sites within a given MBGP MVPN might be within the same organization or in different organizations, which means that an MBGP MVPN can be either an intranet or an extranet. A given site can be in more than one MBGP MVPN, so MBGP MVPNs might overlap. Not all sites of a given MBGP MVPN have to be connected to the same service provider, meaning that an MBGP MVPN can span multiple service providers.

Feature parity for the MVPN extranet functionality or overlapping MVPNs on the Junos Trio chipset is supported in Junos OS Releases 11.1R2, 11.2R2, and 11.4.

Another way to look at an MBGP MVPN is to say that an MBGP MVPN is defined by a set of administrative policies. These policies determine both the sender site set and the receiver site set. These policies are established by MBGP MVPN customers, but implemented by service providers using the existing BGP and MPLS VPN infrastructure.

See Also

- Example: Allowing MBGP MVPN Remote Sources on page 633
- Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN on page 624

Multicast VPN Standards

MBGP MVPNs are defined in the following IETF Internet drafts:

- Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-03.txt, *BGP Encodings for Multicast in MPLS/BGP IP VPNs*
- Internet draft draft-ietf-l3vpn-2547bis-mcast-02.txt, *Multicast in MPLS/BGP IP VPNs*

PIM Sparse Mode, PIM Dense Mode, Auto-RP, and BSR for MBGP MVPNs

You can configure PIM sparse mode, PIM dense mode, auto-RP, and bootstrap router (BSR) for MBGP MVPN networks:

- PIM sparse mode—Allows a router to use any unicast routing protocol and performs reverse-path forwarding (RPF) checks using the unicast routing table. PIM sparse mode includes an explicit join message, so routers determine where the interested receivers are and send join messages upstream to their neighbors, building trees from the receivers to the rendezvous point (RP).
- PIM dense mode—Allows a router to use any unicast routing protocol and performs reverse-path forwarding (RPF) checks using the unicast routing table. Packets are forwarded to all interfaces except the incoming interface. Unlike PIM sparse mode, where explicit joins are required for packets to be transmitted downstream, packets are flooded to all routers in the routing instance in PIM dense mode.
- Auto-RP—Uses PIM dense mode to propagate control messages and establish RP mapping. You can configure an auto-RP node in one of three different modes: discovery mode, announce mode, and mapping mode.
- BSR—Establishes RPs. A selected router in a network acts as a BSR, which selects a unique RP for different group ranges. BSR messages are flooded using a data tunnel between PE routers.

See Also
- Example: Allowing MBGP MVPN Remote Sources on page 633
- Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN on page 624

MBGP-Based Multicast VPN Trees

MBGP-based MVPNs (next-generation MVPNs) are based on Internet drafts and extend unicast VPNs based on RFC 2547 to include support for IP multicast traffic. These MVPNs follow the same architectural model as the unicast VPNs and use BGP as the provider edge (PE)-to-PE control plane to exchange information. The next generation MVPN approach is based on Internet drafts draft-ietf-l3vpn-2547bis-mcast.txt, draft-ietf-l3vpn-2547bis-mcast-bgp.txt, and draft-morin-l3vpn-mvpn-considerations.txt.

MBGP-based MVPNs introduce two new types of tree:

Inclusive tree—A single multicast distribution tree in the backbone carrying all the multicast traffic from a specified set of one or more MVPNs. An inclusive tree carrying

the traffic of more than one MVPN is an *aggregate inclusive tree*. All the PEs that attach to MVPN receiver sites using the tree belong to that inclusive tree.

Selective tree—A single multicast distribution tree in the backbone carrying traffic for a specified set of one or more multicast groups. When multicast groups belonging to more than one MVPN are on the tree, it is called an *aggregate selective tree*.

By default, traffic from most multicast groups can be carried by an inclusive tree, while traffic from some groups (for example, high bandwidth groups) can be carried by one of the selective trees. Selective trees, if they contain only those PEs that need to receive multicast data from one or more groups assigned to the tree, can provide more optimal routing than inclusive trees alone, although this requires more state information in the P routers.

An MPLS-based VPN running BGP with autodiscovery is used as the basis for a next-generation MVPN. The autodiscovered route information is carried in MBGP network layer reachability information (NLRI) updates for multicast VPNs (MCAST-VPNs). These MCAST-VPN NLRIs are handled in the same way as IPv4 routes: route distinguishers are used to distinguish between different VPNs in the network. These NLRIs are imported and exported based on the route target extended communities, just as IPv4 unicast routes. In other words, existing BGP mechanisms are used to distribute multicast information on the provider backbone without requiring multicast directly.

For example, consider a customer running Protocol-Independent Multicast (PIM) sparse mode in source-specific multicast (SSM) mode. Only source tree join customer multicast (c-multicast) routes are required. (PIM sparse mode in anysource multicast (ASM) mode can be supported with a few enhancements to SSM mode.)

The customer multicast route carrying a particular multicast source S needs to be imported only into the VPN routing and forwarding (VRF) table on the PE router connected to the site that contains the source S and not into any other VRF, even for the same MVPN. To do this, each VRF on a particular PE has a distinct VRF route import extended community associated with it. This community consists of the PE router's IP address and local PE number. Different MVPNs on a particular PE have different route imports, and for a particular MVPN, the VRF instances on different PE routers have different route imports. This VRF route import is auto-configured and not controlled by the user.

Also, all the VRFs within a particular MVPN will have information about VRF route imports for each VRF. This is accomplished by “piggybacking” the VRF route import extended community onto the unicast VPN IPv4 routes. To make sure a customer multicast route carrying multicast source S is imported only into the VRF on the PE router connected to the site contained the source S, it is necessary to find the unicast VPN IPv4 route to S and set the route target of the customer multicast route to the VRF import route carried by the VPN IPv4 route just found.

The process of originating customer multicast routes in an MBGP-based MVPN is shown in Figure 84 on page 530.

In the figure, an MVPN has three receiver sites (R1, R2, and R3) and one source site (S). The site routers are connected to four PE routers, and PIM is running between the PE

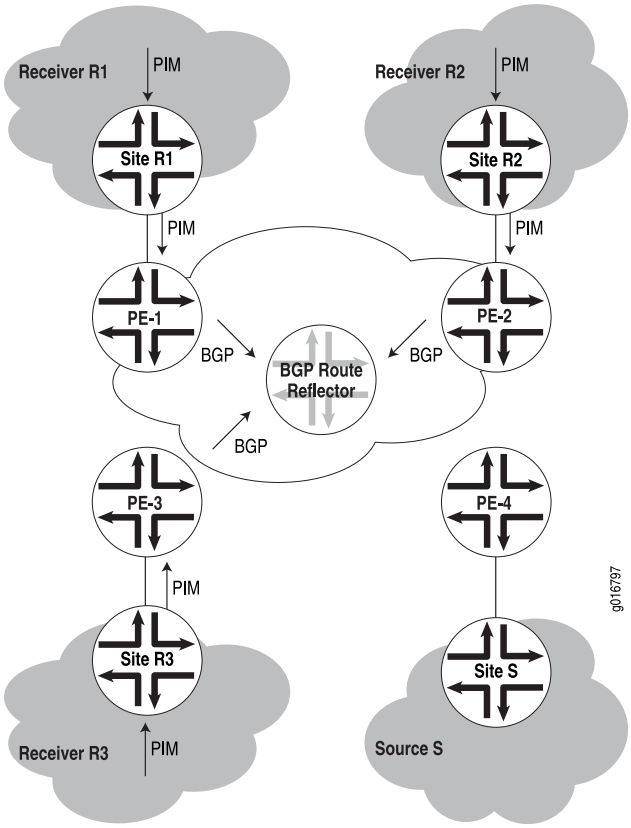
routers and the site routers. However, only BGP runs between the PE routers on the provider's network.

When router PE-1 receives a PIM join message for (S,G) from site router R1, this means that site R1 has one or more receivers for a given source and multicast group (S,G) combination. In that case, router PE-1 constructs and originates a customer multicast route after doing three things:

1. Finding the unicast VPN IPv4 router to source S
2. Extracting the route distinguisher and VRF route import form this route
3. Putting the (S,G) information from the PIM join, the router distinguisher from the VPN IPv4 route, and the route target from the VRF route import of the VPN IPv4 route into a MBGP update

The update is distributed around the VPN through normal BGP mechanisms such as router reflectors.

Figure 84: Source and Receiver Sites in an MVPN



What happens when the source site S receives the MBGP information is shown in Figure 85 on page 531. In the figure, the customer multicast route information is distributed by the BGP route reflector as an MBGP update.

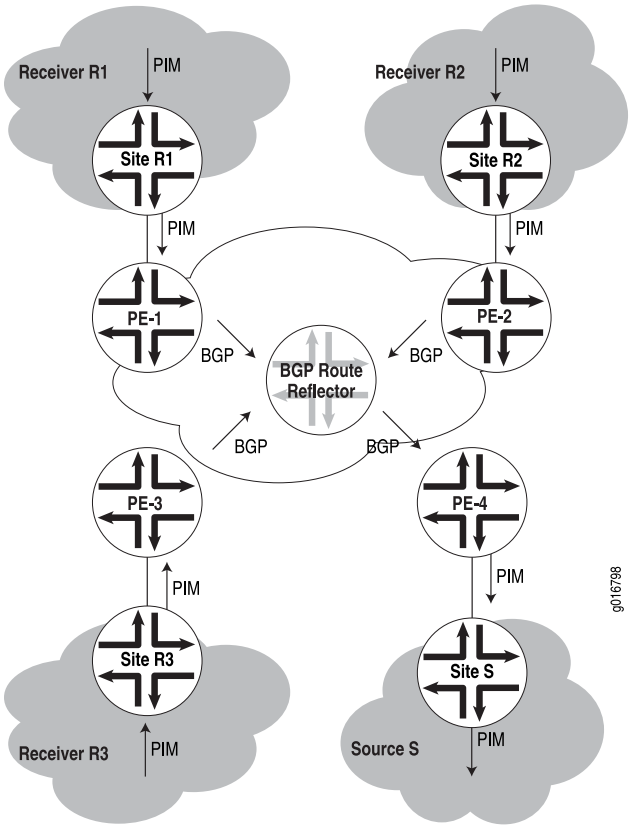
The provider router PE-4 will then:

1. Receive the customer multicast route originated by the PE routers and aggregated by the route reflector.

2. Accept the customer multicast route into the VRF for the correct MVPN (because the VRF route import matches the route target carried in the customer multicast route information).

3. Create the proper (S,G) state in the VRF and propagate the information to the customer routers of source site S using PIM.

Figure 85: Adding a Receiver to an MVPN Source Site Using MBGP



See Also

- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 449

Release History Table

Release	Description
11.1R2	Feature parity for the MVPN extranet functionality or overlapping MVPNs on the Junos Trio chipset is supported in Junos OS Releases 11.1R2, 11.2R2, and 11.4.

Related Documentation

- Configuring Multiprotocol BGP Multicast VPNs on page 584

Understanding Next-Generation MVPN Network Topology

Layer 3 BGP-MPLS virtual private networks (VPNs) are widely deployed in today's networks worldwide. Multicast applications, such as IPTV, are rapidly gaining popularity as is the number of networks with multiple, media-rich services merging over a shared Multiprotocol Label Switching (MPLS) infrastructure. The demand for delivering multicast service across a BGP-MPLS infrastructure in a scalable and reliable way is also increasing.

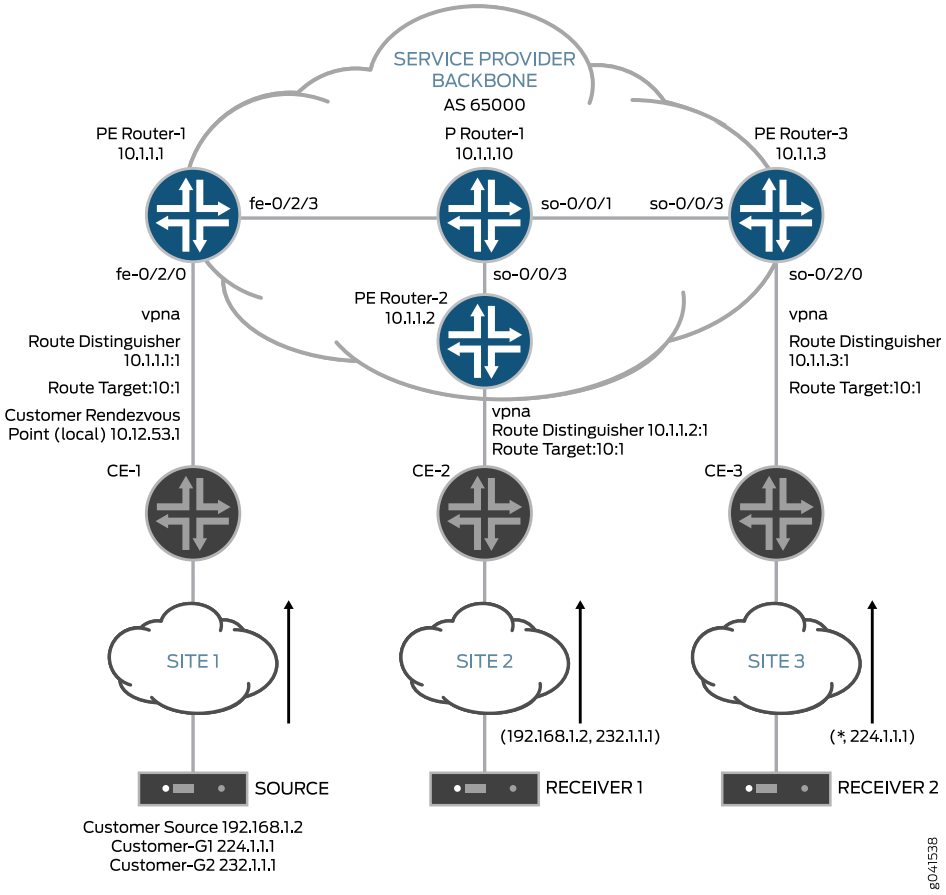
RFC 4364 describes protocols and procedures for building unicast BGP-MPLS VPNs. However, there is no framework specified in the RFC for provisioning multicast VPN (MVPN) services. In the past, Multiprotocol Label Switching Virtual Private Network (MVPN) traffic was overlaid on top of a BGP-MPLS network using a virtual LAN model based on Draft Rosen. Using the Draft Rosen approach, service providers were faced with control and data plane scaling issues of an overlay model and the maintenance of two routing/forwarding mechanisms: one for VPN unicast service and one for VPN multicast service. For more information about the limitations of Draft Rosen, see draft-rekhter-mboned-mvpn-deploy.

As a result, the IETF Layer 3 VPN working group published an Internet draft draft-ietf-l3vpn-2547bis-mcast-10.txt, *Multicast in MPLS/BGP IP VPNs*, that outlines a different architecture for next-generation MVPNs, as well as an accompanying RFC 2547 that proposes a BGP control plane for MVPNs. In turn, Juniper Networks delivered the industry's first implementation of BGP next-generation MVPNs in 2007.

All examples in this document refer to the network topology shown in Figure 86 on page 533:

- The service provider in this example offers VPN unicast and multicast services to Customer A (vpna).
- The VPN multicast source is connected to Site 1 and transmits data to groups 232.1.1.1 and 224.1.1.1.
- VPN multicast receivers are connected to Site 2 and Site 3.
- The provider edge router 1 (Router PE1) VRF table acts as the C-RP (using address 10.12.53.1) for C-PIM-SM ASM groups.
- The service provider uses RSVP-TE point-to-multipoint LSPs for transmitting VPN multicast data across the network.

Figure 86: Next-Generation MVPN Topology



Related Documentation

- Understanding Next-Generation MVPN Concepts and Terminology on page 533
- Understanding Next-Generation MVPN Control Plane on page 536
- Next-Generation MVPN Data Plane Overview on page 556
- Example: Configuring MBGP Multicast VPNs on page 606

Understanding Next-Generation MVPN Concepts and Terminology

This section includes background material about how next-generation MVPNs work.

Route Distinguisher and VRF Route Target Extended Community

Route distinguisher and VPN routing and forwarding (VRF) route target extended communities are an integral part of unicast BGP-MPLS virtual private networks (VPNs). Route distinguisher and route target are often confused in terms of their purpose in BGP-MPLS networks. As they play an important role in BGP next-generation MVPNs, it is important to understand what they are and how they are used as described in RFC 4364.

RFC 4364 describes the purpose of route distinguisher as the following:

"A VPN-IPv4 address is a 12-byte quantity, beginning with an 8-byte Route Distinguisher (RD) and ending with a 4-byte IPv4 address. If several VPNs use the same IPv4 address prefix, the PEs translate these into unique VPN-IPv4 address prefixes. This ensures that if the same address is used in several different VPNs, it is possible for BGP to carry several completely different routes to that address, one for each VPN."

Typically, each VRF table on a provider edge (PE) router is configured with a unique route distinguisher. Depending on the routing design, the route distinguisher can be unique or the same for a given VRF on other PE routers. A route distinguisher is an 8-byte number with two fields. The first field can be either an AS number (2 or 4 bytes) or an IP address (4 bytes). The second field is assigned by the user.

RFC 4364 describes the purpose of a VRF route target extended community as the following:

"Every VRF is associated with one or more Route Target (RT) attributes.

When a VPN-IPv4 route is created (from an IPv4 route that the PE router has learned from a CE) by a PE router, it is associated with one or more route target attributes. These are carried in BGP as attributes of the route.

Any route associated with Route Target T must be distributed to every PE router that has a VRF associated with Route Target T. When such a route is received by a PE router, it is eligible to be installed in those of the PE's VRFs that are associated with Route Target T."

The route target also contains two fields and is structured similar to a route distinguisher. The first field of the route target is either an AS number (2 or 4 bytes) or an IP address (4 bytes), and the second field is assigned by the user. Each PE router advertises its VPN-IPv4 routes with the route target (as one of the BGP path attributes) configured for the VRF table. The route target attached to the advertised route is referred to as the export route target. On the receiving PE router, the route target attached to the route is compared to the route target configured for the local VRF tables. The locally configured route target that is used in deciding whether a VPN-IPv4 route should be installed in a VRF table is referred to as the import route target.

C-Multicast Routing

Customer multicast (C-multicast) routing information exchange refers to the distribution of customer PIM (C-PIM) join/prune messages received from local customer edge (CE) routers to other PE routers (toward the VPN multicast source).

BGP MVPNs

BGP MVPNs use BGP as the control plane protocol between PE routers for MVPNs, including the exchange of C-multicast routing information. The support of BGP as a PE-PE protocol for exchanging C-multicast routes is mandated by Internet draft draft-ietf-l3vpn-mvpn-considerations-06.txt, *Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution*. The use of BGP for distributing C-multicast routing information is closely modeled after its highly successful counterpart of VPN unicast route distribution. Using BGP as the control plane protocol allows service providers to take advantage of

this widely deployed, feature-rich protocol. It also enables service providers to leverage their knowledge and investment in managing BGP-MPLS VPN unicast service to offer VPN multicast services.

Sender and Receiver Site Sets

Internet draft draft-ietf-l3vpn-2547bis-mcast-10.txt describes an MVPN as a set of administrative policies that determine the PE routers that are in sender and receiver site sets.

A PE router can be a sender, a receiver, or both a sender and a receiver, depending on the configuration:

- A sender site set includes PE routers with local VPN multicast sources (VPN customer multicast sources either directly connected or connected via a CE router). A PE router that is in the sender site set is the sender PE router.
- A receiver site set includes PE routers that have local VPN multicast receivers. A PE router that is in the receiver site set is the receiver PE router.

Provider Tunnels

Internet draft draft-ietf-l3vpn-2547bis-mcast-10.txt defines provider tunnels as the transport mechanisms used for forwarding VPN multicast traffic across service provider networks. Different tunneling technologies, such as generic routing encapsulation (GRE) and MPLS, can be used to create provider tunnels. Provider tunnels can be signaled by a variety of signaling protocols. This topic describes only PIM-SM (ASM) signaled IP GRE provider tunnels and RSVP-Traffic Engineering (RSVP-TE) signaled MPLS provider tunnels.

In BGP MVPNs, the sender PE router distributes information about the provider tunnel in a BGP attribute called provider multicast service interface (PMSI). By default, all receiver PE routers join and become the leaves of the provider tunnel rooted at the sender PE router.

Provider tunnels can be inclusive or selective:

- An inclusive provider tunnel (I-PMSI provider tunnel) enables a PE router that is in the sender site set of an MVPN to transmit multicast data to all PE routers that are members of that MVPN.
- A selective provider tunnel (S-PMSI provider tunnel) enables a PE router that is in the sender site set of an MVPN to transmit multicast data to a subset of the PE routers.

Related Documentation

- Understanding Next-Generation MVPN Network Topology on page 532
- Generating Next-Generation MVPN VRF Import and Export Policies Overview on page 563
- Exchanging C-Multicast Routes on page 550
- Example: Configuring MBGP Multicast VPNs on page 606

Understanding Next-Generation MVPN Control Plane

The BGP next-generation multicast virtual private network (MVPN) control plane, as specified in Internet draft draft-ietf-l3vpn-2547bis-mcast-10.txt and Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt, distributes all the necessary information to enable end-to-end C-multicast routing exchange via BGP. The main tasks of the control plane (Table 20 on page 536) include MVPN autodiscovery, distribution of provider tunnel information, and PE-PE C-multicast route exchange.

Table 20: Next-Generation MVPN Control Plane Tasks

Control Plane Task	Description
MVPN autodiscovery	A provider edge (PE) router discovers the identity of the other PE routers that participate in the same MVPN.
Distribution of provider tunnel information	A sender PE router advertises the type and identifier of the provider tunnel that it will use to transmit VPN multicast packets.
PE-PE C-Multicast route exchange	A receiver PE router propagates C-multicast join messages (C-joins) received over its VPN interface toward the VPN multicast sources.

BGP MCAST-VPN Address Family and Route Types

Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt introduced a BGP address family called MCAST-VPN for supporting next-generation MVPN control plane operations. The new address family is assigned the subsequent address family identifier (SAFI) of 5 by the Internet Assigned Numbers Authority (IANA).

A PE router that participates in a BGP-based next-generation MVPN network is required to send a BGP update message that contains MCAST-VPN network layer reachability information (NLRI). An MCAST-VPN NLRI contains route type, length, and variable fields. The value of each variable field depends on the route type.

Seven types of next-generation MVPN BGP routes (also referred to as routes in this topic) are specified (Table 21 on page 537). The first five route types are called autodiscovery MVPN routes. This topic also refers to Type 1-5 routes as non-C-multicast MVPN routes. Type 6 and Type 7 routes are called C-multicast MVPN routes.

Table 21: Next-generation MVPN BGP Route Types

Usage	Type	Name	Description
Membership autodiscovery routes for inclusive provider tunnels	1	Intra autonomous system (intra-AS) I-PMSI autodiscovery route	• Originated by all next-generation MVPN PE routers. • Used for advertising and learning intra autonomous system (intra-AS) MVPN membership information.
	2	Inter-AS I-PMSI AD route	• Originated by next-generation MVPN ASBR routers. • Used for advertising and learning inter-AS MVPN membership information.
Autodiscovery routes for selective provider tunnels	3	S-PMSI AD route	• Originated by a sender router. • Used for initiating a selective provider tunnel for a particular (C-S, C-G).
	4	Leaf AD route	• Originated by receiver PE routers in response to receiving a Type 3 route. • Used by a sender PE router to discover the leaves of a selective provider tunnel. • Also used for inter-AS operations that are not covered in this topic.
VPN multicast source discovery routes	5	Source active AD route	• Originated by the PE router that discovers an active VPN multicast source. • Used by PE routers to learn the identity of active VPN multicast sources.
C-Multicast routes	6	Shared tree join route	• Originated by receiver PE routers. • Originated when a PE router receives a shared tree C-join (C-*, C-G) through its PE-CE interface.
	7	Source tree join route	• Originated by receiver PE routers. • Originated when a PE router receives a source tree C-join (C-S, C-G) or originated by the PE router that already has a Type 6 route and receives a Type 5 route.

Intra-AS MVPN Membership Discovery (Type 1 Routes)

All next-generation MVPN PE routers create and advertise a Type 1 intra-AS autodiscovery route (Figure 87 on page 538) for each MVPN to which they are connected. Table 22 on page 538 describes the format of each MVPN Type 1 intra-AS autodiscovery route.

Figure 87: Intra-AS I-PMSI AD Route Type MCAST-VPN NLRI Format

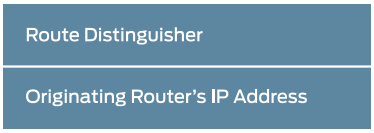


Table 22: Type 1 Intra-AS Autodiscovery Route MVPN Format Descriptions

Field	Description
Route Distinguisher	Set to the route distinguisher configured for the VPN.
Originating Router's IP Address	Set to the IP address of the router originating this route. The address is typically the primary loopback address of the PE router.

Inter-AS MVPN Membership Discovery (Type 2 Routes)

Type 2 routes are used for membership discovery between PE routers that belong to different autonomous systems (ASs). Their use is not covered in this topic.

Selective Provider Tunnels (Type 3 and Type 4 Routes)

A sender PE router that initiates a selective provider tunnel is required to originate a Type 3 intra-AS S-PMSI autodiscovery route with the appropriate PMSI attribute.

A receiver PE router responds to a Type 3 route by originating a Type 4 leaf autodiscovery route if it has local receivers interested in the traffic transmitted on the selective provider tunnel. Type 4 routes inform the sender PE router of the leaf PE routers.

Source Active Autodiscovery Routes (Type 5 Routes)

Type 5 routes carry information about active VPN sources and the groups to which they are transmitting data. These routes can be generated by any PE router that becomes aware of an active source. Type 5 routes apply only for PIM-SM (ASM) when intersite source-tree-only mode is being used.

C-Multicast Route Exchange (Type 6 and Type 7 Routes)

The C-multicast route exchange between PE routers refers to the propagation of C-joins from receiver PE routers to the sender PE routers.

In a next-generation MVPN, C-joins are translated into (or encoded as) BGP C-multicast MVPN routes and advertised via the BGP MCAST-VPN address family toward the sender PE routers.

Two types of C-multicast MVPN routes are specified:

- Type 6 C-multicast routes are used in representing information contained in a shared tree (C-*, C-G) join.
- Type 7 C-multicast routes are used in representing information contained in a source tree (C-S, C-G) join.

PMSI Attribute

The provider multicast service interface (PMSI) attribute (Figure 88 on page 539) carries information about the provider tunnel. In a next-generation MVPN network, the sender PE router sets up the provider tunnel, and therefore is responsible for originating the PMSI attribute. The PMSI attribute can be attached to Type 1, Type 2, or Type 3 routes. Table 23 on page 539 describes each PMSI attribute format.

Figure 88: PMSI Tunnel Attribute Format

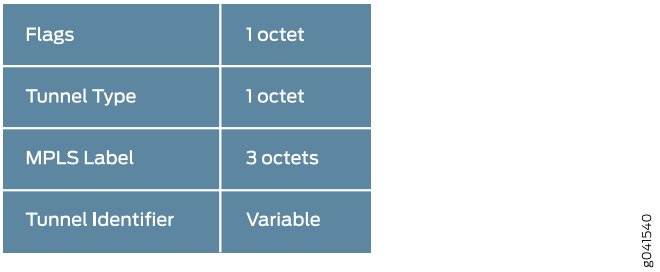


Table 23: PMSI Tunnel Attribute Format Descriptions

Field	Description
Flags	Currently has only one flag specified: Leaf Information Required. This flag is used for S-PMSI provider tunnel setup.
Tunnel Type	Identifies the tunnel technology used by the sender. Currently there are seven types of tunnels supported.
MPLS Label	Used when the sender PE router allocates the MPLS labels (also called upstream label allocation). This technique is described in RFC 5331 and is outside the scope of this topic.
Tunnel Identifier	Uniquely identifies the tunnel. Its value depends on the value set in the tunnel type field.

For example, Router PE1 originates the following PMSI attribute:

PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:6574:10.1.1.1]

VRF Route Import and Source AS Extended Communities

Two extended communities are specified to support next-generation MVPNs: source AS (**src-as**) and VRF route import extended communities.

The source AS extended community is an AS-specific extended community that identifies the AS from which a route originates. This community is mostly used for inter-AS operations, which is not covered in this topic.

The VPN routing and forwarding (VRF) route import extended community is an IP-address-specific extended community that is used for importing C-multicast routes in the VRF table of the active sender PE router to which the source is attached.

Each PE router creates a unique route target import and src-as community for each VPN and attaches them to the VPN-IPv4 routes.

Related Documentation

- Next-Generation MVPN Data Plane Overview on page 556
- Distributing C-Multicast Routes Overview on page 546
- Enabling Next-Generation MVPN Services on page 560
- Signaling Provider Tunnels and Data Plane Setup on page 570
- Originating Type 1 Intra-AS Autodiscovery Routes Overview on page 567
- Understanding Next-Generation MVPN Network Topology on page 532

Understanding Redundant Virtual Tunnel Interfaces in MBGP MVPNs

In multiprotocol BGP (MBGP) multicast VPNs (MVPNs), VT interfaces are needed for multicast traffic on routing devices that function as combined provider edge (PE) and provider core (P) routers to optimize bandwidth usage on core links. VT interfaces prevent traffic replication when a P router also acts as a PE router (an exit point for multicast traffic).

Starting in Junos OS Release 12.3, you can configure up to eight VT interfaces in a routing instance, thus providing Tunnel PIC redundancy inside the same multicast VPN routing instance. When the active VT interface fails, the secondary one takes over, and you can continue managing multicast traffic with no duplication.

Redundant VT interfaces are supported with RSVP point-to-multipoint provider tunnels as well as multicast LDP provider tunnels. This feature also works for extranets.

You can configure one of the VT interfaces to be the primary interface. If a VT interface is configured as the primary, it becomes the next hop that is used for traffic coming in from the core on the label-switched path (LSP) into the routing instance. When a VT interface is configured to be primary and the VT interface is used for both unicast and multicast traffic, only the multicast traffic is affected.

If no VT interface is configured to be the primary or if the primary VT interface is unusable, one of the usable configured VT interfaces is chosen to be the next hop that is used for

traffic coming in from the core on the LSP into the routing instance. If the VT interface in use goes down for any reason, another usable configured VT interface in the routing instance is chosen. When the VT interface in use changes, all multicast routes in the instance also switch their reverse-path forwarding (RPF) interface to the new VT interface to allow the traffic to be received.

To realize the full benefit of redundancy, we recommend that when you configure multiple VT interfaces, at least one of the VT interfaces be on a different Tunnel PIC from the other VT interfaces. However, Junos OS does not enforce this.

Release History Table

Release	Description
12.3	Starting in Junos OS Release 12.3, you can configure up to eight VT interfaces in a routing instance, thus providing Tunnel PIC redundancy inside the same multicast VPN routing instance.

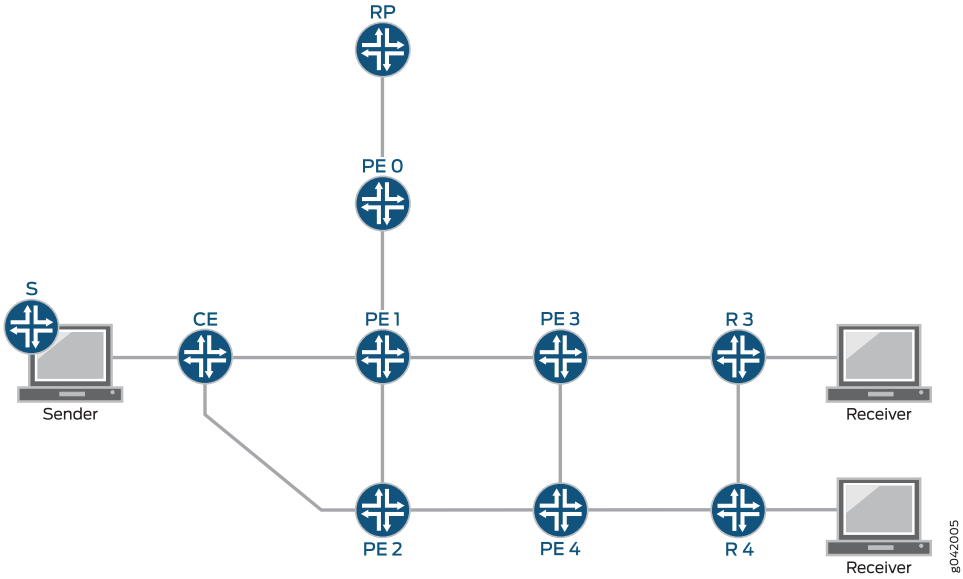
Related Documentation

- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 744

Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels

In a BGP multicast VPN (MVPN) (also called a multiprotocol BGP next-generation multicast VPN), sender-based reverse-path forwarding (RPF) helps to prevent multiple provider edge (PE) routers from sending traffic into the core, thus preventing duplicate traffic being sent to a customer. In the following diagram, sender-based RPF configured on egress Device PE3 and Device PE4 prevents duplicate traffic from being sent to the customers.

Figure 89: Sender-Based RPF



Sender-based RPF is supported on MX Series platforms with MPC line cards. As a prerequisite, the router must be set to **network-services enhanced-ip** mode.

Sender-based RPF (and hot-root standby) are supported only for MPLS BGP MVPNs with RSVP point-to-multipoint provider tunnels. Both SPT-only and SPT-RPT MVPN modes are supported.

Sender-based RPF does not work when point-to-multipoint provider tunnels are used with label-switched interfaces (LSI). Junos OS only allocates a single LSI label for each VRF, and uses this label for all point-to-multipoint tunnels. Therefore, the label that the egress receives does not indicate the sending PE router. LSI labels currently cannot scale to create a unique label for each point-to-multipoint tunnel. As such, virtual tunnel interfaces (vt) must be used for sender-based RPF functionality with point-to-multipoint provider tunnels.

Optionally, LSI interfaces can continue to be used for unicast purposes, and virtual tunnel interfaces can be configured to be used for multicast only.

In general, it is important to avoid (or recover from) having multiple PE routers send duplicate traffic into the core because this can result in duplicate traffic being sent to

the customer. The sender-based RPF has a use case that is limited to BGP MVPNs. The use-case scope is limited for the following reasons:

- A traditional RPF check for native PIM is based on the incoming interface. This RPF check prevents loops but does not prevent multiple forwarders on a LAN. The traditional RPF has been used because current multicast protocols either avoid duplicates on a LAN or have data-driven events to resolve the duplicates once they are detected.
- In PIM sparse mode, duplicates can occur on a LAN in normal protocol operation. The protocol has a data-driven mechanism (PIM assert messages) to detect duplication when it happens and resolve it.
- In PIM bidirectional mode, a designated forwarder (DF) election is performed on all LANs to avoid duplication.
- Draft Rosen MVPNs use the PIM assert mechanism because with Draft Rosen MVPNs the core network is analogous to a LAN.

Sender-based RPF is a solution to be used in conjunction with BGP MVPNs because BGP MVPNs use an alternative to data-driven-event solutions and bidirectional mode DF election. This is so, because, for one thing, the core network is not exactly a LAN. In an MVPN scenario, it is possible to determine which PE router has sent the traffic. Junos OS uses this information to only forward the traffic if it is sent from the correct PE router. With sender-based RPF, the RPF check is enhanced to check whether data arrived on the correct incoming virtual tunnel (vt-) interface and that the data was sent from the correct upstream PE router.

More specifically, the data must arrive with the correct MPLS label in the outer header used to encapsulate data through the core. The label identifies the tunnel and, if the tunnel is point-to-multipoint, the upstream PE router.

Sender-based RPF is not a replacement for single-forwarder election, but is a complementary feature. Configuring a higher primary loopback address (or router ID) on one PE device (PE1) than on another (PE2) ensures that PE1 is the single-forwarder election winner. The unicast-umh-election statement causes the unicast route preference to determine the single-forwarder election. If single-forwarder election is not used or if it is not sufficient to prevent duplicates in the core, sender-based RPF is recommended.

For RSVP point-to-multipoint provider tunnels, the transport label identifies the sending PE router because it is a requirement that penultimate hop popping (PHP) is disabled when using point-to-multipoint provider tunnels with MVPNs. PHP is disabled by default when you configure the MVPN protocol in a routing instance. The label identifies the tunnel, and (because the RSVP-TE tunnel is point-to-multipoint) the sending PE router.

The sender-based RPF mechanism is described in RFC 6513, *Multicast in MPLS/BGP IP VPNs* in section 9.1.1.

NOTE: The hot-root standby technique described in Internet draft draft-morin-l3vpn-mvpn-fast-failover-05 *Multicast VPN fast upstream failover* is an egress PE router functionality in which the egress PE router sends source-tree c-multicast join message to both a primary and a backup upstream PE router. This allows multiple copies of the traffic to flow through the provider core to the egress PE router. Sender-based RPF and hot-root standby can be used together to support *live-live* BGP MVPN traffic. This is a multicast-over-MPLS scheme for carrying mission-critical professional broadcast TV and IPTV traffic. A key requirement for many of these deployments is to have full redundancy of network equipment, including the ingress and egress PE routers. In some cases, a live-live approach is required, meaning that two duplicate traffic flows are sent across the network following diverse paths. When this technique is combined with sender-based forwarding, the two live flows of traffic are received at the egress PE router, and the egress PE router forwards a single stream to the customer network. Any failure in the network can be repaired locally at the egress PE router. For more information about hot-root standby, see hot-root-standby.

Sender-based RPF prevents duplicates from being sent to the customer even if there is duplication in the provider network. Duplication could exist in the provider because of a hot-root standby configuration or if the single-forwarder election is not sufficient to prevent duplicates. Single-forwarder election is used to prevent duplicates to the core network, while sender-based RPF prevents duplicates to the customer even if there are duplicates in the core. There are cases in which single-forwarder election cannot prevent duplicate traffic from arriving at the egress PE router. One example of this (outlined in section 9.3.1 of RFC 6513) is when PIM sparse mode is configured in the customer network and the MVPN is in RPT-SPT mode with an I-PMSI.

Determining the Upstream PE Router

After Junos OS chooses the ingress PE router, the sender-based RPF decision determines whether the correct ingress PE router is selected. As described in RFC 6513, section 9.1.1, an egress PE router, PE1, chooses a specific upstream PE router, for given (C-S,C-G). When PE1 receives a (C-S,C-G) packet from a PMSI, it might be able to identify the PE router that transmitted the packet onto the PMSI. If that transmitter is other than the PE router selected by PE1 as the upstream PE router, PE1 can drop the packet. This means that the PE router detects a duplicate, but the duplicate is not forwarded.

When an egress PE router generates a type 7 C-multicast route, it uses the VRF route import extended community carried in the VPN-IP route toward the source to construct the route target carried by the C-multicast route. This route target results in the C-multicast route being sent to the upstream PE router, and being imported into the correct VRF on the upstream PE router. The egress PE router programs the forwarding entry to only accept traffic from this PE router, and only on a particular tunnel rooted at that PE router.

When an egress PE router generates a type 6 C-multicast route, it uses the VRF route import extended community carried in the VPN-IP route toward the rendezvous point (RP) to construct the route target carried by the C-multicast route.

This route target results in the C-multicast route being sent to the upstream PE router and being imported into the correct VRF on the upstream PE router. The egress PE router programs the forwarding entry to accept traffic from this PE router only, and only on a particular tunnel rooted at that PE router. However, if some other PE routers have switched to SPT mode for (C-S, C-G) and have sent source active (SA) autodiscovery (A-D) routes (type 5 routes), and if the egress PE router only has (C-*, C-G) state, the upstream PE router for (C-S, C-G) is not the PE router toward the RP to which it sent a type 6 route, but the PE router that originates a SA A-D route for (C-S, C-G). The traffic for (C-S, C-G) might be carried over a I-PMSI or S-PMSI, depending on how it was advertised by the upstream PE router.

Additionally, when an egress PE router has only the (C-*, C-G) state and does not have the (C-S, C-G) state, the egress PE router might be receiving (C-S, C-G) type 5 SA routes from multiple PE routers, and chooses the best one, as follows: For every received (C-S, C-G) SA route, the egress PE router finds in its upstream multicast hop (UMH) route-candidate set for C-S a route with the same route distinguisher (RD). Among all such found routes the PE router selects the UMH route (based on the UMH selection). The best (C-S, C-G) SA route is the one whose RD is the same as of the selected UMH route.

When an egress PE router has only the (C-*, C-G) state and does not have the (C-S, C-G) state, and if later the egress PE router creates the (C-S, C-G) state (for example, as a result of receiving a PIM join (C-S, C-G) message from one of its customer edge [CE] neighbors), the upstream PE router for that (C-S, C-G) is not necessarily going to be the same PE router that originated the already-selected best SA A-D route for (C-S, C-G). It is possible to have a situation in which the PE router that originated the best SA A-D route for (C-S, C-G) carries the (C-S, C-G) over an I-PMSI, while some other PE router, that is also connected to the site that contains C-S, carries (C-S,C-G) over an S-PMSI. In this case, the downstream PE router would not join the S-PMSI, but continue to receive (C-S, C-G) over the I-PMSI, because the UMH route for C-S is the one that has been advertised by the PE router that carries (C-S, C-G) over the I-PMSI. This is expected behavior.

The egress PE router determines the sender of a (C-S, C-G) type 5 SA A-D route by finding in its UMH route-candidate set for C-S a route whose RD is the same as in the SA A-D route. The VRF route import extended community of the found route contains the IP address of the sender of the SA A-D route.

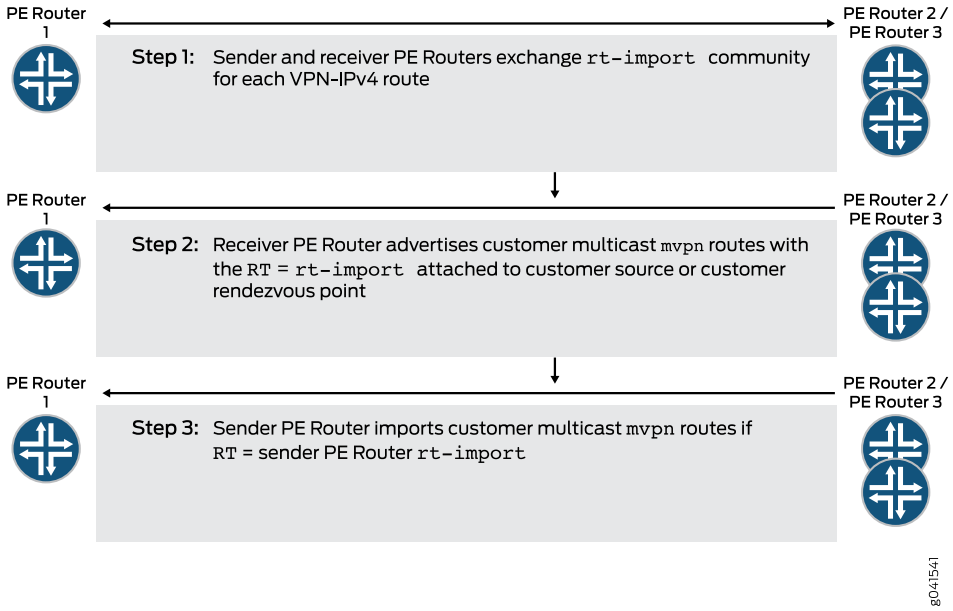
Related Documentation

- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- unicast-umh-election on page 1364

Distributing C-Multicast Routes Overview

While non-C-multicast multicast virtual private network (MVPN) routes (Type 1 – Type 5) are generally used by all provider edge (PE) routers in the network, C-multicast MVPN routes (Type 6 and Type 7) are only useful to the PE router connected to the active C-S or candidate rendezvous point (RP). Therefore, C-multicast routes need to be installed only in the VPN routing and forwarding (VRF) table on the active sender PE router for a given C-G. To accomplish this, Internet draft draft-ietf-l3vpn-2547bis-mcast-10.txt specifies to attach a special and dynamic route target to C-multicast MVPN routes (Figure 90 on page 546).

Figure 90: Attaching a Special and Dynamic Route Target to C-Multicast MVPN Routes



The route target attached to C-multicast routes is also referred to as the C-multicast import route target and should not to be confused with route target import (Table 24 on page 546). Note that C-multicast MVPN routes differ from other MVPN routes in one essential way: they carry a dynamic route target whose value depends on the identity of the active sender PE router at a given time and can change if the active PE router changes.

Table 24: Distinction Between Route Target Improt Attached to VPN-IPv4 Routes and Route Target Attached to C-Multicast MVPN Routes

Route Target Import Attached to VPN-IPV4 Routes	Route Target Attached to C-Multicast MVPN Routes
Value generated by the originating PE router. Must be unique per VRF table.	Value depends on the identity of the active PE router.

Table 24: Distinction Between Route Target Improt Attached to VPN-IPv4 Routes and Route Target Attached to C-Multicast MVPN Routes (continued)

Route Target Import Attached to VPN-IPV4 Routes	Route Target Attached to C-Multicast MVPN Routes
Static. Created upon configuration to help identify to which PE router and to which VPN the VPN unicast routes belong.	Dynamic because if the active sender PE router changes, then the route target attached to the C-multicast routes must change to target the new sender PE router. For example, a new VPN source attached to a different PE router becomes active and preferred.

A PE router that receives a local C-join determines the identity of the active sender PE router by performing a unicast route lookup for the C-S or candidate rendezvous point (router) [candidate RP] in the unicast VRF table. If there is more than one route, the receiver PE router chooses a single forwarder PE router. The procedures used for choosing a single forwarder are outlined in Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt and are not covered in this topic.

After the active sender (upstream) PE router is selected, the receiver PE router constructs the C-multicast MVPN route corresponding to the local C-join.

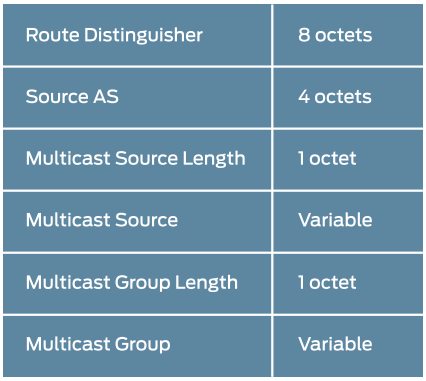
After the C-multicast route is constructed, the receiver PE router needs to attach the correct route target to this route targeting the active sender PE router. As mentioned, each PE router creates a unique VRF route target import community and attaches it to the VPN-IPv4 routes. When the receiver PE router does a route lookup for C-S or candidate RP, it can extract the value of the route target import associated with this route and set the value of the C-import route target to the value of the route target import.

On the active sender PE router, C-multicast routes are imported only if they carry the route target whose value is the same as the route target import that the sender PE router generated.

Constructing C-Multicast Routes

A PE router originates a C-multicast MVPN route in response to receiving a C-join through its PE-CE interface. See Figure 91 on page 547 for the formats in the C-multicast route encoded in MCAST-VPN NLRI. Table 25 on page 548 describes each field.

Figure 91: C-Multicast Route Type MCAST-VPN NLRI Format



g041542

Table 25: C-Multicast Route Type MCAST-VPN NLRI Format Descriptions

Field	Description
Route Distinguisher	Set to the route distinguisher of the C-S or candidate RP (the route distinguisher associated with the upstream PE router).
Source AS	Set to the value found in the **src-as** community of the C-S or candidate RP.
Multicast Source Length	Set to 32 for IPv4 and to 128 for IPv6 C-S or candidate RP IP addresses.
Multicast Source	Set to the IP address of the C-S or candidate RP.
Multicast Group Length	Set to 32 for IPv4 and to 128 for IPv6 C-G addresses.
Multicast Group	Set to the C-G of the received C-join.

This same structure is used for encoding both Type 6 and Type 7 routes with two differences:

- The first difference is the value used for the multicast source field. For Type 6 routes, this field is set to the IP address of the candidate RP configured. For Type 7 routes, this field is set to the IP address of the C-S contained in the (C-S, C-G) message.
- The second difference is the value used for the route distinguisher. For Type 6 routes, this field is set to the route distinguisher that is attached to the IP address of the candidate RP. For Type 7 routes, this field is set to the route distinguisher that is attached to the IP address of the C-S.

Eliminating PE-PE Distribution of (C-*, C-G) State Using Source Active Autodiscovery Routes

PE routers must maintain additional state when the C-multicast routing protocol is Protocol Independent Multicast-Sparse Mode (PIM-SM) in any-source multicast (ASM). This is a requirement because with ASM, the receivers first join the shared tree rooted at the candidate RP (called a candidate RP tree or candidate RPT). However, as the VPN multicast sources become active, receivers learn the identity of the sources and join the tree rooted at the source (called a customer shortest-path tree or C-SPT). The receivers then send a prune message to the candidate RP to stop the traffic coming through the shared tree for the group that they joined to the C-SPT. The switch from candidate RPT to C-SPT is a complicated process requiring additional state.

Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt specifies optional procedures that completely eliminate the need for joining the candidate RPT. These procedures require PE routers to keep track of all active VPN sources using one of two options. The first option is to colocate the candidate RP on one of the PE routers. The second option is to use the Multicast Source Discovery Protocol (MSDP) between one of the PE routers and the customer candidate RP.

In this approach, a PE router that receives a local (C-*, C-G) join creates a Type 6 route, but does not advertise the route to the remote PE routers until it receives information about an active source. The PE router acting as the candidate RP (or that learns about

active sources via MSDP) is responsible for originating a Type 5 route. A Type 5 route carries information about the active source and the group addresses. The information contained in a Type 5 route is enough for receiver PE routers to join the C-SPT by originating a Type 7 route toward the sender PE router, completely skipping the advertisement of the Type 6 route that is created when a C-join is received. Figure 92 on page 549 shows the format of a source active (SA) autodiscovery route. Table 26 on page 549 describes each format.

Figure 92: Source Active Autodiscovery Route Type MCAST-VPN NLRI Format

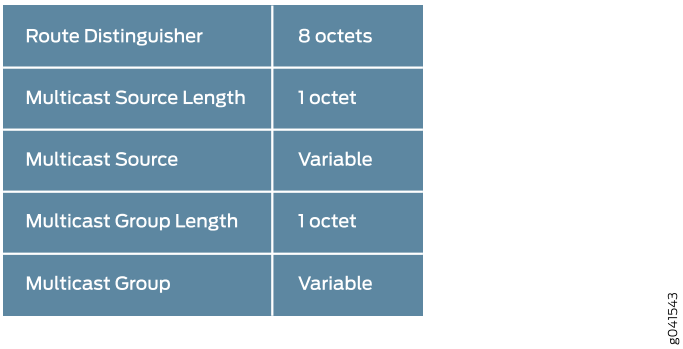


Table 26: Source Active Autodiscovery Route Type MCAST-VPN NLRI Format Descriptions

Field	Description
Route Distinguisher	Set to the route distinguisher configured on the router originating the SA autodiscovery route.
Multicast Source Length	Set to 32 for IPv4 and to 128 for IPv6 C-S IP addresses.
Multicast Source	Set to the IP address of the C-S that is actively transmitting data to C-G.
Multicast Group Length	Set to 32 for IPv4 and to 128 for IPv6 C-G addresses.
Multicast Group	Set to the IP address of the C-G to which C-S is transmitting data.

Receiving C-Multicast Routes

The sender PE router imports C-multicast routes into the VRF table based on the route target of the route. If the route target attached to the C-multicast MVPN route matches the route target import community originated by this router, the C-multicast MVPN route is imported into the VRF table. If not, it is discarded.

Once the C-multicast MVPN routes are imported, they are translated back to C-joins and passed on to the VRF C-PIM protocol for further processing per normal PIM procedures.

Related Documentation

- Enabling Next-Generation MVPN Services on page 560
- Exchanging C-Multicast Routes on page 550
- Understanding Next-Generation MVPN Network Topology on page 532

Exchanging C-Multicast Routes

This section describes PE-PE distribution of Type 7 routes discussed in "Signaling Provider Tunnels and Data Plane Setup" on page 570.

In source-tree-only mode, a receiver provider edge (PE) router generates and installs a Type 6 route in its **<routing-instance-name>.mvpn.0** table in response to receiving a (C-*, C-G) message from a local receiver, but does not advertise this route to other PE routers via BGP. The receiver PE router waits for a Type 5 route corresponding to the C-join.

Type 5 routes carry information about active sources and can be advertised by any PE router. In Junos OS, a PE router originates a Type 5 route if one of the following conditions occurs:

- PE router starts receiving multicast data directly from a VPN multicast source.
- PE router is the candidate rendezvous point (router) (candidate RP) and starts receiving C-PIM register messages.
- PE router has a Multicast Source Discovery Protocol (MSDP) session with the candidate RP and starts receiving MSDP Source Active routes.

Once both Type 6 and Type 5 routes are installed in the **<routing-instance-name>.mvpn.0** table, the receiver PE router is ready to originate a Type 7 route

Advertising C-Multicast Routes Using BGP

If the C-join received over a VPN interface is a source tree join (C-S, C-G), then the receiver PE router simply originates a Type 7 route (Step 7 in the following procedure). If the C-join is a shared tree join (C-*, C-G), then the receiver PE router needs to go through a few steps (Steps 1-7) before originating a Type 7 route.

Note that Router PE1 is the candidate RP that is conveniently located in the same router as the sender PE router. If the sender PE router and the PE router acting as (or MSDP peering with) the candidate RP are different, then the VPN multicast register messages first need to be delivered to the PE router acting as the candidate RP that is responsible for originating the Type 5 route. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

1. A PE router that receives a (C-*, C-G) join message processes the message using normal C-PIM procedures and updates its C-PIM database accordingly.

   Enter the **show pim join extensive instance vpna 224.1.1.1** command on Router PE3 to verify that Router PE3 creates the C-PIM database after receiving the (*, 224.1.1.1) C-join message from Router CE3:

```
user@PE3> show pim join extensive instance vpna 224.1.1.1
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 10.12.53.1
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
  Upstream interface: Through BGP
  Upstream neighbor: Through MVPN
  Upstream state: Join to RP
  Downstream neighbors:
      Interface: so-0/2/0.0
            10.12.87.1 State: Join Flags: SRW Timeout: Infinity
```

2. The (C-*, C-G) entry in the C-PIM database triggers the generation of a Type 6 route that is then installed in the **<routing-instance-name>.mvpn.0** table by C-PIM. The Type 6 route uses the candidate RP IP address as the source.

   Enter the **show route table vpna.mvpn.0 detail | find 6:10.1.1.1** command on Router PE3 to verify that Router PE3 installs the following Type 6 route in the **vpna.mvpn.0** table:

```
user@PE3> show route table vpna.mvpn.0 detail | find 6:10.1.1.1
6:10.1.1.1:1:65000:32:10.12.53.1:32:224.1.1.1/240 (1 entry, 1 announced)
               *PIM     Preference: 105
         Next hop type: Multicast (IPv4), Next hop index: 262144
         Next-hop reference count: 11
         State: <Active Int>
         Age: 1d 1:32:58
         Task: PIM.vpna
         Announcement bits (2): 0-PIM.vpna 1-mvpn global task
           AS path: I
         Communities: no-advertise target:10.1.1.1:64
```

3. The route distinguisher and route target attached to the Type 6 route are learned from a route lookup in the **<routing-instance-name>.inet.0** table for the IP address of the candidate RP.

   Enter the **show route table vpna.inet.0 10.12.53.1 detail** command on Router PE3 to verify that Router PE3 has the following entry for **C-RP 10.12.53.1** in the **vpna.inet.0** table:

```
user@PE3> show route table vpna.inet.0 10.12.53.1 detail
vpna.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
10.12.53.1/32 (1 entry, 1 announced)
        *BGP     Preference: 170/-101
        Route Distinguisher: 10.1.1.1:1
        Next hop type: Indirect
        Next-hop reference count: 6
        Source: 10.1.1.1
```

```
                Next hop type: Router, Next hop index: 588
                Next hop: via so-0/0/3.0, selected
                Label operation: Push 16, Push 299808(top)
                Protocol next hop: 10.1.1.1
                Push 16
                         Indirect next hop: 8da91f8 262143
                         State: <Secondary Active Int Ext>
                         Local AS: 65000 Peer AS: 65000
                         Age: 4:49:25 Metric2: 1
                         Task: BGP_65000.10.1.1.1+179
                         Announcement bits (1): 0-KRT
                         AS path: I
                        Communities: target:10:1 src-as:65000:0 rt-import:10.1.1.1:64

                         Import Accepted
                         VPN Label: 16
                         Localpref: 100
                         Router ID: 10.1.1.1
                    Primary Routing Table bgp.l3vpn.0
```

4. After the VPN source starts transmitting data, the first PE router that becomes aware of the active source (either by receiving register messages or the MSDP source-active routes) installs a Type 5 route in its **VRF mvpn** table.

   Enter the **show route table vpna.mvpn.0 detail | find 5:10.1.1.1** command on Router PE1 to verify that Router PE1 has installed the following entry in the **vpna.mvpn.0** table and starts receiving C-PIM register messages from Router CE1:

```
user@PE1> show route table vpna.mvpn.0 detail | find 5:10.1.1.1
5:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
         *PIM    Preference: 105
                 Next hop type: Multicast (IPv4)
                 Next-hop reference count: 30
                 State: <Active Int>
                 Age: 1d 1:36:33
                 Task: PIM.vpna
                 Announcement bits (3): 0-PIM.vpna 1-mvpn global task 2-BGP
RT Background
                 AS path: I
```

5. Type 5 routes that are installed in the **<routing-instance-name>.mvpn.0** table are picked up by BGP and advertised to remote PE routers.

   Enter the **show route advertising-protocol bgp 10.1.1.3 detail table vpna.mvpn.0 | find 5:** command on Router PE1 to verify that Router PE1 advertises the following Type 5 route to remote PE routers:

```
user@PE1> show route advertising-protocol bgp 10.1.1.3 detail table vpna.mvpn.0 | find 5:
* 5:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
         BGP group int type Internal
             Route Distinguisher: 10.1.1.1:1
             Nexthop: Self
             Flags: Nexthop Change
             Localpref: 100
             AS path: [65000] I
             Communities: target:10:1
```

6. The receiver PE router that has both a Type 5 and Type 6 route for (C-*, C-G) is now ready to originate a Type 7 route.

   Enter the **show route table vpna.mvpn.0 detail** command on Router PE3 to verify that Router PE3 has the following Type 5, 6, and 7 routes in the **vpna.mvpn.0** table.

   The Type 6 route is installed by C-PIM in Step 2. The Type 5 route is learned via BGP in Step 5. The Type 7 route is originated by the MVPN module in response to having both Type 5 and Type 6 routes for the same (C-*, C-G). The route target of the Type 7 route is the same as the route target of the Type 6 route because both routes (IP address of the candidate RP [10.12.53.1] and the address of the VPN multicast source [192.168.1.2]) are reachable via the same router [PE1]). Therefore, **10.12.53.1** and **192.168.1.2** carry the same route target import (**10.1.1.1:64**) community

```
user@PE3> show route table vpna.mvpn.0 detail
5:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
         *BGP     Preference: 170/-101
                  Next hop type: Indirect
                  Next-hop reference count: 4
                  Source: 10.1.1.1
                  Protocol next hop: 10.1.1.1
                  Indirect next hop: 2 no-forward
                  State: <Secondary Active Int Ext>
                  Local AS: 65000 Peer AS: 65000
                  Age: 1d 1:43:13 Metric2: 1
                  Task: BGP_65000.10.1.1.1+55384
                  Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                  AS path: I
                  Communities: target:10:1
                  Import Accepted
                  Localpref: 100
                  Router ID: 10.1.1.1
                  Primary Routing Table bgp.mvpn.0

6:10.1.1.1:1:65000:32:10.12.53.1:32:224.1.1.1/240 (1 entry, 1 announced)
       *PIM      Preference: 105
                 Next hop type: Multicast (IPv4), Next hop index: 262144
                 Next-hop reference count: 11
                 State: <Active Int>
                 Age: 1d 1:44:09
                 Task: PIM.vpna
                 Announcement bits (2): 0-PIM.vpna 1-mvpn global task
                 AS path: I
                 Communities: no-advertise target:10.1.1.1:64

7:10.1.1.1:1:65000:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
       *MVPN         Preference: 70
                Next hop type: Multicast (IPv4), Next hop index: 262144
                Next-hop reference count: 11
                State: <Active Int Ext>
                Age: 1d 1:44:09 Metric2: 1
                Task: mvpn global task
                Announcement bits (3): 0-PIM.vpna 1-mvpn global task 2-BGP RT
 Background
                AS path: I
                Communities: target:10.1.1.1:64
```

7. The Type 7 route installed in the VRF MVPN table is picked up by BGP and advertised to remote PE routers.

   Enter the **show route advertising-protocol bgp 10.1.1.1 detail table vpna.mvpn.0 | find 7:10.1.1.1** command on Router PE3 to verify that Router PE3 advertises the following Type 7 route:

```
user@PE3> show route advertising-protocol bgp 10.1.1.1 detail table vpna.mvpn.0 | find 7:10.1.1.1
* 7:10.1.1.1:1:65000:32:192.168.1.2:32:224.1.1.1/240 (1 entry, 1 announced)
         BGP group int type Internal
             Route Distinguisher: 10.1.1.3:1
             Nexthop: Self
             Flags: Nexthop Change
             Localpref: 100
             AS path: [65000] I
             Communities: target:10.1.1.1:64
```

8. If the C-join is a source tree join, then the Type 7 route is originated immediately (without waiting for a Type 5 route).

   Enter the **show route table vpna.mvpn.0 detail | find 7:10.1.1.1** command on Router PE2 to verify that Router PE2 originates the following Type 7 route in response to receiving a (**192.168.1.2, 232.1.1.1**) C-join:

```
user@PE2> show route table vpna.mvpn.0 detail | find 7:10.1.1.1
7:10.1.1.1:1:65000:32:192.168.1.2:32:232.1.1.1/240 (1 entry, 1 announced)
          *PIM    Preference: 105
                  Next hop type: Multicast (IPv4), Next hop index: 262146
                  Next-hop reference count: 4
                  State: <Active Int>
                  Age: 2d 18:59:56
                  Task: PIM.vpna
                  Announcement bits (3): 0-PIM.vpna 1-mvpn global task 2-BGP
RT Background
                  AS path: I
                  Communities: target:10.1.1.1:64
```

Receiving C-Multicast Routes

A sender PE router imports a Type 7 route if the route is carrying a route target that matches the locally originated route target import community. All Type 7 routes must pass the **__vrf-mvpn-import-cmcast-<routing-instance-name>-internal__** policy in order to be installed in the **<routing-instance-name>.mvpn.0** table.

When a sender PE router receives a Type 7 route via BGP, this route is installed in the **<routing-instance-name>.mvpn.0** table. The BGP route is then translated back into a normal C-join inside the VRF table, and the C-join is installed in the local C-PIM database of the receiver PE router. A new C-join added to the C-PIM database triggers C-PIM to originate a Type 6 or Type 7 route. The C-PIM on the sender PE router creates its own version of the same Type 7 route received via BGP.

Use the **show route table vpna.mvpn.0 detail | find 7:10.1.1.1** command to verify that Router PE1 contains the following entries for a Type 7 route in the **vpna.mvpn.0** table corresponding to a (**192.168.1.2, 224.1.1.1**) join message. There are two entries; one entry

is installed by PIM and the other entry is installed by BGP. This example also shows the Type 7 route corresponding to the (**192.168.1.2, 232.1.1.1**) join.

```
user@PE1> show route table vpna.mvpn.0 detail | find 7:10.1.1.1
7:10.1.1.1:1:65000:32:192.168.1.2:32:224.1.1.1/240 (2 entries, 2 announced)
*PIM     Preference: 105
         Next hop type: Multicast (IPv4)
         Next-hop reference count: 30
         State: <Active Int>
         Age: 1d 2:19:04
         Task: PIM.vpna
         Announcement bits (2): 0-PIM.vpna 1-mvpn global task
         AS path: I
         Communities: no-advertise target:10.1.1.1:64
BGP      Preference: 170/-101
         Next hop type: Indirect
         Next-hop reference count: 4
         Source: 10.1.1.3
         Protocol next hop: 10.1.1.3
         Indirect next hop: 2 no-forward
         State: <Secondary Int Ext>
         Inactive reason: Route Preference
         Local AS: 65000 Peer AS: 65000
         Age: 53:27      Metric2: 1
         Task: BGP_65000.10.1.1.3+179
         Announcement bits (2): 0-PIM.vpna 1-mvpn global task
         AS path: I
         Communities: target:10.1.1.1:64
         Import Accepted
         Localpref: 100
         Router ID: 10.1.1.3
         Primary Routing Table bgp.mvpn.0
7:10.1.1.1:1:65000:32:192.168.1.2:32:232.1.1.1/240 (2 entries, 2 announced)
*PIM     Preference: 105
         Next hop type: Multicast (IPv4)
         Next-hop reference count: 30
         State: <Active Int>
         Age: 2d 19:21:17
         Task: PIM.vpna
         Announcement bits (2): 0-PIM.vpna 1-mvpn global task
         AS path: I
         Communities: no-advertise target:10.1.1.1:64
BGP      Preference: 170/-101
         Next hop type: Indirect
         Next-hop reference count: 4
         Source: 10.1.1.2
         Protocol next hop: 10.1.1.2
         Indirect next hop: 2 no-forward
         State: <Secondary Int Ext>
         Inactive reason: Route Preference
         Local AS: 65000 Peer AS: 65000
         Age: 53:27      Metric2: 1
         Task: BGP_65000.10.1.1.2+49165
         Announcement bits (2): 0-PIM.vpna 1-mvpn global task
         AS path: I
         Communities: target:10.1.1.1:64
         Import Accepted
         Localpref: 100
         Router ID: 10.1.1.2
         Primary Routing Table bgp.mvpn.0
```

Remote C-joins (Type 7 routes learned via BGP translated back to normal C-joins) are installed in the VRF C-PIM database on the sender PE router and are processed based on regular C-PIM procedures. This process completes the end-to-end C-multicast routing exchange.

Use the **show pim join extensive instance vpna** command to verify that Router PE1 has installed the following entries in the C-PIM database:

```
user@PE1> show pim join extensive instance vpna
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: 192.168.1.2
    Flags: sparse,spt
    Upstream interface: fe-0/2/0.0
    Upstream neighbor: 10.12.97.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 201
    Downstream neighbors:
    Interface: Pseudo-MVPN

Group: 232.1.1.1
    Source: 192.168.1.2
    Flags: sparse,spt
    Upstream interface: fe-0/2/0.0
    Upstream neighbor: 10.12.97.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout:
    Downstream neighbors:
    Interface: Pseudo-MVPN

Instance: PIM.vpna Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

Related Documentation

- Signaling Provider Tunnels and Data Plane Setup on page 570
- Distributing C-Multicast Routes Overview on page 546
- Understanding MBGP Multicast VPN Extranets on page 669

Next-Generation MVPN Data Plane Overview

A next-generation multicast virtual private network (MVPN) data plane is composed of provider tunnels originated by and rooted at the sender provider edge (PE) routers and the receiver PE routers as the leaves of the provider tunnel.

A provider tunnel can carry data for one or more VPNs. Those provider tunnels that carry data for more than one VPN are called aggregate provider tunnels and are outside the scope of this topic. Here, we assume that a provider tunnel carries data for only one VPN.

This topic covers two types of tunnel technologies: IP generic routing encapsulation (GRE) provider tunnels signaled by Protocol Independent Multicast-Sparse Mode

(PIM-SM) any-source multicast (ASM) and MPLS provider tunnels signaled by RSVP-Traffic Engineering (RSVP-TE).

When a provider tunnel is signaled by PIM, the sender PE router runs another instance of the PIM protocol on the provider's network (P-PIM) that signals a provider tunnel for that VPN. When a provider tunnel is signaled by RSVP-TE, the sender PE router initiates a point-to-multipoint label-switched path (LSP) toward receiver PE routers by using point-to-multipoint RSVP-TE protocol messages. In either case, the sender PE router advertises the tunnel signaling protocol and the tunnel ID to other PE routers via BGP by attaching the provider multicast service interface (PMSI) attribute to either the Type 1 intra-AS autodiscovery routes (inclusive provider tunnels) or Type 3 S-PMSI autodiscovery routes (selective provider tunnels).

NOTE: The sender PE router goes through two steps when setting up the data plane. First, using the PMSI attribute, it advertises the provider tunnel it is using via BGP. Second, it actually signals the tunnel using whatever tunnel signaling protocol is configured for that VPN. This allows receiver PE routers to bind the tunnel that is being signaled to the VPN that imported the Type 1 intra-AS autodiscovery route. Binding a provider tunnel to a VRF table enables a receiver PE router to map the incoming traffic from the core network on the provider tunnel to the local target VRF table.

The PMSI attribute contains the provider tunnel type and an identifier. The value of the provider tunnel identifier depends on the tunnel type. Table 27 on page 557 identifies the tunnel types specified in Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt.

Table 27: Tunnel Types Supported by PMSI Tunnel Attribute

Tunnel Type	Description
0	No tunnel information present
1	RSVP-TE point-to-multipoint LSP
2	Multicast LDP point-to-multipoint LSP
3	PIM-SSM tree
4	PIM-SM tree
5	PIM-Bidir tree
6	Ingress replication
7	Multicast LDP multipoint-to-multipoint LSP

Inclusive Provider Tunnels

This section describes various types of provider tunnels and attributes of provider tunnels.

PMSI Attribute of Inclusive Provider Tunnels Signaled by PIM-SM

When the Tunnel Type field of the PMSI attribute is set to 4 (PIM-SM Tree), the tunnel identifier field contains **<Sender Address, P-Multicast Group Address>**. The **Sender Address** field is set to the router ID of the sender PE router. The P-multicast group address is set to a multicast group address from the service provider's P-multicast address space and uniquely identifies the VPN. A receiver PE router that receives an intra-AS autodiscovery route with a PMSI attribute whose tunnel type is PIM-SM is required to join the provider tunnel.

For example, if the service provider deploys PIM-SM provider tunnels (instead of RSVP-TE provider tunnels), Router PE1 advertises the following PMSI attribute:

PMSI: 0:PIM-SM:label[0:0:0]:Sender10.1.1.1 Group 239.1.1.1

PMSI Attribute of Inclusive Provider Tunnels Signaled by RSVP-TE

When the tunnel type field of the PMSI attribute is set to 1 (RSVP-TE point-to-multipoint LSP), the tunnel identifier field contains an RSVP-TE point-to-multipoint session object as described in RFC 4875. The session object contains the **<Extended Tunnel ID, Reserved, Tunnel ID, P2MP ID>** associated with the point-to-multipoint LSPs.

The PE router that originates the PMSI attribute is required to signal an RSVP-TE point-to-multipoint LSP and the sub-LSPs. A PE router that receives this PMSI attribute must establish the appropriate state to properly handle the traffic received over the sub-LSP.

For example, Router PE1 advertises the following PMSI attribute:

PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:6574:10.1.1.1]

Selective Provider Tunnels (S-PMSI Autodiscovery/Type 3 and Leaf Autodiscovery/Type 4 Routes)

A selective provider tunnel is used for mapping a specific C-multicast flow (a (C-S, C-G) pair) onto a specific provider tunnel. There are a variety of situations in which selective provider tunnels can be useful. For example, they can be used for putting high-bandwidth VPN multicast data traffic onto a separate provider tunnel rather than the default inclusive provider tunnel, thus restricting the distribution of traffic to only those PE routers with active receivers.

In BGP next-generation multicast virtual private networks (MVPNs), selective provider tunnels are signaled using Type 3 Selective-PMSI (S-PMSI) autodiscovery routes. See Figure 93 on page 559 and Table 28 on page 559 for details. The sender PE router sends a Type 3 route to signal that it is sending traffic for a particular (C-S, C-G) flow using an S-PMSI provider tunnel.

Figure 93: S-PMSI Autodiscovery Route Type Multicast (MCAST)-VPN Network Layer Reachability Information (NLRI) Format

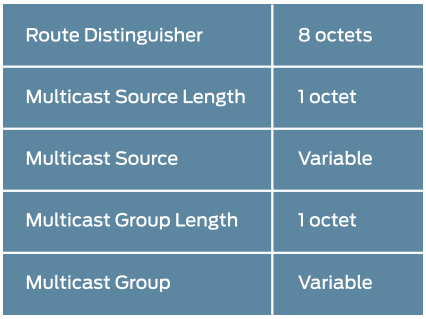


Table 28: S-PMSI Autodiscovery Route Type Format Descriptions

Field	Description
Route Distinguisher	Set to the route distinguisher configured on the router originating this route.
Multicast Source Length	Set to 32 for IPv4 and to 128 for IPv6 C-S IP addresses.
Multicast Source	Set to the C-S IP address.
Multicast Group Length	Set to 32 for IPv4 and to 128 for IPv6 C-G addresses.
Multicast Group	Set to the C-G address.

The S-PMSI autodiscovery (Type 3) route carries a PMSI attribute similar to the PMSI attribute carried with intra-AS autodiscovery (Type 1) routes. The **Flags** field of the PMSI attribute carried by the S-PMSI autodiscovery route is set to the leaf information required. This flag signals receiver PE routers to originate a Type 4 leaf autodiscovery route (Figure 94 on page 559) to join the selective provider tunnel if they have active receivers. See Table 29 on page 559 for details of leaf autodiscovery route type MCAST-VPN NLRI format descriptions.

Figure 94: Leaf Autodiscovery Route Type MCAST-VPN NLRI Format

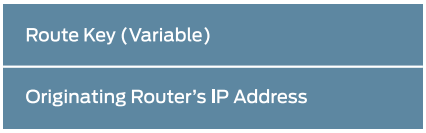


Table 29: Leaf Autodiscovery Route Type MCAST-VPN NLRI Format Descriptions

Field	Description
Route Key	Contains the original Type 3 route received.

Table 29: Leaf Autodiscovery Route Type MCAST-VPN NLRI Format Descriptions (continued)

Field	Description
Originating Router's IP Address	Set to the IP address of the PE router originating the leaf autodiscovery route This is typically the primary loopback address.

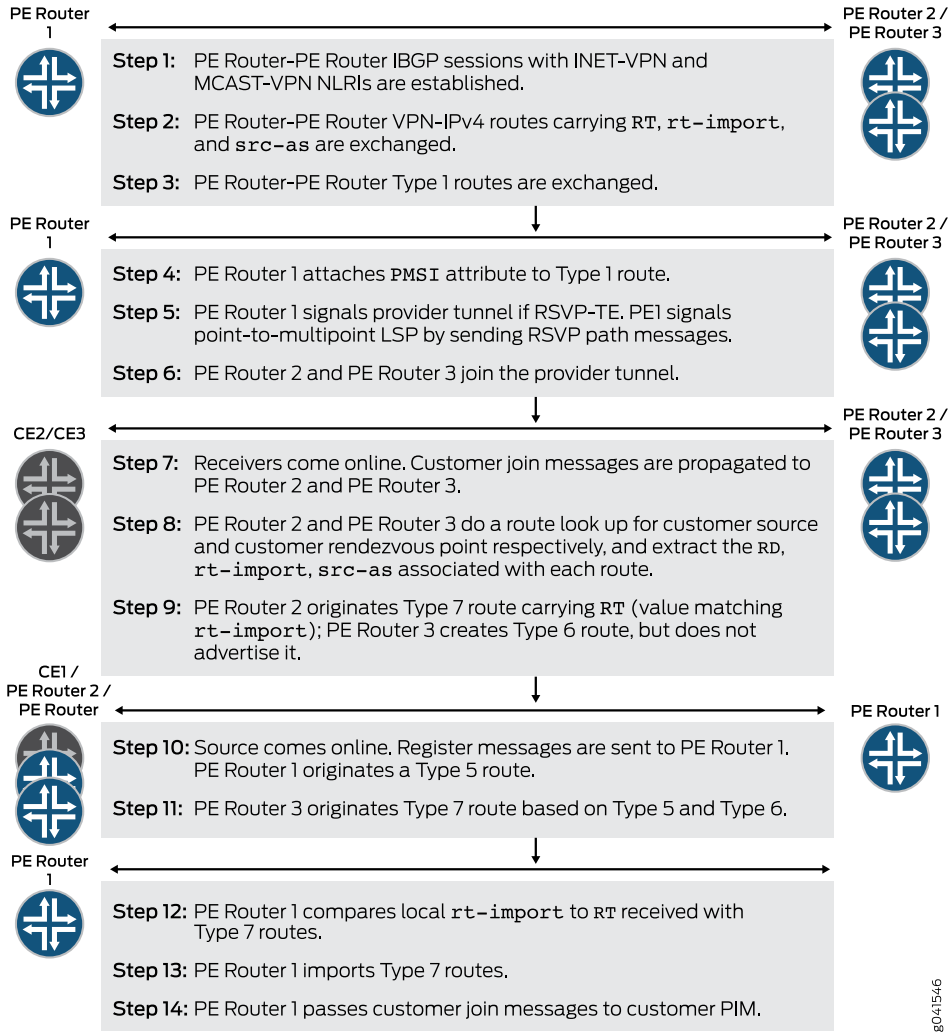
Related Documentation

- Understanding Next-Generation MVPN Control Plane on page 536
- Enabling Next-Generation MVPN Services on page 560
- Signaling Provider Tunnels and Data Plane Setup on page 570
- Understanding Next-Generation MVPN Network Topology on page 532

Enabling Next-Generation MVPN Services

Juniper Networks introduced the industry's first implementation of BGP next-generation multicast virtual private networks (MVPNs). See Figure 95 on page 561 for a summary of a Junos OS next-generation MVPN routing flow.

Figure 95: Junos OS Next-Generation MVPN Routing Flow



Next-generation MVPN services are configured on top of BGP-MPLS unicast VPN services.

You can configure a Juniper Networks PE router that is already providing unicast BGP-MPLS VPN connectivity to support multicast VPN connectivity in three steps:

1. Configure the provider edge (PE) routers to support the BGP multicast VPN address family by including the **signaling** statement at the **[edit protocols bgp group *group-name* family inet-mvpn]** hierarchy level. This address family enables PE routers to exchange MVPN routes.

2. Configure the PE routers to support the MVPN control plane tasks by including the **mvpn** statement at the **[edit routing-instances *routing-instance-name* protocols]** hierarchy level. This statement signals PE routers to initialize the MVPN module that is responsible for the majority of next-generation MVPN control plane tasks.

3. Configure the sender PE router to signal a provider tunnel by including the **provider-tunnel** statement at the **[edit routing-instances *routing-instance-name*]** hierarchy level. You must also enable the tunnel signaling protocol (RSVP-TE or P-PIM) if it is not part of the unicast VPN service configuration. To enable the tunnel signaling protocol, include the **rsvp-te** or **pim-asm** statements at the **[edit routing-instances *routing-instance-name* provider-tunnel]** hierarchy level.

After these three statements are configured and each PE router has established internal BGP (IBGP) sessions using both INET-VPN and MCAST-VPN address families, four routing tables are automatically created. These tables are **bgp.l3vpn.0, bgp.mvpn.0**, **<routing-instance-name>.inet.0**, and **<routing-instance-name>.mvpn.0**. See Table 30 on page 562

Table 30: Automatically Generated Routing Tables

Automatically Generated Routing Table	Description
bgp.l3vpn.0	Populated with VPN-IPv4 routes received from remote PE routers via the INET-VPN address family. The routes in the **bgp.l3vpn.0** table are in the form of **RD:IPv4-address** and carry one or more routing table communities. In a next-generation MVPN network, these routes also carry **rt-import** and **src-as** communities.
bgp.mvpn.0	Populated by MVPN routes (Type 1 – Type 7). Received from remote PE routers via the MCAST-VPN address family. Routes in this table carry one or more routing table communities.
<routing-instance-name>.inet.0	Populated by local and remote VPN unicast routes. The local VPN routes are typically learned from local CE routers via protocols such as BGP, OSPF, and RIP, or via a static configuration. The remote VPN routes are imported from the **bgp.l3vpn.0** table if their routing table matches one of the import routing tables configured for the VPN. When remote VPN routes are imported from the **bgp.l3vpn.0** table, their route distinguisher is removed, leaving them as regular unicast IPv4 addresses.
<routing-instance-name>.mvpn.0	Populated by local and remote MVPN routes. The local MVPN routes are typically the locally originated routes, such as Type 1 intra-AS autodiscovery routes, or Type 7 C-multicast routes. The remote MVPN routes are imported from the **bgp.mvpn.0** table based on their route target. The import route target used for accepting MVPN routes into the **<routing-instance-name>.mvpn.0** table is different for C-multicast MVPN routes (Type 6 and Type 7) versus non-C-multicast MVPN routes (Type 1 – Type 5).

Related Documentation

- Understanding Next-Generation MVPN Network Topology on page 532
- Generating Next-Generation MVPN VRF Import and Export Policies Overview on page 563
- Generating Source AS and Route Target Import Communities Overview on page 566
- Originating Type 1 Intra-AS Autodiscovery Routes Overview on page 567
- Signaling Provider Tunnels and Data Plane Setup on page 570

Generating Next-Generation MVPN VRF Import and Export Policies Overview

In Junos OS, the policy module is responsible for VPN routing and forwarding (VRF) route import and export decisions. You can configure these policies explicitly, or Junos OS can generate them internally for you to reduce user-configured statements and simplify configuration. Junos OS generates all necessary policies for supporting next-generation multicast virtual private network (MVPN) import and export decisions. Some of these policies affect normal VPN unicast routes.

The system gives a name to each internal policy it creates. The name of an internal policy starts and ends with a "__" notation. Also the keyword **internal** is added at the end of each internal policy name. You can display these internal policies using the **show policy** command.

Policies That Support Unicast BGP-MPLS VPN Services

A Juniper Networks provider edge (PE) router requires a vrf-import and a vrf-export policy to control unicast VPN route import and export decisions for a VRF. You can configure these policies explicitly at the **[edit routing-instances *routing-instance-name* vrf-import *import_policy_name*]** and **[edit routing-instances *routing-instance-name* vrf-export *export_policy_name*]** hierarchy level. Alternately, you can configure only the route target for the VRF at the **[edit routing-instances *routing-instance-name* vrf-target]** hierarchy level, and Junos OS then generates these policies automatically for you. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

The following list identifies the automatically generated policy names and where they are applied:

Policy: vrf-import

Naming convention: __vrf-import-<routing-instance-name>-internal__

Applied to: VPN-IPv4 routes in the bgp.l3vpn.0 table

Policy: vrf-export

Naming convention: __vrf-export-<routing-instance-name>-internal__

Applied to: Local VPN routes in the <routing-instance-name>.inet.0 table

Use the **show policy __vrf-import-vpna-internal__** command to verify that Router PE1 has created the following **vrf-import** and **vrf-export** policies based on a vrf-target of **target:10:1**. In this example, we see that the **vrf-import** policy is constructed to accept a route if the route target of the route matches **target:10:1**. Similarly, a route is exported with a route target of **target:10:1**.

```
user@PE1> show policy __vrf-import-vpna-internal__
Policy __vrf-import-vpna-internal__:
  Term unnamed:
    from community __vrf-community-vpna-common-internal__ [target:10:1]
    then accept
```

```
  Term unnamed:
    then reject
```

```
user@PE1> show policy __vrf-export-vpna-internal__
Policy __vrf-export-vpna-internal__:
  Term unnamed:
    then community + __vrf-community-vpna-common-internal__ [target:10:1] accept
```

The values in this example are as follows:

- Internal import policy name: __vrf-import-vpna-internal__
- Internal export policy name: __vrf-export-vpna-internal__
- RT community used in both import and export policies: __vrf-community-vpna-common-internal__
- RT value: target:10:1

Policies That Support Next-Generation MVPN Services

When you configure the **mvpn** statement at the **[edit routing-instances *routing-instance-name* protocols]** hierarchy level, Junos OS automatically creates three new internal policies: one for export, one for import, and one for handling Type 4 routes. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

The following list identifies the automatically generated policy names and where they are applied:

Policy 1: This policy is used to attach **rt-import** and **src-as** extended communities to VPN-IPv4 routes.

Policy name: __vrf-mvpn-export-inet-<routing-instance-name>-internal__

Applied to: All routes in the <routing-instance-name>inet.0 table

Use the **show policy __vrf-mvpn-export-inet-vpna-internal__** command to verify that the following export policy is created on Router PE1. Router PE1 adds **rt-import:10.1.1.1:64** and **src-as:65000:0** communities to unicast VPN routes through this policy.

```
user@PE1> show policy __vrf-mvpn-export-inet-vpna-internal__
Policy __vrf-mvpn-export-inet-vpna-internal__:
     Term unnamed:
    then community + __vrf-mvpn-community-rt_import-vpna-internal__
[rt-import:10.1.1.1:64 ] community + __vrf-mvpn-community-src_as-vpna-internal__
 [src-as:65000:0 ] accept
```

The values in this example are as follows:

- Policy name: __vrf-mvpn-export-inet-vpna-internal__
- rt-import community name: __vrf-mvpn-community-rt_import-vpna-internal__
- rt-import community value: rt-import:10.1.1.1:64

- src-as community name: __vrf-mvpn-community-src_as-vpna-internal__
- src-as community value: src-as:65000:0

Policy 2: This policy is used to import C-Mmulticast routes from the **bgp.mvpn.0** table to the **<routing-instance-name>.mvpn.0** table.

Policy name: __vrf-mvpn-import-cmcast-<routing-instance-name>-internal__

Applied to: C-multicast (MVPN) routes in the bgp.mvpn.0 table

Use the **show policy __vrf-mvpn-import-cmcast-vpna-internal__** command to verify that the following import policy is created on Router PE1. The policy accepts those C-multicast MVPN routes carrying a route target of **target:10.1.1.1:64** and installs them in the **vpna.mvpn.0** table.

```
user@PE1> show policy __vrf-mvpn-import-cmcast-vpna-internal__
Policy __vrf-mvpn-import-cmcast-vpna-internal__:
  Term unnamed:
   from community __vrf-mvpn-community-rt_import-target-vpna-internal__
[target:10.1.1.1:64 ]
  then accept
 Term unnamed:
  then reject
```

The values in this example are as follows:

- Policy name: __vrf-mvpn-import-cmcast-vpna-internal__
- C-multicast import RT community: __vrf-mvpn-community-rt_import-target-vpna-internal__
- Community value: target:10.1.1.1:64

Policy 3: This policy is used for importing Type 4 routes and is created by default even if a selective provider tunnel is not configured. The policy affects only Type 4 routes received from receiver PE routers.

Policy name: __vrf-mvpn-import-cmcast-leafAD-global-internal__

Applied to: Type 4 routes in the bgp.mvpn.0 table

Use the **show policy __vrf-mvpn-import-cmcast-leafAD-global-internal__** command to verify that the following import policy is created on Router PE1.

```
user@PE1> show policy __vrf-mvpn-import-cmcast-leafAD-global-internal__
Policy __vrf-mvpn-import-cmcast-leafAD-global-internal__:
  Term unnamed:
    from community __vrf-mvpn-community-rt_import-target-global-internal__
[target:10.1.1.1:0 ]
    then accept
  Term unnamed:
    then reject
```

Related Documentation

- Understanding MBGP Multicast VPN Extranets on page 669
- Example: Configuring MBGP Multicast VPN Extranets on page 671
- Example: Configuring MBGP Multicast VPNs on page 606
- Enabling Next-Generation MVPN Services on page 560

Generating Source AS and Route Target Import Communities Overview

Both route target import (**rt-import**) and source autonomous system (**src-as**) communities contain two fields (following their respective keywords). In Junos OS, a provider edge (PE) router constructs the route target import community using its router ID in the first field and a per-VRF unique number in the second field. The router ID is normally set to the primary loopback IP address of the PE router. The unique number used in the second field is an internal number derived from the routing-instance table index. The combination of the two numbers creates a route target import community that is unique to the originating PE router and unique to the VPN routing and forwarding (VRF) instance from which it is created.

For example, Router PE1 creates the following route target import community: **rt-import:10.1.1.1:64**.

Since the route target import community is constructed using the primary loopback address and the routing-instance table index of the PE router, any event that causes either number to change triggers a change in the value of the route target import community. This in turn requires VPN-IPv4 routes to be re-advertised with the new route target import community. Under normal circumstances, the primary loopback address and the routing-instance table index numbers do not change. If they do change, Junos OS updates all related internal policies and re-advertises VPN-IPv4 routes with the new **rt-import** and **src-as** values per those policies.

To ensure that the route target import community generated by a PE router is unique across VRF tables, the Junos OS Policy module restricts the use of primary loopback addresses to next-generation multicast virtual private network (MVPN) internal policies only. You are not permitted to configure a route target for any VRF table (MVPN or otherwise) using the primary loopback address. The commit fails with an error if the system finds a user-configured route target that contains the IP address used in constructing the route target import community.

The global administrator field of the **src-as** community is set to the local AS number of the PE router originating the community, and the local administrator field is set to **0**. This community is used for inter-AS operations but needs to be carried along with all VPN-IPv4 routes.

For example, Router PE1 creates an **src-as** community with a value of **src-as:65000:0**.

Related Documentation

- Originating Type 1 Intra-AS Autodiscovery Routes Overview on page 567
- Generating Next-Generation MVPN VRF Import and Export Policies Overview on page 563
- Enabling Next-Generation MVPN Services on page 560

Originating Type 1 Intra-AS Autodiscovery Routes Overview

Every provider edge (PE) router that is participating in the next-generation multicast virtual private network (MVPN) is required to originate a Type 1 intra-AS autodiscovery route. In Junos OS, the MVPN module is responsible for installing the intra-AS autodiscovery route in the local **<routing-instance-name>.mvpn.0** table. All PE routers advertise their local Type 1 routes to each other. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show route table vpna.mvpn.0** command to verify that Router PE1 has installed intra-AS AD routes in the **vpna.mvpn.0** table. The route is installed by the MVPN protocol (meaning it is the MVPN module that originated the route), and the mask for the entire route is /240.

```
user@PE1> show route table vpna.mvpn.0
vpna.mvpn.0: 6 destinations, 9 routes (6 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:10.1.1.1:1:10.1.1.1/240
                   *[MVPN/70] 04:09:44, metric2 1
                      Indirect
```

Attaching Route Target Community to Type 1 Routes

Intra-AS AD routes are picked up by the BGP protocol from the **<routing-instance-name>.mvpn.0** table and advertised to the remote PE routers via the MCAST-VPN address family. By default, intra-AS autodiscovery routes carry the same route target community that is attached to the unicast VPN-IPv4 routes. If the unicast and multicast network topologies are not congruent, then you can configure a different set of import route target and export route target communities for non-C-multicast MVPN routes (C-multicast MVPN routes always carry a dynamic import route target).

Multicast route targets are configured by including the **import-target** and **export-target** statements at the **[edit routing-instances *routing-instance-name* protocols mvpn route-target]** hierarchy level.

Junos OS creates two additional internal policies in response to configuring multicast route targets. These policies are applied to non-C-multicast MVPN routes during import and export decisions. Multicast VPN routing and forwarding (VRF) internal import and export polices follow a naming convention similar to unicast VRF import and export policies. The contents of these polices are also similar to policies applied to unicast VPN routes.

The following list identifies the default policy names and where they are applied:

Multicast VRF import policy:
__vrf-mvpn-import-target-<routing-instance-name>-internal__

Multicast VRF export policy: __vrf-mvpn-export-target-<routing-instance-name>-internal__

Use the **show policy __vrf-mvpn-import-target-vpna-internal__** command on Router PE1 to verify that Router PE1 has created the following internal MVPN policies if import-target and export-target are configured to be target:10:2:

```
user@PE1> show policy __vrf-mvpn-import-target-vpna-internal__
Policy __vrf-mvpn-import-target-vpna-internal__:
  Term unnamed:
   from community __vrf-mvpn-community-import-vpna-internal__ [target:10:2 ]
   then accept
  Term unnamed:
   then reject
```

```
user@PE1> show policy __vrf-mvpn-export-target-vpna-internal__
Policy __vrf-mvpn-export-target-vpna-internal__:
Term unnamed:
then community + __vrf-mvpn-community-export-vpna-internal__ [target:10:2 ] accept
```

The values in this example are as follows:

- Multicast import RT community: __vrf-mvpn-community-import-vpna-internal__
- Multicast export RT community: __vrf-mvpn-community-export-vpna-internal__ Value: target:10:2

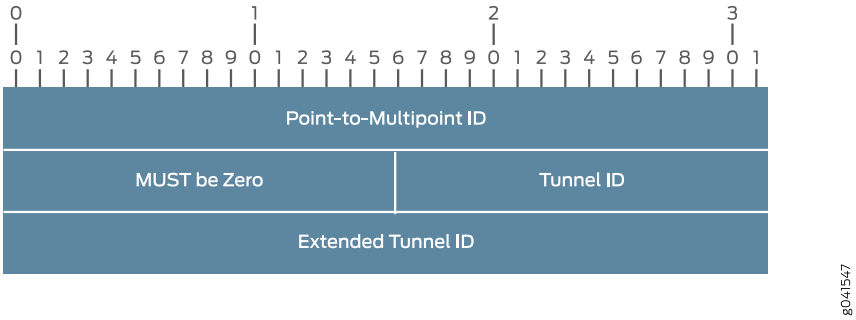
Attaching the PMSI Attribute to Type 1 Routes

The provider multicast service interface (PMSI) attribute is originated and attached to Type 1 intra-AS autodiscovery routes by the sender PE routers when the **provider-tunnel** statement is included at the **[edit routing-instances *routing-instance-name*]** hierarchy level. Since provider tunnels are signaled by the sender PE routers, this statement is not necessary on the PE routers that are known to have VPN multicast receivers only.

If the provider tunnel configured is Protocol Independent Multicast-Sparse Mode (PIM-SM) any-source multicast (ASM), then the PMSI attribute carries the IP address of the sender-PE and provider tunnel group address. The provider tunnel group address is assigned by the service provider (through configuration) from the provider's multicast address space and is not to be confused with the multicast addresses used by the VPN customer.

If the provider tunnel configured is the RSVP-Traffic Engineering (RSVP-TE) type, then the PMSI attribute carries the RSVP-TE point-to-multipoint session object. This point-to-multipoint session object is used as the identifier for the parent point-to-multipoint label-switched path (LSP) and contains the fields shown in Figure 96 on page 569.

Figure 96: RSVP-TE Point-to-Multipoint Session Object Format



In Junos OS, the **P2MP ID** and **Extended Tunnel ID** fields are set to the router ID of the sender PE router. The **Tunnel ID** is set to the port number used for the point-to-multipoint RSVP session that is unique for the length of the RSVP session.

Use the **show rsvp session p2mp detail** command to verify that Router PE1 signals the following RSVP sessions to Router PE2 and Router PE3 (using port number 6574). In this example, Router PE1 is signaling a point-to-multipoint LSP named 10.1.1.1:65535:mvpn:vpna with two sub-LSPs. Both sub-LSPs **10.1.1.3:10.1.1.1:65535:mvpn:vpna** and **10.1.1.2:10.1.1.1:65535:mvpn:vpna** use the same RSVP port number (6574) as the parent point-to-multipoint LSP.

```
user@PE1> show rsvp session p2mp detail
Ingress RSVP: 2 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 2

10.1.1.3
  From: 10.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: 10.1.1.3:10.1.1.1:65535:mvpn:vpna, LSPpath: Primary
  P2MP LSPname: 10.1.1.1:65535:mvpn:vpna
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299968
  Resv style: 1 SE, Label in: -, Label out: 299968
  Time left: -, Since: Wed May 27 07:36:22 2009
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 6574 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  RESV rcvfrom: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  Explct route: 10.12.100.6 10.12.100.22
  Record route: <self> 10.12.100.6 10.12.100.22

10.1.1.2
  From: 10.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: 10.1.1.2:10.1.1.1:65535:mvpn:vpna, LSPpath: Primary
  P2MP LSPname: 10.1.1.1:65535:mvpn:vpna
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299968
  Resv style: 1 SE, Label in: -, Label out: 299968
  Time left: -, Since: Wed May 27 07:36:22 2009
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 6574 protocol 0
```

```
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  RESV rcvfrom: 10.12.100.6 (fe-0/2/3.0) 27 pkts
  Explct route: 10.12.100.6 10.12.100.9
  Record route: <self> 10.12.100.6 10.12.100.9
Total 2 displayed, Up 2, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Sender-Only and Receiver-Only Sites

In Junos OS, you can configure a PE router to be a sender-site only or a receiver-site only. These options are enabled by including the **sender-site** and **receiver-site** statements at the **[edit routing-instances *routing-instance-name* protocols mvpn]** hierarchy level.

- A sender-site only PE router does not join the provider tunnels advertised by remote PE routers
- A receiver-site only PE router does not send a PMSI attribute

The commit fails if you include the **receiver-site** and **provider-tunnel** statements in the same VPN.

Related Documentation

- Generating Source AS and Route Target Import Communities Overview on page 566
- Understanding MBGP Multicast VPN Extranets on page 669
- Signaling Provider Tunnels and Data Plane Setup on page 570
- Generating Next-Generation MVPN VRF Import and Export Policies Overview on page 563

Signaling Provider Tunnels and Data Plane Setup

In a next-generation multicast virtual private network (MVPN), provider tunnel information is communicated to the receiver PE routers in an out-of-band manner. This information is advertised via BGP and is independent of the actual tunnel signaling process. Once the tunnel is signaled, the sender PE router binds the VPN routing and forwarding (VRF) table to the locally configured tunnel. The receiver PE routers bind the tunnel signaled to the VRF table where the Type 1 autodiscovery route with the matching provider multicast service interface (PMSI) attribute is installed. The same binding process is used for both Protocol Independent Multicast (PIM) and RSVP-Traffic Engineering (RSVP-TE) signaled provider tunnels.

Provider Tunnels Signaled by PIM (Inclusive)

A sender provider edge (PE) router configured to use an inclusive PIM-sparse mode (PIM-SM) any-source multicast (ASM) provider tunnel for a VPN creates a multicast tree (using the P-group address configured) in the service provider network. This tree is

rooted at the sender PE router and has the receiver PE routers as the leaves. VPN multicast packets received from the local VPN source are encapsulated by the sender PE router with a multicast generic routing encapsulation (GRE) header containing the P-group address configured for the VPN. These packets are then forwarded on the service provider network as normal IP multicast packets per normal P-PIM procedures. At the leaf nodes, the GRE header is stripped and the packets are passed on to the local VRF C-PIM protocol for further processing.

In Junos OS, a logical interface called multicast tunnel (MT) is used for GRE encapsulation and de-encapsulation of VPN multicast packets. The multicast tunnel interface is created automatically if a Tunnel PIC is present.

- Encapsulation subinterfaces are created from an mt-x/y/z.[32768-49151] range.
- De-encapsulation subinterfaces are created from an mt-x/y/z.[49152-65535] range.

The multicast tunnel subinterfaces act as pseudo upstream or downstream interfaces between C-PIM and P-PIM.

In the following two examples, assume that the network uses PIM-SM (ASM) signaled GRE tunnels as the tunneling technology. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show interfaces mt-0/1/0 terse** command to verify that Router PE1 has created the following multicast tunnel subinterface. The logical interface number is 32768, indicating that this sub-unit is used for GRE encapsulation.

```
user@PE1> show interfaces mt-0/1/0 terse
Interface       Admin   Link   Proto   Local   Remote
        mt-0/1/0            up          up
        mt-0/1/0.32768      up          up      inet
                                                inet6
```

Use the **show interfaces mt-0/1/0 terse** command to verify that Router PE2 has created the following multicast tunnel subinterface. The logical interface number is 49152, indicating that this sub-unit is used for GRE de-encapsulation.

```
user@PE2> show interfaces mt-0/1/0 terse
Interface       Admin  Link   Proto   Local   Remote
       mt-0/1/0            up          up
       mt-0/1/0.49152      up          up       inet
                                                inet6
```

P-PIM and C-PIM on the Sender PE Router

The sender PE router installs a local join entry in its P-PIM database for each VRF table configured to use PIM as the provider tunnel. The outgoing interface list (OIL) of this entry points to the core-facing interface. Since the P-PIM entry is installed as **Local**, the sender PE router sets the source address to its primary loopback IP address.

Use the **show pim join extensive** command to verify that Router PE1 has installed the following state in its P-PIM database.

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local Source
    Keepalive timeout: 339
    Downstream neighbors:
        Interface: fe-0/2/3.0
            10.12.100.6 State: Join Flags: S Timeout: 195

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

On the VRF side of the sender PE router, C-PIM installs a **Local Source** entry in its C-PIM database for the active local VPN source. The OIL of this entry points to **Pseudo-MVPN**, indicating that the downstream interface points to the receivers in the next-generation MVPN network. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show pim join extensive instance vpna 224.1.1.1** command to verify that Router PE1 has installed the following entry in its C-PIM database.

```
user@PE1> show pim join extensive instance vpna 224.1.1.1
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: 192.168.1.2
    Flags: sparse,spt
    Upstream interface: fe-0/2/0.0
    Upstream neighbor: 10.12.97.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 0
    Downstream neighbors:
        Interface: Pseudo-MVPN
```

The forwarding entry corresponding to the C-PIM **Local Source** (or **Local RP**) on the sender PE router points to the multicast tunnel encapsulation subinterface as the downstream interface. This indicates that the local multicast data packets are encapsulated as they are passed on to the P-PIM protocol.

Use the **show multicast route extensive instance vpna group 224.1.1.1** command to verify that Router PE1 has the following multicast forwarding entry for group 224.1.1.1. The upstream interface is the PE-CE interface and the downstream interface is the multicast tunnel encapsulation subinterface:

```
user@PE1> show multicast route extensive instance vpna group 224.1.1.1
```

```
Family: INET

Group: 224.1.1.1
    Source: 192.168.1.2/32
    Upstream interface: fe-0/2/0.0
    Downstream interface list:
        mt-0/1/0.32768
    Session description: ST Multicast Groups
    Statistics: 7 kBps, 79 pps, 719738 packets
    Next-hop ID: 262144
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
```

P-PIM and C-PIM on the Receiver PE Router

On the receiver PE router, multicast data packets received from the network are de-encapsulated as they are passed through the multicast tunnel de-encapsulation interface.

The P-PIM database on the receiver PE router contains two P-joins. One is for P-RP, and the other is for the sender PE router. For both entries, the OIL contains the multicast tunnel de-encapsulation interface from which the GRE header is stripped. The upstream interface for P-joins is the core-facing interface that faces towards the sender PE router.

Use the **show pim join extensive** command to verify that Router PE3 has the following state in its P-PIM database. The downstream neighbor interface points to the GRE de-encapsulation subinterface:

```
user@PE3> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: *
    RP: 10.1.1.10
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/3.0
    Upstream neighbor: 10.12.100.21
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: mt-1/2/0.49152
            10.12.53.13 State: Join Flags: SRW Timeout: Infinity

Group: 239.1.1.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: so-0/0/3.0
    Upstream neighbor: 10.12.100.21
    Upstream state: Join to Source
    Keepalive timeout: 351
    Downstream neighbors:
        Interface: mt-1/2/0.49152
            10.12.53.13 State: Join Flags: S Timeout: Infinity
```

```
Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

On the VRF side of the receiver PE router, C-PIM installs a join entry in its C-PIM database. The OIL of this entry points to the local VPN interface, indicating active local receivers. The upstream protocol, interface, and neighbor of this entry point to the next-generation-MVPN network. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show pim join extensive instance vpna 224.1.1.1** command to verify that Router PE3 has the following state in its C-PIM database:

```
user@PE3> show pim join extensive instance vpna 224.1.1.1
Instance: PIM.vpna Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 10.12.53.1
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP
    Upstream neighbor: Through MVPN
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: so-0/2/0.0
            10.12.87.1 State: Join Flags: SRW Timeout: Infinity

Group: 224.1.1.1
    Source: 192.168.1.2
    Flags: sparse
    Upstream protocol: BGP
    Upstream interface: Through BGP
    Upstream neighbor: Through MVPN
    Upstream state: Join to Source
    Keepalive timeout:
    Downstream neighbors:
        Interface: so-0/2/0.0
            10.12.87.1 State: Join Flags: S Timeout: 195

Instance: PIM.vpna Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

The forwarding entry corresponding to the C-PIM entry on the receiver PE router uses the multicast tunnel de-encapsulation subinterface as the upstream interface.

Use the **show multicast route extensive instance vpna group 224.1.1.1** command to verify that Router PE3 has installed the following multicast forwarding entry for the local receiver:

```
user@PE3> show multicast route extensive instance vpna group 224.1.1.1
Family: INET

Group: 224.1.1.1
    Source: 192.168.1.2/32
    Upstream interface: mt-1/2/0.49152
    Downstream interface list:
```



```
    so-0/2/0.0
Session description: ST Multicast Groups
Statistics: 1 kBps, 10 pps, 149 packets
Next-hop ID: 262144
Upstream protocol: MVPN
Route state: Active
Forwarding state: Forwarding
Cache lifetime/timeout: forever
Wrong incoming interface notifications: 0
```

Provider Tunnels Signaled by RSVP-TE (Inclusive and Selective)

Junos OS supports signaling both inclusive and selective provider tunnels by RSVP-TE point-to-multipoint label-switched paths (LSPs). You can configure a combination of inclusive and selective provider tunnels per VPN.

- If you configure a VPN to use an inclusive provider tunnel, the sender PE router signals one point-to-multipoint LSP for the VPN.
- If you configure a VPN to use selective provider tunnels, the sender PE router signals a point-to-multipoint LSP for each selective tunnel configured.

Sender (ingress) PE routers and receiver (egress) PE routers play different roles in the point-to-multipoint LSP setup. Sender PE routers are mainly responsible for initiating the parent point-to-multipoint LSP and the sub-LSPs associated with it. Receiver PE routers are responsible for setting up state such that they can forward packets received over a sub-LSP to the correct VRF table (binding a provider tunnel to the VRF).

Inclusive Tunnels: Ingress PE Router Point-to-Multipoint LSP Setup

The point-to-multipoint LSP and associated sub-LSPs are signaled by the ingress PE router. The information about the point-to-multipoint LSP is advertised to egress PE routers in the PMSI attribute via BGP.

The ingress PE router signals point-to-multipoint sub-LSPs by originating point-to-multipoint RSVP path messages toward egress PE routers. The ingress PE router learns the identity of the egress PE routers from Type 1 routes installed in its **<routing-instance-name>.mvpn.0** table. Each RSVP path message carries an **S2L_Sub_LSP** object along with the point-to-multipoint session object. The **S2L_Sub_LSP** object carries a 4-byte sub-LSP destination (egress) IP address.

In Junos OS, sub-LSPs associated with a point-to-multipoint LSP can be signaled automatically by the system or via a static sub-LSP configuration. When they are automatically signaled, the system chooses a name for the point-to-multipoint LSP and each sub-LSP associated with it using the following naming convention.

Point-to-multipoint LSPs naming convention:

<ingress PE rid>:<a per VRF unique number>:mvpn:<routing-instance-name>

Sub-LSPs naming convention:

<egress PE rid>:<ingress PE rid>:<a per VRF unique number>:mvpn:<routing-instance-name>

Use the **show mpls lsp p2mp** command to verify that the following LSPs have been created by Router PE1:

Parent P2MP LSP: 10.1.1.1:65535:mvpn:vpna

Sub-LSPs: 10.1.1.2:10.1.1.1:65535:mvpn:vpna (Router PE1 to Router PE2) and

10.1.1.3:10.1.1.1:65535:mvpn:vpna (Router PE1 to Router PE3)

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 1 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 2
To              From            State Rt P  ActivePath   LSPname
10.1.1.2        10.1.1.1        Up     0 *
10.1.1.2:10.1.1.1:65535:mvpn:vpna
10.1.1.3        10.1.1.1        Up     0 *
10.1.1.3:10.1.1.1:65535:mvpn:vpna
Total 2 displayed, Up 2, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

The values in this example are as follows:

- I-PMSI P2MP LSP name: 10.1.1.1:65535:mvpn:vpna
- I-PMSI P2MP sub-LSP name (to PE2): 10.1.1.2:10.1.1.1:65535:mvpn:vpna
- I-PMSI P2MP sub-LSP name (to PE3): 10.1.1.3:10.1.1.1:65535:mvpn:vpna

Inclusive Tunnels: Egress PE Router Point-to-Multipoint LSP Setup

An egress PE router responds to an RSVP path message by originating an RSVP reservation (RESV) message per normal RSVP procedures. The RESV message contains the MPLS label allocated by the egress PE router for this sub-LSP and is forwarded hop by hop toward the ingress PE router, thus setting up state on the network. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show rsvp session** command to verify that Router PE2 has assigned label **299840** for the sub-LSP **10.1.1.2:10.1.1.1:65535:mvpn:vpna**:

```
user@PE2> show rsvp session
Total 0 displayed, Up 0, Down 0
Egress RSVP: 1 sessions
To              From      State     Rt Style     Labelin    Labelout    LSPname
10.1.1.2     10.1.1.1  Up  0 1 SE  299840          -
10.1.1.2:10.1.1.1:65535:mvpn:vpna
Total 1 displayed, Up 1, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Use the **show mpls lsp p2mp** command to verify that Router PE3 has assigned label **16** for the sub-LSP **10.1.1.3:10.1.1.1:65535:mvpn:vpna**:

```
user@PE3> show mpls lsp p2mp
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress LSP: 1 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 1
To              From      State     Rt Style     Labelin    Labelout    LSPname
10.1.1.3     10.1.1.1 Up 0 1 SE       16           -
10.1.1.3:10.1.1.1:65535:mvpn:vpna
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

Inclusive Tunnels: Egress PE Router Data Plane Setup

The egress PE router installs a forwarding entry in its **mpls** table for the label it allocated for the sub-LSP. The MPLS label is installed with a pop operation (a pop operation removes the top MPLS label), and the packet is passed on to the VRF table for a second route lookup. The second lookup on the egress PE router is necessary for the VPN multicast data packets to be processed inside the VRF table using normal C-PIM procedures.

Use the **show route table mpls label 16** command to verify that Router PE3 has installed the following label entry in its MPLS forwarding table:

```
user@PE3> show route table mpls label 16
+ = Active Route, - = Last Active, * = Both

16                 *[VPN/0] 03:03:17
                    to table vpna.inet.0, Pop
```

In Junos OS, VPN multicast routing entries are stored in the **<routing-instance-name>.inet.1** table, which is where the second route lookup occurs. In the example above, even though **vpna.inet.0** is listed as the routing table where the second lookup happens after the pop operation, internally the lookup is pointed to the **vpna.inet.1** table. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show route table vpna.inet.1** command to verify that Router PE3 contains the following entry in its VPN multicast routing table:

```
user@PE3> show route table vpna.inet.1
vpna.inet.1: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

224.1.1.1,192.168.1.2/32*[MVPN/70] 00:04:10
                        Multicast (IPv4)
```

Use the **show multicast route extensive instance vpna** command to verify that Router PE3 contains the following VPN multicast forwarding entry corresponding to the multicast routing entry for the Llocal join. The upstream interface points to **lsi.0** and the downstream

interface (OIL) points to the **so-0/2/0.0** interface (toward local receivers). The **Upstream protocol** value is **MVPN** because the VPN multicast source is reachable via the next-generation MVPN network. The **lsi.0** interface is similar to the multicast tunnel interface used when PIM-based provider tunnels are used. The **lsi.0** interface is used for removing the top MPLS header.

```
user@PE3> show multicast route extensive instance vpna
Family: INET

Group: 224.1.1.1
    Source: 192.168.1.2/32
    Upstream interface: lsi.0
    Downstream interface list:
        so-0/2/0.0
    Session description: ST Multicast Groups
    Statistics: 1 kBps, 10 pps, 3472 packets
    Next-hop ID: 262144
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Family: INET6
```

The requirement for a double route lookup on the VPN packet header requires two additional configuration statements on the egress PE routers when provider tunnels are signaled by RSVP-TE.

First, since the top MPLS label used for the point-to-multipoint sub-LSP is actually tied to the VRF table on the egress PE routers, the penultimate-hop popping (PHP) operation is not used for next-generation MVPNs. Only ultimate-hop popping is used. PHP allows the penultimate router (router before the egress PE router) to remove the top MPLS label. PHP works well for VPN unicast data packets because they typically carry two MPLS labels: one for the VPN and one for the transport LSP.

After the LSP label is removed, unicast VPN packets still have a VPN label that can be used for determining the VPN to which the packets belong. VPN multicast data packets, on the other hand, carry only one MPLS label that is directly tied to the VPN. Therefore, the MPLS label carried by VPN multicast packets must be preserved until the packets reach the egress PE router. Normally, PHP must be disabled through manual configuration.

To simplify the configuration, PHP is disabled by default on Juniper Networks PE routers when you include the **mvpn** statement at the **[edit routing-instances *routing-interface-name* interface]** hierarchy level. PHP is also disabled by default when you include the **vrf-table-label** statement at the **[edit routing-instances *routing-instance-name*]** hierarchy level.

Second, in Junos OS, VPN labels associated with a VRF table can be allocated in two ways.

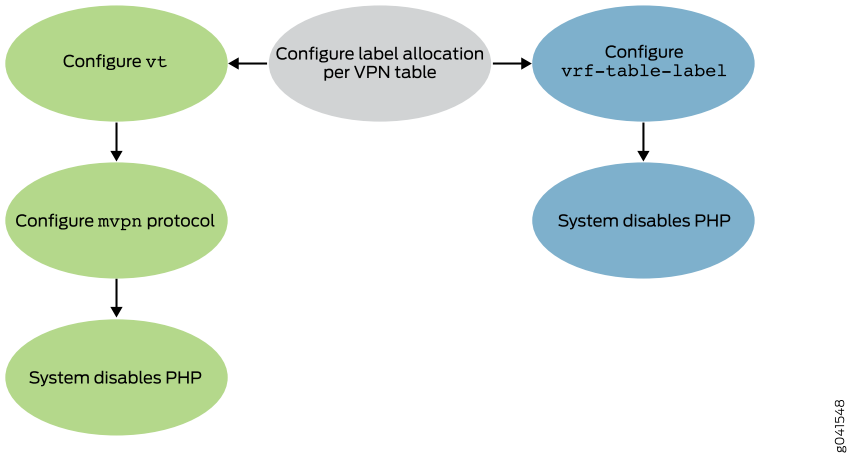
- Allocate a unique label for each VPN next hop (PE-CE interface). This is the default behavior.

- Allocate one label for the entire VRF table, which requires additional configuration. Only allocating a label for the entire VRF table allows a second lookup on the VPN packet's header. Therefore, PE routers supporting next-generation-MVPN services must be configured to allocate labels for the VRF table. There are two ways to do this as shown in Figure 97 on page 579.
  - One is by including a virtuall tunnel interface named **vt** at the **[edit routing-instances *routing-instance-name* interfaces]** hierarchy level, which requires a Tunnel PIC.
  - The second is by including the **vrf-table-label** statement at the **[routing-instances *routing-instance-name*]** hierarchy level, which does not require a Tunnel PIC.

Both of these options enable an egress PE router to perform two route lookups. However, there are some differences in the way in which the second lookup is done

If the **vt** interface is used, the allocated label is installed in the **mpls** table with a **pop** operation and a forwarding next hop pointing to the **vt** interface.

Figure 97: Enabling Double Route Lookup on VPN Packet Headers



Use the **show route table mpls label 299840** command to verify that Router PE2 has installed the following entry and uses a **vt** interface in the **mpls** table. The label associated with the point-to-multipoint sub-LSP (**299840**) is installed with a pop and a forward operation with the **vt-0/1/0.0** interface being the next hop. VPN multicast packets received from the core exit the **vt-0/1/0.0** interface without their MPLS header, and the egress Router PE2 does a second lookup on the packet header in the **vpna.inet.1** table.

```
user@PE2> show route table mpls label 299840
mpls.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299840             *[VPN/0] 00:00:22
                    > via vt-0/1/0.0, Pop
```

If the **vrf-table-label** is configured, the allocated label is installed in the **mpls** table with a pop operation, and the forwarding entry points to the **<routing-instance-name>.inet.0**

table (which internally triggers the second lookup to be done in the **<routing-instance-name>.inet.1** table).

Use the **show route table mpls label 16** command to verify that Router PE3 has installed the following entry in its **mpls** table and uses the **vrf-table-label** statement:

```
user@PE3> show route table mpls label 16
mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

16                 *[VPN/0] 03:03:17
                      to table vpna.inet.0, Pop
```

Configuring label allocation for each VRF table affects both unicast VPN and MVPN routes. However, you can enable per-VRF label allocation for MVPN routes only if per-VRF allocation is configured via **vt**. This feature is configured via multicast and unicast keywords at the **[edit routing-instances *routing-instance-name* interface vt-x/y/z.0]** hierarchy level.

Note that including the **vrf-table-label** statement enables per-VRF label allocation for both unicast and MVPN routes and cannot be turned off for either type of routes (it is either on or off for both).

If a PE router is a bud router, meaning it has local receivers and also forwards MPLS packets received over a point-to-multipoint LSP downstream to other P and PE routers, then there is a difference in how the **vrf-table-label** and **vt** statements work. When, the **vrf-table-label** statement is included, the bud PE router receives two copies of the packet from the penultimate router: one to be forwarded to local receivers and the other to be forwarded to downstream P and PE routers. When the **vt** statement is included, the PE router receives a single copy of the packet.

Inclusive Tunnels: Ingress and Branch PE Router Data Plane Setup

On the ingress PE router, local VPN data packets are encapsulated with the MPLS label received from the network for sub-LSPs.

Use the **show rsvp session** command to verify that on the ingress Router PE1, VPN multicast data packets are encapsulated with MPLS label **300016** (advertised by Router P1 per normal RSVP RESV procedures) and forwarded toward Router P1 down the sub-LSPs **10.1.1.3:10.1.1.1:65535:mvpn:vpna** and **10.1.1.2:10.1.1.1:65535:mvpn:vpna**.

```
user@PE1> show rsvp session
Ingress RSVP: 2 sessions
To          From        State     Rt Style   Labelin   Labelout   LSPname
10.1.1.3   10.1.1.1 Up 0 1 SE         -      300016
10.1.1.3:10.1.1.1:65535:mvpn:vpna
10.1.1.2 10.1.1.1   Up 0 1 SE         -      300016
10.1.1.2:10.1.1.1:65535:mvpn:vpna
Total 2 displayed, Up 2, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

RFC 4875 describes a branch node as "an LSR that replicates the incoming data on to one or more outgoing interfaces." On a branch Rrouter, the incoming data carrying an MPLS label is replicated onto one or more outgoing interfaces that can use different MPLS labels. Branch nodes keep track of incoming and outgoing labels associated with point-to-multipoint LSPs. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show rsvp session** command to verify that branch node P1 has the incoming label **300016** and outgoing labels **16** for sub-LSP **10.1.1.3:10.1.1.1:65535:mvpn:vpna** (to Router PE3) and **299840** for sub-LSP **10.1.1.2:10.1.1.1:65535:mvpn:vpna** (to Router PE2).

```
user@P1> show rsvp session
Ingress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 2 sessions
To          From        State     Rt Style   Labelin   Labelout   LSPname
10.1.1.3 10.1.1.1 Up    0 1 SE     300016       16
10.1.1.3:10.1.1.1:65535:mvpn:vpna
10.1.1.2 10.1.1.1 Up    0 1 SE     300016   299840
10.1.1.2:10.1.1.1:65535:mvpn:vpna
Total 2 displayed, Up 2, Down 0
```

Use the **show route table mpls label 300016** command to verify that the corresponding forwarding entry on Router P1 shows that the packets coming in with one MPLS label (**300016**) are swapped with labels **16** and **299840** and forwarded out through their respective interfaces (**so-0/0/3.0** and **so-0/0/1.0** respectively toward Router PE2 and Router PE3).

```
user@P1> show route table mpls label 300016
mpls.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

300016               *[RSVP/7] 01:58:15, metric 1
                      > via so-0/0/3.0, Swap 16
                        via so-0/0/1.0, Swap 299840
```

Selective Tunnels: Type 3 S-PMSI Autodiscovery and Type 4 Leaf Autodiscovery Routes

Selective provider tunnels are configured by including the **selective** statement at the **[edit routing-instances *routing-instance-name* provider-tunnel]** hierarchy level. You can configure a threshold to trigger the signaling of a selective provider tunnel. Including the **selective** statement triggers the following events.

First, the ingress PE router originates a Type 3 S-PMSI autodiscovery route. The S-PMSI autodiscovery route contains the route distinguisher of the VPN where the tunnel is configured and the (C-S, C-G) pair that uses the selective provider tunnel.

In this section assume that Router PE1 is signaling a selective tunnel for (**192.168.1.2, 224.1.1.1**) and Router PE3 has an active receiver.

Use the **show route table vpna.mvpn.0 | find 3:** command to verify that Router PE1 has installed the following Type 3 route after the selective provider tunnel is configured:

```
user@PE1> show route table vpna.mvpn.0 | find 3:
3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1/240
                           *[MVPN/70] 00:05:07, metric2 1
                              Indirect
```

Second, the ingress PE router attaches a PMSI attribute to a Type 3 route. This PMSI attribute is similar to the PMSI attribute advertised for inclusive provider tunnels with one difference: the PMSI attribute carried with Type 3 routes has its **Flags** bit set to **Leaf Information Required**. This means that the sender PE router is requesting receiver PE routers to send a Type 4 route if they have active receivers for the (C-S, C-G) carried in the Type 3 route. Also, remember that for each selective provider tunnel, a new point-to-multipoint and associated sub-LSPs are signaled. The PMSI attribute of a Type 3 route carries information about the new point-to-multipoint LSP.

Use the **show route advertising-protocol bgp 10.1.1.3 detail table vpna.mvpn | find 3:** command to verify that Router PE1 advertises the following Type 3 route and the **PMSI** attribute. The point-to-multipoint session object included in the **PMSI** attribute has a different port number (**29499**) than the one used for the inclusive tunnel (**6574**) indicating that this is a new point-to-multipoint tunnel.

```
user@PE1> show route advertising-protocol bgp 10.1.1.3 detail table vpna.mvpn | find 3:
* 3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1/240 (1 entry, 1 announced)
BGP group int type Internal
    Route Distinguisher: 10.1.1.1:1
    Nexthop: Self
    Flags: Nexthop Change
    Localpref: 100
    AS path: [65000] I
    Communities: target:10:1
    PMSI: Flags 1:RSVP-TE:label[0:0:0]:Session_13[10.1.1.1:0:29499:10.1.1.1]
```

Egress PE routers with active receivers should respond to a Type 3 route by originating a Type 4 leaf autodiscovery route. A leaf autodiscovery route contains a route key and the originating router's IP address fields. The **Route Key** field of the leaf autodiscovery route contains the original Type 3 route that is received. The originating router's IP address field is set to the router ID of the PE router originating the leaf autodiscovery route.

The ingress PE router adds each egress PE router that originated the leaf autodiscovery route as a leaf (destination of the sub-LSP for the selective point-to-multipoint LSP). Similarly, the egress PE router that originated the leaf autodiscovery route sets up forwarding state to start receiving data through the selective provider tunnel.

Egress PE routers advertise Type 4 routes with a route target that is specific to the PE router signaling the selective provider tunnel. This route target is in the form of target:<rid of the sender PE>:0. The sender PE router (the PE router signaling the selective provider tunnel) applies a special internal import policy to Type 4 routes that looks for a route

target with its own router ID. Routers referenced in this topic are shown in "Understanding Next-Generation MVPN Network Topology" on page 532.

Use the **show route table vpna.mvpn | find 4:3:** command to verify that Router PE3 originates the following Type 4 route. The local Type 4 route is installed by the MVPN module.

```
user@PE3> show route table vpna.mvpn | find 4:3:
4:3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1:1.1.1.3/240
                           *[MVPN/70] 00:15:29, metric2 1
                              Indirect
```

Use the **show route advertising-protocol bgp 10.1.1.1 table vpna.mvpn detail | find 4:3:** command to verify that Router PE3 has advertised the local Type 4 route with the following route target community. This route target carries the IP address of the sender PE router (**10.1.1.1**) followed by a 0.

```
user@PE3> show route advertising-protocol bgp 10.1.1.1 table vpna.mvpn detail | find 4:3:
* 4:3:10.1.1.1:1:32:192.168.1.2:32:224.1.1.1:10.1.1.1:10.1.1.3/240 (1 entry, 1
announced)
BGP group int type Internal
    Nexthop: Self
    Flags: Nexthop Change
    Localpref: 100
    AS path: [65000] I
    Communities: target:10.1.1.1:0
```

Use the **show policy __vrf-mvpn-import-cmcast-leafAD-global-internal__** command to verify that Router PE1 (the PE router signaling the selective provider tunnel) has applied the following import policy to Type 4 routes. The routes are accepted if their route target matches **target:10.1.1.1:0**.

```
user@PE1> show policy __vrf-mvpn-import-cmcast-leafAD-global-internal__
Policy __vrf-mvpn-import-cmcast-leafAD-global-internal__:
Term unnamed:
from community __vrf-mvpn-community-rt_import-target-global-internal__
[target:10.1.1.1:0 ]
then accept
Term unnamed:
then reject
```

For each selective provider tunnel configured, a Type 3 route is advertised and a new point-to-multipoint LSP is signaled. Point-to-multipoint LSPs created by Junos OS for selective provider tunnels are named using the following naming conventions:

- Selective point-to-multipoint LSPs naming convention:

  <ingress PE rid>:<a per VRF unique number>:mv<a unique number>:<routing-instance-name>

- Selective point-to-multipoint sub-LSP naming convention:

  <egress PE rid>:<ingress PE rid>:<a per VRF unique>:mv<a unique number>:<routing-instance-name>

Use the **show mpls lsp p2mp** command to verify that Router PE1 signals point-to-multipoint LSP **10.1.1.1:65535:mv5:vpna** with one sub-LSP **10.1.1.3:10.1.1.1:65535:mv5:vpna**. The first point-to-multipoint LSP **10.1.1.1:65535:mvpn:vpna** is the LSP created for the inclusive tunnel.

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 2 sessions
P2MP name: 10.1.1.1:65535:mvpn:vpna, P2MP branch count: 2
To          From           State       Rt P        ActivePath            LSPname
10.1.1.3 10.1.1.1 Up 0 *                              10.1.1.3:10.1.1.1:65535:mvpn
:vpna
10.1.1.2 10.1.1.1 Up 0 *                              10.1.1.2:10.1.1.1:65535:mvpn
:vpna
P2MP name: 10.1.1.1:65535:mv5:vpna, P2MP branch count: 1
To          From           State       Rt P        ActivePath            LSPname
10.1.1.3 10.1.1.1 Up 0 *                              10.1.1.3:10.1.1.1:65535:mv5
:vpna
Total 3 displayed, Up 3, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

The values in this example are as follows.

- I-PMSI P2MP LSP name: 10.1.1.1:65535:mvpn:vpna
- I-PMSI P2MP sub-LSP name (to PE2): 10.1.1.2:10.1.1.1:65535:mvpn:vpna
- I-PMSI P2MP sub-LSP name (to PE3): 10.1.1.3:10.1.1.1:65535:mvpn:vpna
- S-PMSI P2MP LSP name: 10.1.1.1:65535:mv5:vpna
- S-PMSI P2MP sub-LSP name (to PE3): 10.1.1.3:10.1.1.1:65535:mv5:vpna

Related Documentation

- Next-Generation MVPN Data Plane Overview on page 556
- Originating Type 1 Intra-AS Autodiscovery Routes Overview on page 567
- Exchanging C-Multicast Routes on page 550

Configuring Multiprotocol BGP Multicast VPNs

- Understanding Multiprotocol BGP-Based Multicast VPNs: Next-Generation on page 585
- Example: Configuring Point-to-Multipoint LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs on page 586
- Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs on page 591
- Example: Configuring MBGP Multicast VPNs on page 606
- Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN on page 624
- Example: Allowing MBGP MVPN Remote Sources on page 633

- Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family on page 637
- Example: Configuring MBGP Multicast VPN Topology Variations on page 648
- Configuring Nonstop Active Routing for BGP Multicast VPN on page 659

Understanding Multiprotocol BGP-Based Multicast VPNs: Next-Generation

Multiprotocol BGP-based multicast VPNs (also referred to as next-generation Layer 3 VPN multicast) constitute the next evolution after dual multicast VPNs (draft-rosen) and provide a simpler solution for administrators who want to configure multicast over Layer 3 VPNs.

The main characteristics of multiprotocol BGP-based multicast VPNs are:

- They extend Layer 3 VPN service (RFC 2547) to support IP multicast for Layer 3 VPN service providers.
- They follow the same architecture as specified by RFC 2547 for unicast VPNs. Specifically, BGP is used as the control plane.
- They eliminate the requirement for the virtual router (VR) model, which is specified in Internet draft draft-rosen-vpn-mcast, *Multicast in MPLS/BGP VPNs*, for multicast VPNs.
- They rely on RFC-based unicast with extensions for intra-AS and inter-AS communication.

Multiprotocol BGP-based VPNs are defined by two sets of sites: a sender set and a receiver set. Hosts within a receiver site set can receive multicast traffic and hosts within a sender site set can send multicast traffic. A site set can be both receiver and sender, which means that hosts within such a site can both send and receive multicast traffic. Multiprotocol BGP-based VPNS can span organizations (so the sites can be intranets or extranets), can span service providers, and can overlap.

Site administrators configure multiprotocol BGP-based VPNs based on customer requirements and the existing BGP and MPLS VPN infrastructure.

Route Reflector Behavior in MVPNs

BGP-based multicast VPN (MVPN) customer multicast routes are aggregated by route reflectors. A route reflector (RR) might receive a customer multicast route with the same NLRI from more than one provider edge (PE) router, but the RR readvertises only one such NLRI. If the set of PE routers that advertise this NLRI changes, the RR does not update the route. This minimizes route churn. To achieve this, the RR sets the next hop to self. In addition, the RR sets the originator ID to itself. The RR avoids unnecessary best-path computation if it receives a subsequent customer multicast route for an NLRI that the RR is already advertising. This allows aggregation of source active and customer multicast routes with the same MVPN NLRI.

See Also

- Example: Configuring Point-to-Multipoint LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs on page 586

Example: Configuring Point-to-Multipoint LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs

This example shows how to configure point-to-multipoint (P2MP) LDP label-switched paths (LSPs) as the data plane for intra-autonomous system (AS) multiprotocol BGP (MBGP) multicast VPNs (MVPNs). This feature is well suited for service providers who are already running LDP in the MPLS backbone and need MBGP MVPN functionality.

Requirements

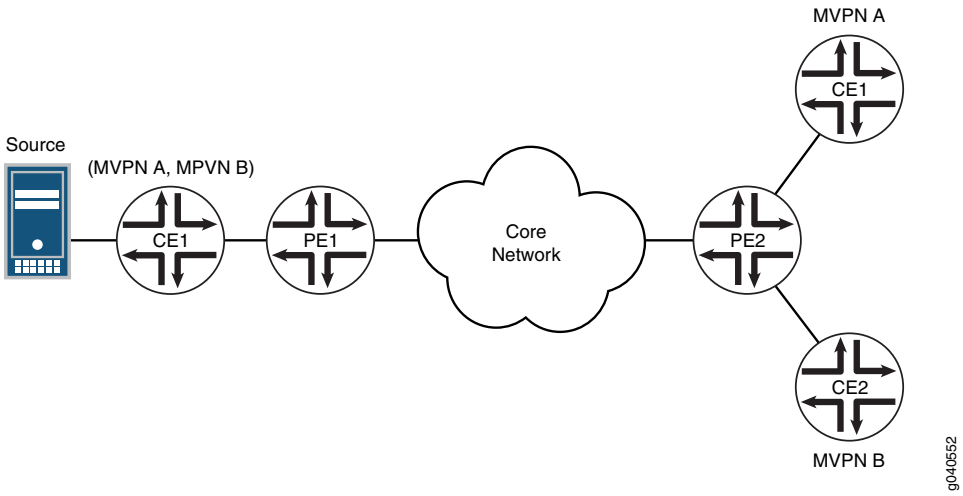
Before you begin:

- Configure the router interfaces. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure a BGP-MVPN control plane. See "MBGP-Based Multicast VPN Trees" on page 528 in the *Multicast Protocols Feature Guide* .
- Configure LDP as the signaling protocol on all P2MP provider and provider-edge routers. See *LDP Operation* in the *Junos OS MPLS Applications Library for Routing Devices*.
- Configure P2MP LDP LSPs as the provider tunnel technology on each PE router in the MVPN that belongs to the sender site set. See the *Junos OS MPLS Applications Library for Routing Devices*.
- Configure either a virtual loopback tunnel interface (requires a Tunnel PIC) or the **vrf-table-label** statement in the MVPN routing instance. If you configure the **vrf-table-label** statement, you can configure an optional virtual loopback tunnel interface as well.
- In an extranet scenario when the egress PE router belongs to multiple MVPN instances, all of which need to receive a specific multicast stream, a virtual loopback tunnel interface (and a Tunnel PIC) is required on the egress PE router. See *Configuring Virtual Loopback Tunnels for VRF Table Lookup* in the in the *Junos OS Services Interfaces Library for Routing Devices*.
- If the egress PE router is also a transit router for the point-to-multipoint LSP, a virtual loopback tunnel interface (and a Tunnel PIC) is required on the egress PE router. See *Configuring Virtual Loopback Tunnels for VRF Table Lookup* in the *Multicast Protocols Feature Guide* .
- Some extranet configurations of MBGP MVPNs with point-to-multicast LDP LSPs as the data plane require a virtual loopback tunnel interface (and a Tunnel PIC) on egress PE routers. When an egress PE router belongs to multiple MVPN instances, all of which need to receive a specific multicast stream, the **vrf-table-table** statement cannot be

used. In Figure 98 on page 587, the CE1 and CE2 routers belong to different MVPNs. However, they want to receive a multicast stream being sent by Source. If the **vrf-table-label** statement is configured on Router PE2, the packet cannot be forwarded to both CE1 and CE2. This causes packet loss. The packet is forwarded to both Routers CE1 and CE2 if a virtual loopback tunnel interface is used in both MVPN routing instances on Router PE2. Thus, you need to set up a virtual loopback tunnel interface if you are using an extranet scenario wherein the egress PE router belongs to multiple MVPN instances that receive a specific multicast stream, or if you are using the egress PE router as a transit router for the point-to-multipoint LSP.

NOTE: Starting in Junos OS Release 15.1X49-D50 and Junos OS Release 17.3R1, the **vrf-table-label** statement allows mapping of the inner label to a specific Virtual Routing and Forwarding (VRF). This mapping allows examination of the encapsulated IP header at an egress VPN router. For SRX Series devices, the **vrf-table-label** statement is currently supported only on physical interfaces. As a workaround, deactivate **vrf-table-label** or use physical interfaces.

Figure 98: Extranet Configuration of MBGP MVPN with P2MP LDP LSPs as Data Plane



See *Configuring Virtual Loopback Tunnels for VRF Table Lookup* for more information.

Overview

This topic describes how P2MP LDP LSPs can be configured as the data plane for intra-AS selective provider tunnels. Selective P2MP LSPs are triggered only based on the bandwidth threshold of a particular customer's multicast stream. A separate P2MP LDP LSP is set up for a given customer source and customer group pair (C-S, C-G) by a PE router. The C-S is behind the PE router that belongs in the sender site set. Aggregation of intra-AS selective provider tunnels across MVPNs is not supported.

When you configure selective provider tunnels, leaves discover the P2MP LSP root as follows. A PE router with a receiver for a customer multicast stream behind it needs to

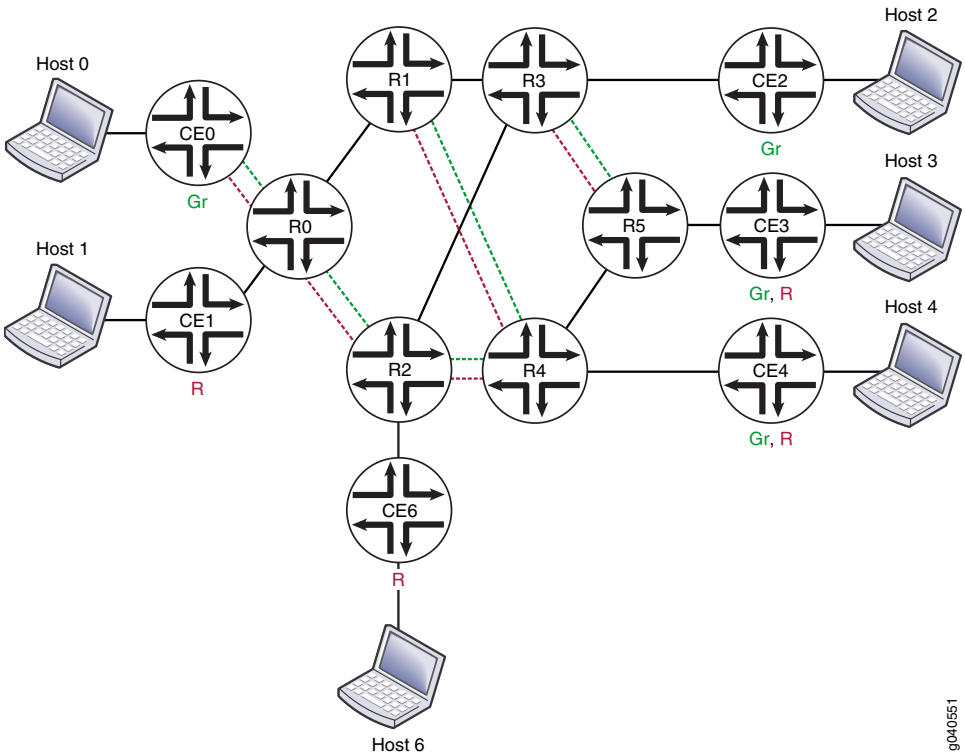
discover the identity of the PE router (and the provider tunnel information) with the source of the customer multicast stream behind it. This information is auto-discovered dynamically using the S-PMSI AD routes originated by the PE router with the C-S behind it.

The Junos OS also supports P2MP LDP LSPs as the data plane for intra-AS inclusive provider tunnels. These tunnels are triggered based on the MVPN configuration. A separate P2MP LSP LSP is set up for a given MVPN by a PE router that belongs in the sender site set. This PE router is the root of the P2MP LSP. Aggregation of intra-AS inclusive provider tunnels across MVPNs is not supported.

When you configure inclusive provider tunnels, leaves discover the P2MP LSP root as follows. A PE router with a receiver site for a given MVPN needs to discover the identities of PE routers (and the provider tunnel information) with sender sites for that MVPN. This information is auto-discovered dynamically using the intra-AS auto-discovery routes originated by the PE routers with sender sites.

Figure 99 on page 588 shows the topology used in this example.

Figure 99: P2MP LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs



In Figure 99 on page 588, the routers perform the following functions:

- R1 and R2 are provider (P) routers.
- R0, R3, R4, and R5 are provider edge (PE) routers.
- MBGP MVPN is configured on all PE routers.

- Two VPNs are defined: green and red.
- Router R0 serves both green and red CE routers in separate routing instances.
- Router R3 is connected to a green CE router.
- Router R5 is connected to overlapping green and red CE routers in a single routing instance.
- Router R4 is connected to overlapping green and red CE routers in a single routing instance.
- OSPF and multipoint LDP (mLDP) are running in the core.
- Router R1 is a route reflector (RR), and router R2 is a redundant RR.
- Routers R0, R3, R4, and R5 are client internal BGP (IBGP) peers.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols ldp interface fe-0/2/1.0
set protocols ldp interface fe-0/2/3.0
set protocols ldp p2mp
set routing-instance red instance-type mvpn
set routing-instance red interface vt-0/1/0.1
set routing-instance red interface lo0.1
set routing-instance red route-distinguisher 10.254.1.1:1
set routing-instance red provider-tunnel ldp-p2mp
set routing-instance red provider-tunnel selective group 224.1.1.1/32 source 192.168.1.1/32
  ldp-p2mp
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure P2MP LDP LSPs as the data plane for intra-AS MBGP MVPNs:

1. Configure LDP on all routers.

```
[edit protocols ldp]
user@host# set interface fe-0/2/1.0
user@host# set interface fe-0/2/3.0
user@host# set p2mp
```

2. Configure the provider tunnel.

```
[edit routing-instance red ]
user@host# set instance-type mvpn
user@host# set interface vt-0/1/0.1
user@host# set interface lo0.1
```

```
user@host# set route-distinguisher 10.254.1.1:1
user@host# set provider-tunnel ldp-p2mp
```

3. Configure the selective provider tunnel.

```
user@host# set provider-tunnel selective group 224.1.1.1/32 source 192.168.1.1/32
  ldp-p2mp
```

4. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** and **show routing-intances** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@host# show protocols
ldp {
   interface fe-0/2/1.0;
   interface fe-0/2/3.0;
   p2mp;
}
user@host# show routing-instances
red {
   instance-type vrf;
   interface vt-0/1/0.1;
   interface lo0.1;
   route-distinguisher 10.254.1.1:1;
   provider-tunnel {
      ldp-p2mp;
   }
   selective {
      group 224.1.1.1/32 {
         source 192.168.1.1/32 {
            ldp-p2mp;
         }
      }
   }
   }
}
```

Verification

To verify the configuration, run the following commands:

- **ping mpls ldp p2mp** to ping the end points of a P2MP LSP.
- **show ldp database** to display LDP P2MP label bindings and to ensure that the LDP P2MP LSP is signaled.

- **show ldp session detail** to display the LDP capabilities exchanged with the peer. The **Capabilities advertised** and **Capabilities received** fields should include **p2mp**.
- **show ldp traffic-statistics p2mp** to display the data traffic statistics for the P2MP LSP.
- show mvpn instance, show mvpn neighbor, and show mvpn c-multicast to display multicast VPN routing instance information and to ensure that the LDP P2MP LSP is associated with the MVPN as the S-PMSI.
- show multicast route **instance detail** on PE routers to ensure that traffic is received by all the hosts and to display statistics on the receivers.
- show route label ***label*** **detail** to display the P2MP forwarding equivalence class (FEC) if the label is an input label for an LDP P2MP LSP.

See Also

- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- *Point-to-Multipoint LSPs Overview*

Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs

Requirements

The routers used in this example are Juniper Networks M Series Multiservice Edge Routers, T Series Core Routers, or MX Series 3D Universal Edge Routers. When using ingress replication for IP multicast, each participating router must be configured with BGP for control plane procedures and with ingress replication for the data provider tunnel, which forms a full mesh of MPLS point-to-point LSPs. The ingress replication tunnel can be selective or inclusive, depending on the configuration of the provider tunnel in the routing instance.

Overview

The **ingress-replication** provider tunnel type uses unicast tunnels between routers to create a multicast distribution tree.

The mpls-internet-multicast routing instance type uses ingress replication provider tunnels to carry IP multicast data between routers through an MPLS cloud, using MBGP (or Next Gen) MVPN. Ingress replication can also be configured when using MVPN to carry multicast data between PE routers.

The mpls-internet-multicast routing instance is a non-forwarding instance used only for control plane procedures. It does not support any interface configurations. Only one mpls-internet-multicast routing instance can be defined for a logical system. All multicast and unicast routes used for IP multicast are associated only with the default routing instance (**inet.0**), not with a configured routing instance. The mpls-internet-multicast

routing instance type is configured for the default master instance on each router, and is also included at the **[edit protocols pim]** hierarchy level in the default instance.

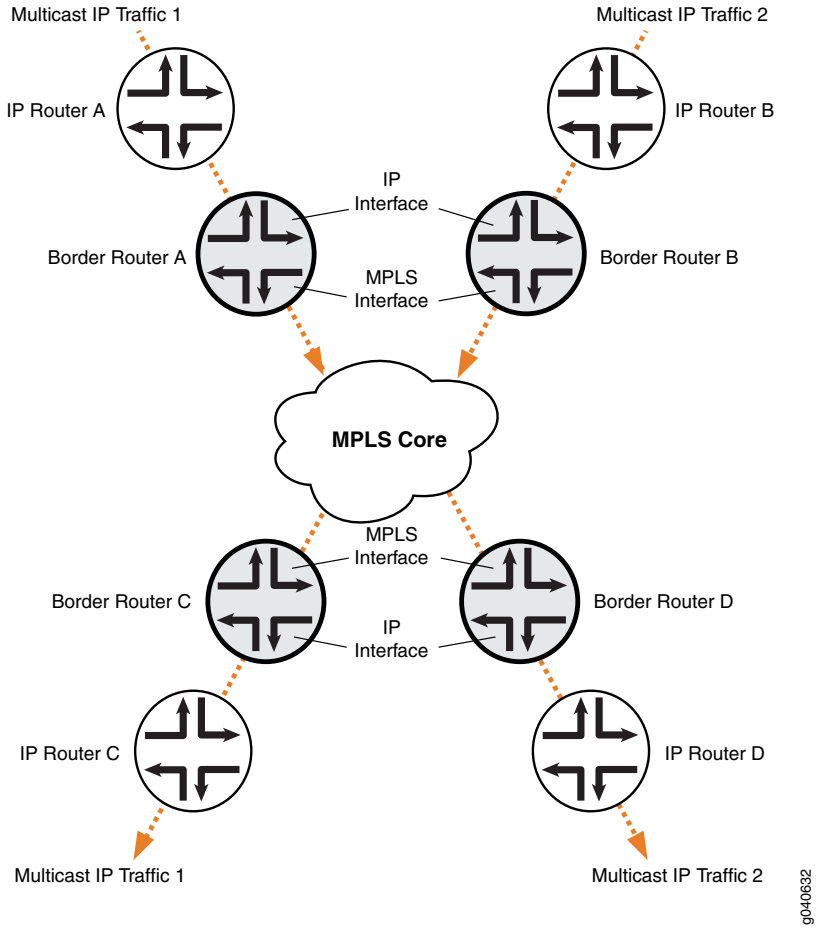
For each mpls-internet-multicast routing instance, the **ingress-replication** statement is required under the **provider-tunnel** statement and also under the **[edit routing-instances routing-instance-name provider-tunnel selective group source]** hierarchy level.

When a new destination needs to be added to the ingress replication provider tunnel, the resulting behavior differs depending on how the ingress replication provider tunnel is configured:

- create-new-ucast-tunnel—When this statement is configured, a new unicast tunnel to the destination is created, and is deleted when the destination is no longer needed. Use this mode for RSVP LSPs using ingress replication.
- label-switched-path-template (Multicast)—When this statement is configured, an LSP template is used for the for the point-to-multipoint LSP for ingress replication.

The IP topology consists of routers on the edge of the IP multicast domain. Each router has a set of IP interfaces configured toward the MPLS cloud and a set of interfaces configured toward the IP routers. See Figure 100 on page 593. Internet multicast traffic is carried between the IP routers, through the MPLS cloud, using ingress replication tunnels for the data plane and a full-mesh IBGP session for the control plane.

Figure 100: Internet Multicast Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Border Router C

```
set protocols mpls ipv6-tunneling
set protocols mpls interface all
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 10.255.10.61
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet6 unicast
set protocols bgp group ibgp family inet6-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp family inet6-mvpn signaling
set protocols bgp group ibgp export to-bgp
set protocols bgp group ibgp neighbor 10.255.10.97
set protocols bgp group ibgp neighbor 10.255.10.55
```

```
set protocols bgp group ibgp neighbor 10.255.10.57
set protocols bgp group ibgp neighbor 10.255.10.59
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ospf area 0.0.0.0 interface so-1/3/1.0
set protocols ospf area 0.0.0.0 interface so-0/3/0.0
set protocols ospf3 area 0.0.0.0 interface lo0.0
set protocols ospf3 area 0.0.0.0 interface so-1/3/1.0
set protocols ospf3 area 0.0.0.0 interface so-0/3/0.0
set protocols ldp interface all
set protocols pim rp static address 192.0.2.2
set protocols pim rp static address 2::192.0.2.2
set protocols pim interface fe-0/1/0.0
set protocols pim mpls-internet-multicast
set routing-instances test instance-type mpls-internet-multicast
set routing-instances test provider-tunnel ingress-replication label-switched-path
set routing-instances test protocols mvpn
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

The following example shows how to configure ingress replication on an IP multicast instance with the routing instance type **mpls-internet-multicast**. Additionally, this example shows how to configure a selective provider tunnel that selects a new unicast tunnel each time a new destination needs to be added to the multicast distribution tree.

This example shows the configuration of the link between Border Router C and edge IP Router C, from which Border Router C receives PIM join messages.

1. Enable MPLS.

```
[edit protocols mpls]
user@Border_Router_C# set ipv6-tunneling
user@Border_Router_C# set interface all
```

2. Configure a signaling protocol, such as RSVP or LDP.

```
[edit protocols ldp]
user@Border_Router_C# set interface all
```

3. Configure a full-mesh of IBGP peering sessions.

```
[edit protocols bgp group ibgp]
user@Border_Router_C# set type internal
user@Border_Router_C# set local-address 10.255.10.61
user@Border_Router_C# set neighbor 10.255.10.97
user@Border_Router_C# set neighbor 10.255.10.55
user@Border_Router_C# set neighbor 10.255.10.57
user@Border_Router_C# set neighbor 10.255.10.59
user@Border_Router_C# set export to-bgp
```

4. Configure the multiprotocol BGP-related settings so that the BGP sessions carry the necessary NLRI.

```
[edit protocols bgp group ibgp]
user@Border_Router_C# set family inet unicast
user@Border_Router_C# set family inet-vpn any
user@Border_Router_C# set family inet6 unicast
user@Border_Router_C# set family inet6-vpn any
user@Border_Router_C# set family inet-mvpn signaling
user@Border_Router_C# set family inet6-mvpn signaling
```

5. Configure an interior gateway protocol (IGP).

   This example shows a dual stacking configuration with OSPF and OSPF version 3 configured on the interfaces.

```
[edit protocols ospf3]
user@Border_Router_C# set area 0.0.0.0 interface lo0.0
user@Border_Router_C# set area 0.0.0.0 interface so-1/3/1.0
user@Border_Router_C# set area 0.0.0.0 interface so-0/3/0.0
```

```
[edit protocols ospf]
user@Border_Router_C# set traffic-engineering
user@Border_Router_C# set area 0.0.0.0 interface fxp0.0 disable
user@Border_Router_C# set area 0.0.0.0 interface lo0.0
user@Border_Router_C# set area 0.0.0.0 interface so-1/3/1.0
user@Border_Router_C# set area 0.0.0.0 interface so-0/3/0.0
```

6. Configure a global PIM instance on the interface facing the edge device.

   PIM is not configured in the core.

```
[edit protocols pim]
user@Border_Router_C# set rp static address 192.0.2.2
user@Border_Router_C# set rp static address 2::192.0.2.2
user@Border_Router_C# set interface fe-0/1/0.0
user@Border_Router_C# set mpls-internet-multicast
```

7. Configure the ingress replication provider tunnel to create a new unicast tunnel each time a destination needs to be added to the multicast distribution tree.

```
[edit routing-instances test]
user@Border_Router_C# set instance-type mpls-internet-multicast
user@Border_Router_C# set provider-tunnel ingress-replication label-switched-path
user@Border_Router_C# set protocols mvpn
```

NOTE: Alternatively, use the **label-switched-path-template** statement to configure a point-to-point LSP for the ingress tunnel.

Configure the point-to-point LSP to use the default template settings (this is needed only when using RSVP tunnels). For example:

```
[edit routing-instances test provider-tunnel]
user@Border_Router_C# set ingress-replication label-switched-path
  label-switched-path-template default-template
user@Border_Router_C# set selective group 203.0.113.0/24 source
  192.168.195.145/32 ingress-replication label-switched-path
```

8. Commit the configuration.

```
user@Border_Router_C# commit
```

Results From configuration mode, confirm your configuration by issuing the **show protocols** and **show routing-instances** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@Border_Router_C# show protocols
mpls {
  ipv6-tunneling;
  interface all;
}
bgp {
  group ibgp {
    type internal;
    local-address 10.255.10.61;
    family inet {
      unicast;
    }
    family inet-vpn {
      any;
    }
    family inet6 {
      unicast;
    }
    family inet6-vpn {
      any;
    }
    family inet-mvpn {
      signaling;
    }
    family inet6-mvpn {
      signaling;
    }
    export to-bgp; ## 'to-bgp' is not defined
    neighbor 10.255.10.97;
    neighbor 10.255.10.55;
```

```
      neighbor 10.255.10.57;
      neighbor 10.255.10.59;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface fxp0.0 {
            disable;
        }
        interface lo0.0;
        interface so-1/3/1.0;
        interface so-0/3/0.0;
    }
}
ospf3 {
    area 0.0.0.0 {
        interface lo0.0;
        interface so-1/3/1.0;
        interface so-0/3/0.0;
    }
}
ldp {
    interface all;
}
pim {
    rp {
        static {
            address 192.0.2.2;
            address 2::192.0.2.2;
        }
    }
    interface fe-0/1/0.0;
    mpls-internet-multicast;
}
```

```
user@Border_Router_C# show routing-instances
test {
    instance-type mpls-internet-multicast;
    provider-tunnel {
        ingress-replication {
            label-switched-path;
        }
    }
    protocols {
        mvpn;
    }
}
```

Verification

Confirm that the configuration is working properly. The following operational output is for LDP ingress replication SPT-only mode. The multicast source behind IP Router B. The multicast receiver is behind IP Router C.

Checking the Ingress Replication Status on Border Router C

Purpose Use the **show ingress-replication mvpn** command to check the ingress replication status.

Action

```
user@Border_Router_C> show ingress-replication mvpn

Ingress Tunnel: mvpn:1
  Application: MVPN
  Unicast tunnels
    Leaf Address       Tunnel-type      Mode         State
    10.255.10.61       P2P LSP          Existing     Up
```

Meaning The ingress replication is using a point-to-point LSP, and is in the Up state.

Checking the Routing Table for the MVPN Routing Instance on Border Router C

Purpose Use the **show route table** command to check the route status.

Action

```
user@Border_Router_C> show route table test.mvpn

test.mvpn.0: 5 destinations, 7 routes (5 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:0:0:10.255.10.61/240
                   *[BGP/170] 00:45:55, localpref 100, from 10.255.10.61
                      AS path: I, validation-state: unverified
                    > via so-2/0/1.0
1:0:0:10.255.10.97/240
                   *[MVPN/70] 00:47:19, metric2 1
                      Indirect
5:0:0:32:192.168.195.106:32:198.51.100.1/240
                   *[PIM/105] 00:06:35
                      Multicast (IPv4) Composite
                    [BGP/170] 00:06:35, localpref 100, from 10.255.10.61
                      AS path: I, validation-state: unverified
                    > via so-2/0/1.0
6:0:0:1000:32:192.0.2.2:32:198.51.100.1/240
                   *[PIM/105] 00:07:03
                      Multicast (IPv4) Composite
7:0:0:1000:32:192.168.195.106:32:198.51.100.1/240
                   *[MVPN/70] 00:06:35, metric2 1
                      Multicast (IPv4) Composite
                    [PIM/105] 00:05:35
                      Multicast (IPv4) Composite

test.mvpn-inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:0:0:10.255.10.61/432
                   *[BGP/170] 00:45:55, localpref 100, from 10.255.10.61
                      AS path: I, validation-state: unverified
                    > via so-2/0/1.0
1:0:0:10.255.10.97/432
                   *[MVPN/70] 00:47:19, metric2 1
                      Indirect
```

Meaning The expected routes are populating the test.mvpn routing table.

Checking the MVPN Neighbors on Border Router C

Purpose Use the **show mvpn neighbor** command to check the neighbor status.

Action

```
user@Border_Router_C> show mvpn neighbor

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : test
  MVPN Mode : SPT-ONLY
  Neighbor                              Inclusive Provider Tunnel
  10.255.10.61                          INGRESS-REPLICATION:MPLS Label
16:10.255.10.61

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET6

Instance : test
  MVPN Mode : SPT-ONLY
  Neighbor                              Inclusive Provider Tunnel
  10.255.10.61                          INGRESS-REPLICATION:MPLS Label
16:10.255.10.61
```

Checking the PIM Join Status on Border Router C

Purpose Use the **show pim join extensive** command to check the PIM join status.

Action

```
user@Border_Router_C> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 198.51.100.1
    Source: *
    RP: 192.0.2.2
    Flags: sparse,rptree,wildcard
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local RP
    Uptime: 00:07:49
    Downstream neighbors:
        Interface: ge-3/0/6.0
            192.0.2.2 State: Join Flags: SRW  Timeout: Infinity
            Uptime: 00:07:49 Time since last Join: 00:07:49
    Number of downstream interfaces: 1

Group: 198.51.100.1
    Source: 192.168.195.106
    Flags: sparse
    Upstream protocol: BGP
    Upstream interface: Through BGP
    Upstream neighbor: Through MVPN
    Upstream state: Local RP, Join to Source, No Prune to RP
    Keepalive timeout: 69
    Uptime: 00:06:21
    Number of downstream interfaces: 0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

Checking the Multicast Route Status on Border Router C

Purpose Use the **show multicast route extensive** command to check the multicast route status.

Action

```
user@Border_Router_C> show multicast route extensive
Instance: master Family: INET

Group: 198.51.100.1
    Source: 192.168.195.106/32
    Upstream interface: lsi.0
    Downstream interface list:
        ge-3/0/6.0
    Number of outgoing interfaces: 1
    Session description: NOB Cross media facilities
    Statistics: 18 kBps, 200 pps, 88907 packets
    Next-hop ID: 1048577
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:07:25

Instance: master Family: INET6
```

Checking the Ingress Replication Status on Border Router B

Purpose Use the **show ingress-replication mvpn** command to check the ingress replication status.

Action

```
user@Border_Router_B> show ingress-replication mvpn

Ingress Tunnel: mvpn:1
  Application: MVPN
  Unicast tunnels
    Leaf Address       Tunnel-type      Mode         State
    10.255.10.97       P2P LSP          Existing     Up
```

Meaning The ingress replication is using a point-to-point LSP, and is in the Up state.

Checking the Routing Table for the MVPN Routing Instance on Border Router B

Purpose Use the **show route table** command to check the route status.

Action

```
user@Border_Router_B> show route table test.mvpn

test.mvpn.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:0:0:10.255.10.61/240
                   *[MVPN/70] 00:49:26, metric2 1
                      Indirect
1:0:0:10.255.10.97/240
                   *[BGP/170] 00:48:22, localpref 100, from 10.255.10.97
                      AS path: I, validation-state: unverified
                    > via so-1/3/1.0
5:0:0:32:192.168.195.106:32:198.51.100.1/240
                   *[PIM/105] 00:09:02
                      Multicast (IPv4) Composite
                    [BGP/170] 00:09:02, localpref 100, from 10.255.10.97
                      AS path: I, validation-state: unverified
                    > via so-1/3/1.0
7:0:0:1000:32:192.168.195.106:32:198.51.100.1/240
                   *[PIM/105] 00:09:02
                      Multicast (IPv4) Composite
                    [BGP/170] 00:09:02, localpref 100, from 10.255.10.97
                      AS path: I, validation-state: unverified
                    > via so-1/3/1.0

test.mvpn-inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:0:0:10.255.10.61/432
                   *[MVPN/70] 00:49:26, metric2 1
                      Indirect
1:0:0:10.255.10.97/432
                   *[BGP/170] 00:48:22, localpref 100, from 10.255.10.97
                      AS path: I, validation-state: unverified
                    > via so-1/3/1.0
```

Meaning The expected routes are populating the test.mvpn routing table.

Checking the MVPN Neighbors on Border Router B

Purpose Use the **show mvpn neighbor** command to check the neighbor status.

Action

```
user@Border_Router_B> show mvpn neighbor

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : test
  MVPN Mode : SPT-ONLY
  Neighbor                              Inclusive Provider Tunnel
  10.255.10.97                          INGRESS-REPLICATION:MPLS Label
16:10.255.10.97

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET6

Instance : test
  MVPN Mode : SPT-ONLY
  Neighbor                              Inclusive Provider Tunnel
  10.255.10.97                          INGRESS-REPLICATION:MPLS Label
16:10.255.10.97
```

Checking the PIM Join Status on Border Router B

Purpose Use the **show pim join extensive** command to check the PIM join status.

Action

```
user@Border_Router_B> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 198.51.100.1
    Source: 192.168.195.106
    Flags: sparse,spt
    Upstream interface: fe-0/1/0.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout: 0
    Uptime: 00:09:39
    Downstream neighbors:
        Interface: Pseudo-MVPN
            Uptime: 00:09:39 Time since last Join: 00:09:39
    Number of downstream interfaces: 1


Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

Checking the Multicast Route Status on Border Router B

Purpose Use the **show multicast route extensive** command to check the multicast route status.

Action

```
user@Border_Router_B> show multicast route extensive
Instance: master Family: INET

Group: 198.51.100.1
    Source: 192.168.195.106/32
    Upstream interface: fe-0/1/0.0
    Downstream interface list:
        so-1/3/1.0
    Number of outgoing interfaces: 1
    Session description: NOB Cross media facilities
    Statistics: 18 kBps, 200 pps, 116531 packets
    Next-hop ID: 1048580
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:09:43
```

See Also

- *Configuring Routing Instances for an MBGP MVPN*
- mpls-internet-multicast on page 1165
- ingress-replication on page 1096
- create-new-ucast-tunnel on page 994
- label-switched-path-template (Multicast) on page 1121

- show ingress-replication mvpn on page 1496

Example: Configuring MBGP Multicast VPNs

This example provides a step-by-step procedure to configure multicast services across a multiprotocol BGP (MBGP) Layer 3 virtual private network. (also referred to as next-generation Layer 3 multicast VPNs)

Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.2 or later
- Five M Series, T Series, TX Series, or MX Series Juniper routers
- One host system capable of sending multicast traffic and supporting the Internet Group Management Protocol (IGMP)
- One host system capable of receiving multicast traffic and supporting IGMP

Depending on the devices you are using, you might be required to configure static routes to:

- The multicast sender
- The Fast Ethernet interface to which the sender is connected on the multicast receiver
- The multicast receiver
- The Fast Ethernet interface to which the receiver is connected on the multicast sender

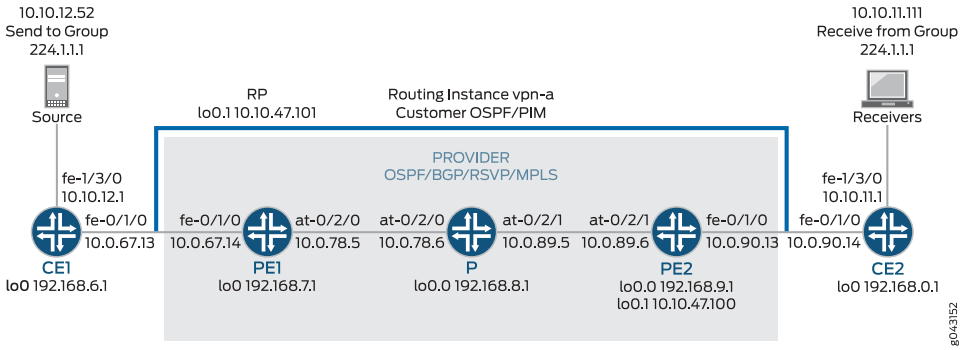
Overview and Topology

This example shows how to configure the following technologies:

- IPv4
- BGP
- OSPF
- RSVP
- MPLS
- PIM sparse mode
- Static RP

The topology of the network is shown in Figure 101 on page 607.

Figure 101: Multicast Over Layer 3 VPN Example Topology



Configuration

NOTE: In any configuration session, it is a good practice to periodically verify that the configuration can be committed using the **commit check** command.

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **P** identifies the provider core (P) router
- **CE2** identifies the customer edge 2 (CE2) router
- **PE2** identifies the provider edge 2 (PE2) router

To configure MBGP multicast VPNs for the network shown in Figure 101 on page 607, perform the following steps:

Configuring Interfaces

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

1. On each router, configure an IP address on the loopback logical interface 0 (**lo0.0**).

   ```
   [edit interfaces]
   user@CE1# set lo0 unit 0 family inet address 192.168.6.1/32 primary

   user@PE1# set lo0 unit 0 family inet address 192.168.7.1/32 primary

   user@P# set lo0 unit 0 family inet address 192.168.8.1/32 primary

   user@PE2# set lo0 unit 0 family inet address 192.168.9.1/32 primary

   user@CE2# set lo0 unit 0 family inet address 192.168.0.1/32 primary
   ```

   Use the **show interfaces terse** command to verify that the IP address is correct on the loopback logical interface.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet interfaces. Specify the **inet** protocol family type.

   ```
   [edit interfaces]
   user@CE1# set fe-1/3/0 unit 0 family inet address 10.10.12.1/24
   user@CE1# set fe-0/1/0 unit 0 family inet address 10.0.67.13/30

   [edit interfaces]
   user@PE1# set fe-0/1/0 unit 0 family inet address 10.0.67.14/30

   [edit interfaces]
   user@PE2# set fe-0/1/0 unit 0 family inet address 10.0.90.13/30

   [edit interfaces]
   user@CE2# set fe-0/1/0 unit 0 family inet address 10.0.90.14/30
   user@CE2# set fe-1/3/0 unit 0 family inet address 10.10.11.1/24
   ```

   Use the **show interfaces terse** command to verify that the IP address is correct on the Fast Ethernet interfaces.

3. On the PE and P routers, configure the ATM interfaces' VPI and maximum virtual circuits. If the default PIC type is different on directly connected ATM interfaces, configure the PIC type to be the same. Configure the logical interface VCI, protocol family, local IP address, and destination IP address.

   ```
   [edit interfaces]
   user@PE1# set at-0/2/0 atm-options pic-type atm1
   user@PE1# set at-0/2/0 atm-options vpi 0 maximum-vcs 256
   user@PE1# set at-0/2/0 unit 0 vci 0.128
   user@PE1# set at-0/2/0 unit 0 family inet address 10.0.78.5/32 destination 10.0.78.6
   ```

```
[edit interfaces]
user@P# set at-0/2/0 atm-options pic-type atm1
user@P# set at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@P# set at-0/2/0 unit 0 vci 0.128
user@P# set at-0/2/0 unit 0 family inet address 10.0.78.6/32 destination 10.0.78.5
user@P# set at-0/2/1 atm-options pic-type atm1
user@P# set at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@P# set at-0/2/1 unit 0 vci 0.128
user@P# set at-0/2/1 unit 0 family inet address 10.0.89.5/32 destination 10.0.89.6
```

```
[edit interfaces]
user@PE2# set at-0/2/1 atm-options pic-type atm1
user@PE2# set at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@PE2# set at-0/2/1 unit 0 vci 0.128
user@PE2# set at-0/2/1 unit 0 family inet address 10.0.89.6/32 destination 10.0.89.5
```

Use the **show configuration interfaces** command to verify that the ATM interfaces' VPI and maximum VCs are correct and that the logical interface VCI, protocol family, local IP address, and destination IP address are correct.

Configuring OSPF

Step-by-Step Procedure

1. On the P and PE routers, configure the provider instance of OSPF. Specify the **lo0.0** and ATM core-facing logical interfaces. The provider instance of OSPF on the PE router forms adjacencies with the OSPF neighbors on the other PE router and Router P.

```
user@PE1# set protocols ospf area 0.0.0.0 interface at-0/2/0.0
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.0
```

```
user@P# set protocols ospf area 0.0.0.0 interface lo0.0
user@P# set protocols ospf area 0.0.0.0 interface all
user@P# set protocols ospf area 0.0.0.0 interface fxp0 disable
```

```
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0
user@PE2# set protocols ospf area 0.0.0.0 interface at-0/2/1.0
```

Use the **show ospf interfaces** command to verify that the **lo0.0** and ATM core-facing logical interfaces are configured for OSPF.

2. On the CE routers, configure the customer instance of OSPF. Specify the loopback and Fast Ethernet logical interfaces. The customer instance of OSPF on the CE routers form adjacencies with the neighbors within the VPN routing instance of OSPF on the PE routers.

```
user@CE1# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@CE1# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@CE1# set protocols ospf area 0.0.0.0 interface lo0.0
```

```
user@CE2# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@CE2# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@CE2# set protocols ospf area 0.0.0.0 interface lo0.0
```

Use the **show ospf interfaces** command to verify that the correct loopback and Fast Ethernet logical interfaces have been added to the OSPF protocol.

3. On the P and PE routers, configure OSPF traffic engineering support for the provider instance of OSPF.

   The **shortcuts** statement enables the master instance of OSPF to use a label-switched path as the next hop.

   ```
   user@PE1# set protocols ospf traffic-engineering shortcuts

   user@P# set protocols ospf traffic-engineering shortcuts

   user@PE2# set protocols ospf traffic-engineering shortcuts
   ```

   Use the **show ospf overview** or **show configuration protocols ospf** command to verify that traffic engineering support is enabled.

Configuring BGP

Step-by-Step Procedure

1. On Router P, configure BGP for the VPN. The local address is the local **lo0.0** address. The neighbor addresses are the PE routers' **lo0.0** addresses.

   The **unicast** statement enables the router to use BGP to advertise network layer reachability information (NLRI). The **signaling** statement enables the router to use BGP as the signaling protocol for the VPN.

   ```
   user@P# set protocols bgp group group-mvpn type internal
   user@P# set protocols bgp group group-mvpn local-address 192.168.8.1
   user@P# set protocols bgp group group-mvpn family inet unicast
   user@P# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@P# set protocols bgp group group-mvpn neighbor 192.168.9.1
   user@P# set protocols bgp group group-mvpn neighbor 192.168.7.1
   ```

   Use the **show configuration protocols bgp** command to verify that the router has been configured to use BGP to advertise NLRI.

2. On the PE and P routers, configure the BGP local autonomous system number.

   ```
   user@PE1# set routing-options autonomous-system 0.65010

   user@P# set routing-options autonomous-system 0.65010

   user@PE2# set routing-options autonomous-system 0.65010
   ```

   Use the **show configuration routing-options** command to verify that the BGP local autonomous system number is correct.

3. On the PE routers, configure BGP for the VPN. Configure the local address as the local **lo0.0** address. The neighbor addresses are the **lo0.0** addresses of Router P and the other PE router, PE2.

```
user@PE1# set protocols bgp group group-mvpn type internal
user@PE1# set protocols bgp group group-mvpn local-address 192.168.7.1
user@PE1# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE1# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE1# set protocols bgp group group-mvpn neighbor 192.168.9.1
user@PE1# set protocols bgp group group-mvpn neighbor 192.168.8.1

user@PE2# set protocols bgp group group-mvpn type internal
user@PE2# set protocols bgp group group-mvpn local-address 192.168.9.1
user@PE2# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE2# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE2# set protocols bgp group group-mvpn neighbor 192.168.7.1
user@PE2# set protocols bgp group group-mvpn neighbor 192.168.8.1
```

Use the **show bgp group** command to verify that the BGP configuration is correct.

4. On the PE routers, configure a policy to export the BGP routes into OSPF.

```
user@PE1# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@PE1# set policy-options policy-statement bgp-to-ospf then accept

user@PE2# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@PE2# set policy-options policy-statement bgp-to-ospf then accept
```

Use the **show policy bgp-to-ospf** command to verify that the policy is correct.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, enable RSVP on the interfaces that participate in the LSP. Configure the Fast Ethernet and ATM logical interfaces.

```
user@PE1# set protocols rsvp interface fe-0/1/0.0
user@PE1# set protocols rsvp interface at-0/2/0.0

user@PE2# set protocols rsvp interface fe-0/1/0.0
user@PE2# set protocols rsvp interface at-0/2/1.0
```

2. On Router P, enable RSVP on the interfaces that participate in the LSP. Configure the ATM logical interfaces.

```
user@P# set protocols rsvp interface at-0/2/0.0
user@P# set protocols rsvp interface at-0/2/1.0
```

Use the **show configuration protocols rsvp** command to verify that the RSVP configuration is correct.

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure an MPLS LSP to the PE router that is the LSP egress point. Specify the IP address of the **lo0.0** interface on the router at the other end of the LSP. Configure MPLS on the ATM, Fast Ethernet, and **lo0.0** interfaces.

To help identify each LSP when troubleshooting, configure a different LSP name on each PE router. In this example, we use the name **to-pe2** as the name for the LSP configured on PE1 and **to-pe1** as the name for the LSP configured on PE2.

```
user@PE1# set protocols mpls label-switched-path to-pe2 to 192.168.9.1
user@PE1# set protocols mpls interface fe-0/1/0.0
user@PE1# set protocols mpls interface at-0/2/0.0
user@PE1# set protocols mpls interface lo0.0

user@PE2# set protocols mpls label-switched-path to-pe1 to 192.168.7.1
user@PE2# set protocols mpls interface fe-0/1/0.0
user@PE2# set protocols mpls interface at-0/2/1.0
user@PE2# set protocols mpls interface lo0.0
```

Use the **show configuration protocols mpls** and **show route label-switched-path to-pe1** commands to verify that the MPLS and LSP configuration is correct.

After the configuration is committed, use the **show mpls lsp name to-pe1** and **show mpls lsp name to-pe2** commands to verify that the LSP is operational.

2. On Router P, enable MPLS. Specify the ATM interfaces connected to the PE routers.

```
user@P# set protocols mpls interface at-0/2/0.0
user@P# set protocols mpls interface at-0/2/1.0
```

Use the **show mpls interface** command to verify that MPLS is enabled on the ATM interfaces.

3. On the PE and P routers, configure the protocol family on the ATM interfaces associated with the LSP. Specify the **mpls** protocol family type.

```
user@PE1# set interfaces at-0/2/0 unit 0 family mpls

user@P# set interfaces at-0/2/0 unit 0 family mpls
user@P# set interfaces at-0/2/1 unit 0 family mpls

user@PE2# set interfaces at-0/2/1 unit 0 family mpls
```

Use the **show mpls interface** command to verify that the MPLS protocol family is enabled on the ATM interfaces associated with the LSP.

Configuring the VRF Routing Instance

Step-by-Step Procedure

1. On the PE routers, configure a routing instance for the VPN and specify the **vrf** instance type. Add the Fast Ethernet and **lo0.1** customer-facing interfaces. Configure the VPN instance of OSPF and include the BGP-to-OSPF export policy.

```
user@PE1# set routing-instances vpn-a instance-type vrf
user@PE1# set routing-instances vpn-a interface lo0.1
user@PE1# set routing-instances vpn-a interface fe-0/1/0.0
user@PE1# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@PE1# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface all
```

```
user@PE2# set routing-instances vpn-a instance-type vrf
user@PE2# set routing-instances vpn-a interface lo0.1
user@PE2# set routing-instances vpn-a interface fe-0/1/0.0
user@PE2# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@PE2# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface all
```

Use the **show configuration routing-instances vpn-a** command to verify that the routing instance configuration is correct.

2. On the PE routers, configure a route distinguisher for the routing instance. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each PE router. This example uses 65010:1 on PE1 and 65010:2 on PE2.

```
user@PE1# set routing-instances vpn-a route-distinguisher 65010:1
```

```
user@PE2# set routing-instances vpn-a route-distinguisher 65010:2
```

Use the **show configuration routing-instances vpn-a** command to verify that the route distinguisher is correct.

3. On the PE routers, configure default VRF import and export policies. Based on this configuration, BGP automatically generates local routes corresponding to the route target referenced in the VRF import policies. This example uses 2:1 as the route target.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@PE1# set routing-instances vpn-a vrf-target target:2:1
```

```
user@PE2# set routing-instances vpn-a vrf-target target:2:1
```

Use the **show configuration routing-instances vpn-a** command to verify that the route target is correct.

4. On the PE routers, configure the VPN routing instance for multicast support.

```
user@PE1# set routing-instances vpn-a protocols mvpn
```

```
user@PE2# set routing-instances vpn-a protocols mvpn
```

Use the **show configuration routing-instance vpn-a** command to verify that the VPN routing instance has been configured for multicast support.

5. On the PE routers, configure an IP address on loopback logical interface 1 (**lo0.1**) used in the customer routing instance VPN.

```
user@PE1# set interfaces lo0 unit 1 family inet address 10.10.47.101/32
```

```
user@PE2# set interfaces lo0 unit 1 family inet address 10.10.47.100/32
```

Use the **show interfaces terse** command to verify that the IP address on the loopback interface is correct.

Configuring PIM

Step-by-Step Procedure

1. On the PE routers, enable PIM. Configure the **lo0.1** and the customer-facing Fast Ethernet interface. Specify the mode as **sparse** and the version as **2**.

```
user@PE1# set routing-instances vpn-a protocols pim interface lo0.1 mode sparse
user@PE1# set routing-instances vpn-a protocols pim interface lo0.1 version 2
user@PE1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 mode
  sparse
user@PE1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 version
  2
```

```
user@PE2# set routing-instances vpn-a protocols pim interface lo0.1 mode sparse
user@PE2# set routing-instances vpn-a protocols pim interface lo0.1 version 2
user@PE2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 mode
  sparse
user@PE2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 version
  2
```

   Use the **show pim interfaces instance vpn-a** command to verify that PIM sparse-mode is enabled on the **lo0.1** interface and the customer-facing Fast Ethernet interface.

2. On the CE routers, enable PIM. In this example, we configure all interfaces. Specify the mode as **sparse** and the version as **2**.

```
user@CE1# set protocols pim interface all
```

```
user@CE2# set protocols pim interface all mode sparse
user@CE2# set protocols pim interface all version 2
```

   Use the **show pim interfaces** command to verify that PIM sparse mode is enabled on all interfaces.

Configuring the Provider Tunnel

Step-by-Step Procedure

1. On Router PE1, configure the provider tunnel. Specify the multicast address to be used.

   The **provider-tunnel** statement instructs the router to send multicast traffic across a tunnel.

```
user@PE1# set routing-instances vpn-a provider-tunnel rsvp-te
  label-switched-path-template default-template
```

   Use the **show configuration routing-instance vpn-a** command to verify that the provider tunnel is configured to use the default LSP template.

2. On Router PE2, configure the provider tunnel. Specify the multicast address to be used.

   ```
   user@PE2# set routing-instances vpn-a provider-tunnel rsvp-te
     label-switched-path-template default-template
   ```

   Use the **show configuration routing-instance vpn-a** command to verify that the provider tunnel is configured to use the default LSP template.

Configuring the Rendezvous Point

Step-by-Step Procedure

1. Configure Router PE1 to be the rendezvous point. Specify the **lo0.1** address of Router PE1. Specify the multicast address to be used.

   ```
   user@PE1# set routing-instances vpn-a protocols pim rp local address 10.10.47.101
   user@PE1# set routing-instances vpn-a protocols pim rp local group-ranges
     224.1.1.1/32
   ```

   Use the **show pim rps instance vpn-a** command to verify that the correct local IP address is configured for the RP.

2. On Router PE2, configure the static rendezvous point. Specify the **lo0.1** address of Router PE1.

   ```
   user@PE2# set routing-instances vpn-a protocols pim rp static address 10.10.47.101
   ```

   Use the **show pim rps instance vpn-a** command to verify that the correct static IP address is configured for the RP.

3. On the CE routers, configure the static rendezvous point. Specify the **lo0.1** address of Router PE1.

   ```
   user@CE1# set protocols pim rp static address 10.10.47.101 version 2

   user@CE2# set protocols pim rp static address 10.10.47.101 version 2
   ```

   Use the **show pim rps** command to verify that the correct static IP address is configured for the RP.

4. Use the **commit check** command to verify that the configuration can be successfully committed. If the configuration passes the check, commit the configuration.

5. Start the multicast sender device connected to CE1.

6. Start the multicast receiver device connected to CE2.

7. Verify that the receiver is receiving the multicast stream.

8. Use **show** commands to verify the routing, VPN, and multicast operation.

Results

The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router CE1 follows.

Router CE1

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.6.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.67.13/30;
            }
        }
    }
    fe-1/3/0 {
        unit 0 {
            family inet {
                address 10.10.12.1/24;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface lo0.0;
            interface fe-1/3/0.0;
        }
    }
    pim {
        rp {
            static {
                address 10.10.47.101 {
                    version 2;
                }
            }
        }
        interface all;
    }
}
```

The relevant sample configuration for Router PE1 follows.

Router PE1

```
interfaces {
```

```
    lo0 {
      unit 0 {
        family inet {
          address 192.168.7.1/32 {
            primary;
          }
        }
      }
    }
    fe-0/1/0 {
      unit 0 {
        family inet {
          address 10.0.67.14/30;
        }
      }
    }
    at-0/2/0 {
      atm-options {
        pic-type atm1;
        vpi 0 {
          maximum-vcs 256;
        }
      }
      unit 0 {
        vci 0.128;
        family inet {
          address 10.0.78.5/32 {
            destination 10.0.78.6;
          }
        }
        family mpls;
      }
    }
    lo0 {
      unit 1 {
        family inet {
          address 10.10.47.101/32;
        }
      }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
      interface fe-0/1/0.0;
      interface at-0/2/0.0;
    }
    mpls {
      label-switched-path to-pe2 {
        to 192.168.9.1;
      }
      interface fe-0/1/0.0;
      interface at-0/2/0.0;
      interface lo0.0;
```

```
    }
    bgp {
      group group-mvpn {
        type internal;
        local-address 192.168.7.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.9.1;
        neighbor 192.168.8.1;
      }
    }
    ospf {
      traffic-engineering {
        shortcuts;
      }
      area 0.0.0.0 {
        interface at-0/2/0.0;
        interface lo0.0;
      }
    }
}
policy-options {
    policy-statement bgp-to-ospf {
      from protocol bgp;
      then accept;
    }
}
routing-instances {
    vpn-a {
      instance-type vrf;
      interface lo0.1;
      interface fe-0/1/0.0;
      route-distinguisher 65010:1;
      provider-tunnel {
        rsvp-te {
          label-switched-path-template {
            default-template;
          }
        }
      }
      vrf-target target:2:1;
      protocols {
        ospf {
          export bgp-to-ospf;
          area 0.0.0.0 {
            interface all;
          }
        }
        pim {
          rp {
            local {
              address 10.10.47.101;
```

```
            group-ranges {
              224.1.1.1/32;
            }
          }
        }
        interface lo0.1 {
          mode sparse;
          version 2;
        }
        interface fe-0/1/0.0 {
          mode sparse;
          version 2;
        }
      }
      mvpn;
    }
  }
}
```

The relevant sample configuration for Router P follows.

Router P

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.8.1/32 {
          primary;
        }
      }
    }
  }
  at-0/2/0 {
    atm-options {
      pic-type atm1;
      vpi 0 {
        maximum-vcs 256;
      }
    }
    unit 0 {
      vci 0.128;
      family inet {
        address 10.0.78.6/32 {
          destination 10.0.78.5;
        }
      }
      family mpls;
    }
  }
  at-0/2/1 {
    atm-options {
      pic-type atm1;
      vpi 0 {
        maximum-vcs 256;
      }
    }
    unit 0 {
```

```
        vci 0.128;
        family inet {
          address 10.0.89.5/32 {
            destination 10.0.89.6;
          }
        }
        family mpls;
      }
    }
  }
  routing-options {
    autonomous-system 0.65010;
  }
  protocols {
    rsvp {
      interface at-0/2/0.0;
      interface at-0/2/1.0;
    }
    mpls {
      interface at-0/2/0.0;
      interface at-0/2/1.0;
    }
    bgp {
      group group-mvpn {
        type internal;
        local-address 192.168.8.1;
        family inet {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.9.1;
        neighbor 192.168.7.1;
      }
    }
    ospf {
      traffic-engineering {
        shortcuts;
      }
      area 0.0.0.0 {
        interface lo0.0;
        interface all;
        interface fxp0.0 {
          disable;
        }
      }
    }
  }
```

The relevant sample configuration for Router PE2 follows.

Router PE2

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
```

```
            address 192.168.9.1/32 {
              primary;
            }
          }
        }
      }
      fe-0/1/0 {
        unit 0 {
          family inet {
            address 10.0.90.13/30;
          }
        }
      }
      at-0/2/1 {
        atm-options {
          pic-type atm1;
          vpi 0 {
            maximum-vcs 256;
          }
        }
        unit 0 {
          vci 0.128;
          family inet {
            address 10.0.89.6/32 {
              destination 10.0.89.5;
            }
          }
          family mpls;
        }
      }
      lo0 {
        unit 1 {
          family inet {
            address 10.10.47.100/32;
          }
        }
      }
    }
    routing-options {
      autonomous-system 0.65010;
    }
    protocols {
      rsvp {
        interface fe-0/1/0.0;
        interface at-0/2/1.0;
      }
      mpls {
        label-switched-path to-pe1 {
          to 192.168.7.1;
        }
        interface lo0.0;
        interface fe-0/1/0.0;
        interface at-0/2/1.0;
      }
      bgp {
        group group-mvpn {
```

```
        type internal;
        local-address 192.168.9.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.7.1;
        neighbor 192.168.8.1;
      }
    }
    ospf {
      traffic-engineering {
        shortcuts;
      }
      area 0.0.0.0 {
        interface lo0.0;
        interface at-0/2/1.0;
      }
    }
}
policy-options {
  policy-statement bgp-to-ospf {
    from protocol bgp;
    then accept;
  }
}
routing-instances {
  vpn-a {
    instance-type vrf;
    interface fe-0/1/0.0;
    interface lo0.1;
    route-distinguisher 65010:2;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    vrf-target target:2:1;
    protocols {
      ospf {
        export bgp-to-ospf;
        area 0.0.0.0 {
          interface all;
        }
      }
      pim {
        rp {
          static {
            address 10.10.47.101;
          }
        }
        interface fe-0/1/0.0 {
```

```
            mode sparse;
            version 2;
          }
          interface lo0.1 {
            mode sparse;
            version 2;
          }
        }
        mvpn;
      }
    }
  }
```

The relevant sample configuration for Router CE2 follows.

Router CE2

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.0.1/32 {
          primary;
        }
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.0.90.14/30;
      }
    }
  }
  fe-1/3/0 {
    unit 0 {
      family inet {
        address 10.10.11.1/24;
      }
      family inet6 {
        address fe80::205:85ff:fe88:ccdb/64;
      }
    }
  }
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface fe-0/1/0.0;
      interface lo0.0;
      interface fe-1/3/0.0;
    }
  }
  pim {
    rp {
      static {
        address 10.10.47.101 {
          version 2;
```

```
        }
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
  }
}
```

See Also

Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN

This example shows how to configure a PIM-SSM provider tunnel for an MBGP MVPN. The configuration enables service providers to carry customer data in the core. This example shows how to configure PIM-SSM tunnels as inclusive PMSI and uses the unicast routing preference as the metric for determining the single forwarder (instead of the default metric, which is the IP address from the global administrator field in the route-import community).

Requirements

Before you begin:

- Configure the router interfaces. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure the BGP-to-OSPF routing policy. See the *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*.

Overview

When a PE receives a customer join or prune message from a CE, the message identifies a particular multicast flow as belonging either to a source-specific tree (S,G) or to a shared tree (*,G). If the route to the multicast source or RP is across the VPN backbone, then the PE needs to identify the upstream multicast hop (UMH) for the (S,G) or (*,G) flow. Normally the UMH is determined by the unicast route to the multicast source or RP.

However, in some cases, the CEs might be distributing to the PEs a special set of routes that are to be used exclusively for the purpose of upstream multicast hop selection using the route-import community. More than one route might be eligible, and the PE needs to elect a single forwarder from the eligible UMHs.

The default metric for the single forwarder election is the IP address from the global administrator field in the route-import community. You can configure a router to use the unicast route preference to determine the single forwarder election.

This example includes the following settings.

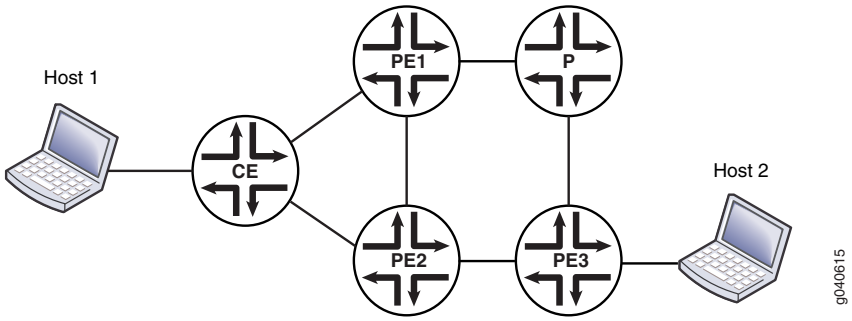
- **provider-tunnel family inet pim-ssm group-address**—Specifies a valid SSM VPN group address. The SSM VPN group address and the source address are advertised by the type-1 autodiscovery route. On receiving an autodiscovery route with the SSM VPN group address and the source address, a PE router sends an (S,G) join in the provider space to the PE advertising the autodiscovery route. All PE routers exchange their PIM-SSM VPN group address to complete the inclusive provider multicast service interface (I-PMSI). Unlike a PIM-ASM provider tunnel, the PE routers can choose a different VPN group address because the (S,G) joins are sent directly toward the source PE.

  NOTE: Similar to a PIM-ASM provider tunnel, PIM must be configured in the default master instance.

- **unicast-umh-election**—Specifies that the PE router uses the unicast route preference to determine the single-forwarder election.

Figure 102 on page 625 shows the topology used in this example.

Figure 102: PIM-SSM Provider Tunnel for an MBGP MVPN Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set interfaces fe-0/2/0 unit 0 family inet address 192.168.195.109/30
set interfaces fe-0/2/1 unit 0 family inet address 192.168.195.5/27
set interfaces fe-0/2/2 unit 0 family inet address 20.10.1.1/30
set interfaces fe-0/2/2 unit 0 family iso
set interfaces fe-0/2/2 unit 0 family mpls
set interfaces lo0 unit 1 family inet address 10.10.47.100/32
set interfaces lo0 unit 1 family inet address 1.1.1.1/32 primary
```

```
set interfaces lo0 unit 2 family inet address 10.10.48.100/32
set protocols mpls interface all set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-preference 120
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 10.255.112.155
set protocols isis level 1 disable set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ldp interface all
set protocols pim rp static address 10.255.112.155
set protocols pim interface all mode sparse-dense
set protocols pim interface all version 2
set protocols pim interface fxp0.0 disable
set routing-instances VPN-A instance-type vrf
set routing-instances VPN-A interface fe-0/2/1.0
set routing-instances VPN-A interface lo0.1
set routing-instances VPN-A route-distinguisher 10.255.112.199:100
set routing-instances VPN-A provider-tunnel family inet pim-ssm group-address 232.1.1.1
set routing-instances VPN-A vrf-target target:100:100
set routing-instances VPN-A vrf-table-label
set routing-instances VPN-A routing-options auto-export
set routing-instances VPN-A protocols ospf export bgp-to-ospf
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface lo0.1
set routing-instances VPN-A protocols ospf area 0.0.0.0 interface fe-0/2/1.0
set routing-instances VPN-A protocols pim rp static address 10.10.47.101
set routing-instances VPN-A protocols pim interface lo0.1 mode sparse-dense
set routing-instances VPN-A protocols pim interface lo0.1 version 2
set routing-instances VPN-A protocols pim interface fe-0/2/1.0 mode sparse-dense
set routing-instances VPN-A protocols pim interface fe-0/2/1.0 version 2
set routing-instances VPN-A protocols mvpn unicast-umh-election
set routing-instances VPN-B instance-type vrf
set routing-instances VPN-B interface fe-0/2/0.0
set routing-instances VPN-B interface lo0.2
set routing-instances VPN-B route-distinguisher 10.255.112.199:200
set routing-instances VPN-B provider-tunnel family inet pim-ssm group-address 232.2.2.2
set routing-instances VPN-B vrf-target target:200:200
set routing-instances VPN-B vrf-table-label
set routing-instances VPN-B routing-options auto-export
set routing-instances VPN-B protocols ospf export bgp-to-ospf
set routing-instances VPN-B protocols ospf area 0.0.0.0 interface lo0.2
set routing-instances VPN-B protocols ospf area 0.0.0.0 interface fe-0/2/0.0
set routing-instances VPN-B protocols pim rp static address 10.10.48.101
set routing-instances VPN-B protocols pim interface lo0.2 mode sparse-dense
set routing-instances VPN-B protocols pim interface lo0.2 version 2
set routing-instances VPN-B protocols pim interface fe-0/2/0.0 mode sparse-dense
set routing-instances VPN-B protocols pim interface fe-0/2/0.0 version 2
set routing-instances VPN-B protocols mvpn unicast-umh-election
set routing-options autonomous-system 100
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a PIM-SSM provider tunnel for an MBGP MVPN:

1. Configure the interfaces in the master routing instance on the PE routers. This example shows the interfaces for one PE router.

```
[edit interfaces]
user@host# set fe-0/2/0 unit 0 family inet address 192.168.195.109/30
user@host# set fe-0/2/1 unit 0 family inet address 192.168.195.5/27
user@host# set fe-0/2/2 unit 0 family inet address 20.10.1.1/30
user@host# set fe-0/2/2 unit 0 family iso
user@host# set fe-0/2/2 unit 0 family mpls
user@host# set lo0 unit 1 family inet address 10.10.47.100/32
user@host# set lo0 unit 2 family inet address 10.10.48.100/32
```

2. Configure the autonomous system number in the global routing options. This is required in MBGP MVPNs.

```
[edit routing-options]
user@host# set autonomous-system 100
```

3. Configure the routing protocols in the master routing instance on the PE routers.

```
user@host# set protocols mpls interface all

[edit protocols bgp group ibgp]
user@host# set type internal
user@host# set family inet-vpn any
user@host# set family inet-mvpn signaling
user@host# set neighbor 10.255.112.155

[edit protocols isis]
user@host# set level 1 disable
user@host# set interface all
user@host# set interface fxp0.0 disable

[edit protocols ospf]
user@host# set traffic-engineering
user@host# set area 0.0.0.0 interface all
user@host# set area 0.0.0.0 interface fxp0.0 disable

user@host# set protocols ldp interface all

[edit protocols pim]
user@host# set rp static address 10.255.112.155
user@host# set interface all mode sparse-dense
user@host# set interface all version 2
user@host# set interface fxp0.0 disable
```

4. Configure routing instance VPN-A.

```
[edit routing-instances VPN-A]
user@host# set instance-type vrf
user@host# set interface fe-0/2/1.0
user@host# set interface lo0.1
user@host# set route-distinguisher 10.255.112.199:100
user@host# set provider-tunnel family inet pim-ssm group-address 232.1.1.1
user@host# set vrf-target target:100:100
user@host# set vrf-table-label
user@host# set routing-options auto-export
user@host# set protocols ospf export bgp-to-ospf
user@host# set protocols ospf area 0.0.0.0 interface lo0.1
user@host# set protocols ospf area 0.0.0.0 interface fe-0/2/1.0
user@host# set protocols pim rp static address 10.10.47.101
user@host# set protocols pim interface lo0.1 mode sparse-dense
user@host# set protocols pim interface lo0.1 version 2
user@host# set protocols pim interface fe-0/2/1.0 mode sparse-dense
user@host# set protocols pim interface fe-0/2/1.0 version 2
user@host# set protocols mvpn family inet
```

5. Configure routing instance VPN-B.

```
[edit routing-instances VPN-B]
user@host# set instance-type vrf
user@host# set interface fe-0/2/0.0
user@host# set interface lo0.2
user@host# set route-distinguisher 10.255.112.199:200
user@host# set provider-tunnel family inet pim-ssm group-address 232.2.2.2
user@host# set vrf-target target:200:200
user@host# set vrf-table-label
user@host# set routing-options auto-export
user@host# set protocols ospf export bgp-to-ospf
user@host# set protocols ospf area 0.0.0.0 interface lo0.2
user@host# set protocols ospf area 0.0.0.0 interface fe-0/2/0.0
user@host# set protocols pim rp static address 10.10.48.101
user@host# set protocols pim interface lo0.2 mode sparse-dense
user@host# set protocols pim interface lo0.2 version 2
user@host# set protocols pim interface fe-0/2/0.0 mode sparse-dense
user@host# set protocols pim interface fe-0/2/0.0 version 2
user@host# set protocols mvpn family inet
```

6. Configure the topology such that the BGP route to the source advertised by PE1 has a higher preference than the BGP route to the source advertised by PE2.

```
[edit protocols bgp]
user@host# set group ibgp local-preference 120
```

7. Configure a higher primary loopback address on PE2 than on PE1. This ensures that PE2 is the MBGP MVPN single-forwarder election winner.

```
[edit]
user@host# set interface lo0 unit 1 family inet address 1.1.1.1/32 primary
```

8. Configure the **unicast-umh-knob** statement on PE3.

```
[edit]
user@host# set routing-instances VPN-A protocols mvpn unicast-umh-election
user@host# set routing-instances VPN-B protocols mvpn unicast-umh-election
```

9. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show interfaces**, **show protocols**, **show routing-instances**, and **show routing-options** commands from configuration mode. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
fe-0/2/0 {
    unit 0 {
        family inet {
            address 192.168.195.109/30;
        }
    }
}
fe-0/2/1 {
    unit 0 {
        family inet {
            address 192.168.195.5/27;
        }
    }
}
fe-0/2/2 {
    unit 0 {
        family inet {
            address 20.10.1.1/30;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 1 {
        family inet {
            address 10.10.47.100/32;
            address 1.1.1.1/32 {
                primary;
            }
        }
    }
    unit 2 {
        family inet {
            address 10.10.48.100/32;
        }
```

```
    }
}

user@host# show protocols
mpls {
    interface all;
}
bgp {
    group ibgp {
        type internal;
        local-preference 120;
        family inet-vpn {
            any;
        }
        family inet-mvpn {
            signaling;
        }
        neighbor 10.255.112.155;
    }
}
isis {
    level 1 disable;
    interface all;
    interface fxp0.0 {
        disable;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all;
}
pim {
    rp {
        static {
            address 10.255.112.155;
        }
    }
    interface all {
        mode sparse-dense;
        version 2;
    }
    interface fxp0.0 {
        disable;
    }
}

user@host# show routing-instances
VPN-A {
    instance-type vrf;
```

```
    interface fe-0/2/1.0;
    interface lo0.1;
    route-distinguisher 10.255.112.199:100;
    provider-tunnel {
      family inet
        pim-ssm {
        group-address 232.1.1.1;
      }
    }
    vrf-target target:100:100;
    vrf-table-label;
    routing-options {
      auto-export;
    }
    protocols {
      ospf {
        export bgp-to-ospf;
        area 0.0.0.0 {
          interface lo0.1;
          interface fe-0/2/1.0;
        }
      }
      pim {
        rp {
          static {
            address 10.10.47.101;
          }
        }
        interface lo0.1 {
          mode sparse-dense;
          version 2;
        }
        interface fe-0/2/1.0 {
          mode sparse-dense;
          version 2;
        }
      }
      mvpn {
        unicast-umh-election;
      }
    }
  }
  VPN-B {
    instance-type vrf;
    interface fe-0/2/0.0;
    interface lo0.2;
    route-distinguisher 10.255.112.199:200;
    provider-tunnel {
      family inet {
        pim-ssm {
        group-address 232.2.2.2;
      }
    }
    vrf-target target:200:200;
    vrf-table-label;
    routing-options {
```

```
    auto-export;
  }
  protocols {
    ospf {
      export bgp-to-ospf;
      area 0.0.0.0 {
        interface lo0.2;
        interface fe-0/2/0.0;
      }
    }
    pim {
      rp {
        static {
          address 10.10.48.101;
        }
      }
      interface lo0.2 {
        mode sparse-dense;
        version 2;
      }
      interface fe-0/2/0.0 {
        mode sparse-dense;
        version 2;
      }
    }
    mvpn {
      unicast-umh-election;
    }
  }
}

fe-0/2/0 {
  unit 0 {
    family inet {
      address 192.168.195.109/30;
    }
  }
}
fe-0/2/1 {
  unit 0 {
    family inet {
      address 192.168.195.5/27;
    }
  }
}

user@host# show routing-options
autonomous-system 100;
```

Verification

To verify the configuration, start the receivers and the source. PE3 should create type-7 customer multicast routes from the local joins. Verify the source-tree customer multicast entries on all PE routers. PE3 should choose PE1 as the upstream PE toward the source. PE1 receives the customer multicast route from the egress PEs and forwards data on the PSMI to PE3.

To confirm the configuration, run the following commands:

- show route table **VPN-A.mvpn.0 extensive**
- show multicast route **extensive instance VPN-A**

See Also

- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668
- *Configuring PIM Provider Tunnels for an MBGP MVPN*

Example: Allowing MBGP MVPN Remote Sources

This example shows how to configure an MBGP MVPN that allows remote sources, even when there is no PIM neighborship toward the upstream router.

Requirements

Before you begin:

- Configure the router interfaces. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure the point-to-multipoint static LSP. See *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*.

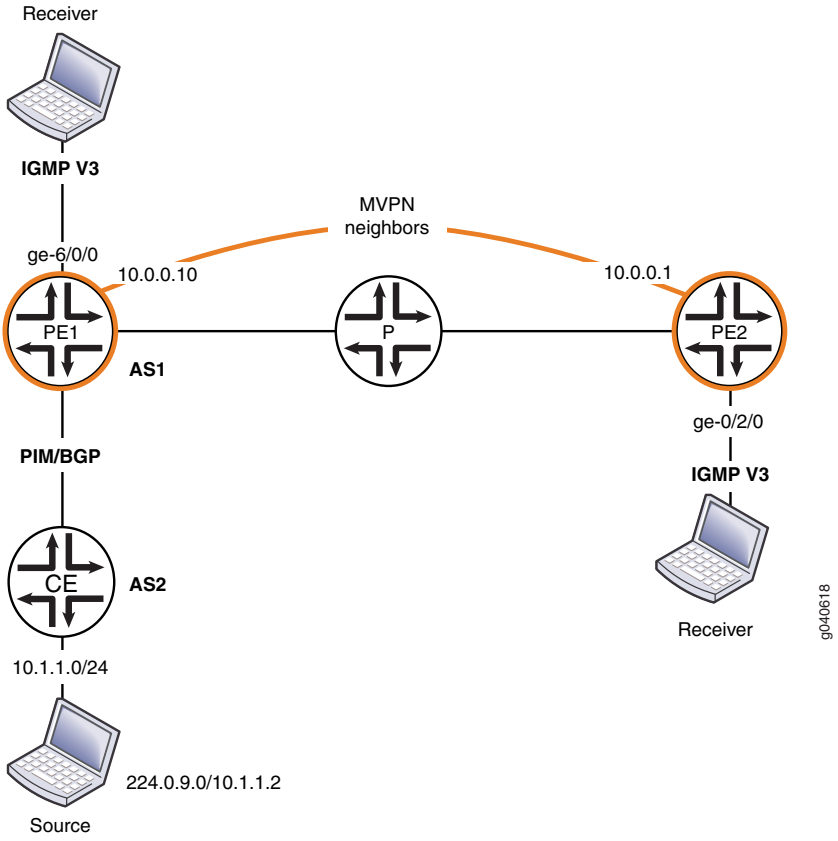
Overview

In this example, a remote CE router is the multicast source. In an MBGP MVPN, a PE router has the PIM interface hello interval set to zero, thereby creating no PIM neighborship. The PIM upstream state is None. In this scenario, directly connected receivers receive traffic in the MBGP MVPN only if you configure the ingress PE's upstream logical interface to accept remote sources. If you do not configure the ingress PE's logical interface to accept remote sources, the multicast route is deleted and the local receivers are no longer attached to the flood next hop.

This example shows the configuration on the ingress PE router. A static LSP is used to receive traffic from the remote source.

Figure 103 on page 634 shows the topology used in this example.

Figure 103: MBGP MVPN Remote Source



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-instances vpn-A instance-type vrf
set routing-instances vpn-A interface ge-1/0/0.213
set routing-instances vpn-A interface ge-1/0/0.484
set routing-instances vpn-A interface ge-1/0/1.200
set routing-instances vpn-A interface ge-1/0/2.0
set routing-instances vpn-A interface ge-1/0/7.0
set routing-instances vpn-A interface vt-1/1/0.0
set routing-instances vpn-A route-distinguisher 10.0.0.10:04
set routing-instances vpn-A provider-tunnel rsvp-te label-switched-path-template
    mvpn-dynamic
set routing-instances vpn-A provider-tunnel selective group 224.0.9.0/32 source 10.1.1.2/32
    rsvp-te static-lsp mvpn-static
```

```
set routing-instances vpn-A vrf-target target:65000:04
set routing-instances vpn-A protocols bgp group 1a type external
set routing-instances vpn-A protocols bgp group 1a peer-as 65213
set routing-instances vpn-A protocols bgp group 1a neighbor 10.2.213.9
set routing-instances vpn-A protocols pim interface all hello-interval 0
set routing-instances vpn-A protocols pim interface ge-1/0/2.0 accept-remote-source
set routing-instances vpn-A protocols mvpn
set routing-options autonomous-system 100
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To allow remote sources:

1. On the ingress PE router, configure the interfaces in the routing instance.

   ```
   [edit routing-instances vpn-A]
   user@host# set instance-type vrf
   user@host# set interface ge-1/0/0.213
   user@host# set interface ge-1/0/0.484
   user@host# set interface ge-1/0/1.200
   user@host# set interface ge-1/0/2.0
   user@host# set interface ge-1/0/7.0
   user@host# set interface vt-1/1/0.0
   ```

2. Configure the autonomous system number in the global routing options. This is required in MBGP MVPNs.

   ```
   user@host# set routing-options autonomous-system 100
   ```

3. Configure the route distinguisher and the VRF target.

   ```
   [edit routing-instances vpn-A]
   user@host# set route-distinguisher 10.0.0.10:04
   user@host# set vrf-target target:65000:04
   ```

4. Configure the provider tunnel.

   ```
   [edit routing-instances vpn-A]
   user@host# set provider-tunnel rsvp-te label-switched-path-template
     mvpn-dynamic
   user@host# set provider-tunnel selective group 224.0.9.0/32 source 10.1.1.2/32
     rsvp-te static-lsp mvpn-static
   ```

5. Configure BGP in the routing instance.

   ```
   [edit routing-instances vpn-A]
   user@host# set protocols bgp group 1a type external
   user@host# set protocols bgp group 1a peer-as 65213
   user@host# set protocols bgp group 1a neighbor 10.2.213.9
   ```

6. Configure PIM in the routing instance, including the **accept-remote-source** statement on the incoming logical interface.

```
[edit routing-instances vpn-A]
user@host# set protocols pim interface all hello-interval 0
user@host# set protocols pim interface ge-1/0/2.0 accept-remote-source
```

7. Enable the MVPN Protocol in the routing instance.

```
[edit routing-instances vpn-A]
user@host# set protocols mvpn
```

8. If you are done configuring the devices, commit the configuration.

```
user@host# commit
```

Results

From configuration mode, confirm your configuration by entering the **show routing-instances** and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show routing-instances
routing-instances {
  vpn-A {
    instance-type vrf;
    interface ge-1/0/0.213;
    interface ge-1/0/0.484;
    interface ge-1/0/1.200;
    interface vt-1/1/0.0;
    interface ge-1/0/2.0;
    interface ge-1/0/7.0;
    route-distinguisher 10.0.0.10:04;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          mvpn-dynamic;
        }
      }
      selective {
        group 224.0.9.0/32 {
          source 10.1.1.2/32 {
            rsvp-te {
              static-lsp mvpn-static;
            }
          }
        }
      }
    }
    vrf-target target:65000:04;
    protocols {
      bgp {
```

```
        group 1a {
          type external;
          peer-as 65213;
          neighbor 10.2.213.9;
        }
      }
      pim {
        interface all {
          hello-interval 0;
        }
        interface ge-1/0/2.0 {
          accept-remote-source;
        }
      }
      mvpn;
    }
  }
```

```
user@host# show routing-options
autonomous-system 100;
```

Verification

To verify the configuration, run the following commands:

- show mpls lsp **p2mp**
- show multicast route **instance vpn-A extensive**
- show mvpn c-multicast
- show pim join **instance vpn-A extensive**
- show route forwarding-table **destination** ***destination***
- show route table **vpn-A.mvpn.0**

See Also

- Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN on page 624
- Configuring the Interface to Accept Traffic from a Remote Source on page 403

Example: Configuring BGP Route Flap Damping Based on the MBGP MVPN Address Family

This example shows how to configure an multiprotocol BGP multicast VPN (also called Next-Generation MVPN) with BGP route flap damping.

Requirements

This example uses Junos OS Release 12.2. BGP route flap damping support for MBGP MVPN, specifically, and on an address family basis, in general, is introduced in Junos OS Release 12.2.

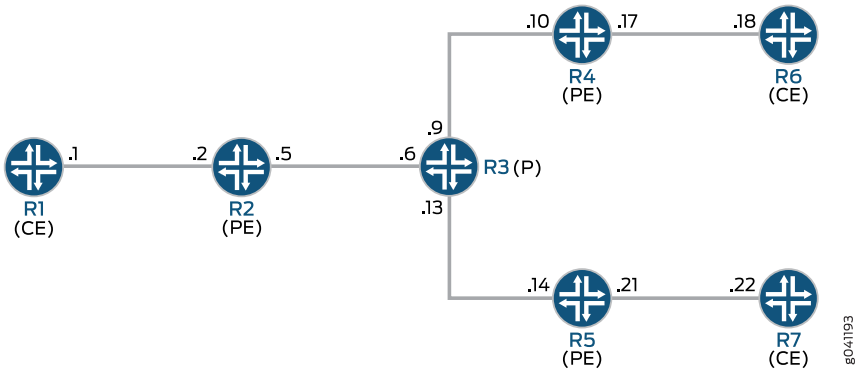
Overview

BGP route flap damping helps to diminish route instability caused by routes being repeatedly withdrawn and readvertised when a link is intermittently failing.

This example uses the default damping parameters and demonstrates an MBGP MVPN scenario with three provider edge (PE) routing devices, three customer edge (CE) routing devices, and one provider (P) routing device.

Figure 104 on page 638 shows the topology used in this example.

Figure 104: MBGP MVPN with BGP Route Flap Damping



On PE Device R4, BGP route flap damping is configured for address family **inet-mvpn**. A routing policy called **dampPolicy** uses the **nlri-route-type** match condition to damp only MVPN route types 3, 4, and 5. All other MVPN route types are not damped.

This example shows the full configuration on all devices in the "CLI Quick Configuration" on page 638 section. The "Configuring Device R4" on page 642 section shows the step-by-step configuration for PE Device R4.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces ge-1/2/0 unit 1 family inet address 10.1.1.1/30
set interfaces ge-1/2/0 unit 1 family mpls
set interfaces lo0 unit 1 family inet address 172.16.1.1/32
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.1
set protocols pim rp static address 172.16.100.1
set protocols pim interface all
```

```
set routing-options router-id 172.16.1.1
```

Device R2

```
set interfaces ge-1/2/0 unit 2 family inet address 10.1.1.2/30
set interfaces ge-1/2/0 unit 2 family mpls
set interfaces ge-1/2/1 unit 5 family inet address 10.1.1.5/30
set interfaces ge-1/2/1 unit 5 family mpls
set interfaces vt-1/2/0 unit 2 family inet
set interfaces lo0 unit 2 family inet address 172.16.1.2/32
set interfaces lo0 unit 102 family inet address 172.16.100.1/32
set protocols mpls interface ge-1/2/1.5
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 172.16.1.2
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 172.16.1.4
set protocols bgp group ibgp neighbor 172.16.1.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/1.5
set protocols ldp interface ge-1/2/1.5
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface ge-1/2/0.2
set routing-instances vpn-1 interface vt-1/2/0.2
set routing-instances vpn-1 interface lo0.102
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 provider-tunnel ldp-p2mp
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.102 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/0.2
set routing-instances vpn-1 protocols pim rp static address 172.16.1.2 with 172.16.4.1100.1
set routing-instances vpn-1 protocols pim interface ge-1/2/0.2 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 172.16.1.2
set routing-options autonomous-system 1001
```

Device R3

```
set interfaces ge-1/2/0 unit 6 family inet address 10.1.1.6/30
set interfaces ge-1/2/0 unit 6 family mpls
set interfaces ge-1/2/1 unit 9 family inet address 10.1.1.9/30
set interfaces ge-1/2/1 unit 9 family mpls
set interfaces ge-1/2/2 unit 13 family inet address 10.1.1.13/30
set interfaces ge-1/2/2 unit 13 family mpls
set interfaces lo0 unit 3 family inet address 172.16.1.3/32
set protocols mpls interface ge-1/2/0.6
set protocols mpls interface ge-1/2/1.9
set protocols mpls interface ge-1/2/2.13
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.6
set protocols ospf area 0.0.0.0 interface ge-1/2/1.9
set protocols ospf area 0.0.0.0 interface ge-1/2/2.13
set protocols ldp interface ge-1/2/0.6
set protocols ldp interface ge-1/2/1.9
```

```
set protocols ldp interface ge-1/2/2.13
set protocols ldp p2mp
set routing-options router-id 172.16.1.3
```

Device R4

```
set interfaces ge-1/2/0 unit 10 family inet address 10.1.1.10/30
set interfaces ge-1/2/0 unit 10 family mpls
set interfaces ge-1/2/1 unit 17 family inet address 10.1.1.17/30
set interfaces ge-1/2/1 unit 17 family mpls
set interfaces vt-1/2/0 unit 4 family inet
set interfaces lo0 unit 4 family inet address 172.16.1.4/32
set interfaces lo0 unit 104 family inet address 172.16.100.1/32
set protocols rsvp interface all aggregate
set protocols mpls interface all
set protocols mpls interface ge-1/2/0.10
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 172.16.1.4
set protocols bgp group ibgp family inet-vpn unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling damping
set protocols bgp group ibgp neighbor 172.16.1.2 import dampPolicy
set protocols bgp group ibgp neighbor 172.16.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10
set protocols ldp interface ge-1/2/0.10
set protocols ldp p2mp
set policy-options policy-statement dampPolicy term term1 from family inet-mvpn
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 3
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 4
set policy-options policy-statement dampPolicy term term1 from nlri-route-type 5
set policy-options policy-statement dampPolicy term term1 then accept
set policy-options policy-statement dampPolicy then damping no-damp
set policy-options policy-statement dampPolicy then accept
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set policy-options damping no-damp disable
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.4
set routing-instances vpn-1 interface ge-1/2/1.17
set routing-instances vpn-1 interface lo0.104
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.104 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.17
set routing-instances vpn-1 protocols pim rp static address 172.16.100.1
set routing-instances vpn-1 protocols pim interface ge-1/2/1.17 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 172.16.1.4
set routing-options autonomous-system 64501
```

Device R5

```
set interfaces ge-1/2/0 unit 14 family inet address 10.1.1.14/30
set interfaces ge-1/2/0 unit 14 family mpls
```

```
set interfaces ge-1/2/1 unit 21 family inet address 10.1.1.21/30
set interfaces ge-1/2/1 unit 21 family mpls
set interfaces vt-1/2/0 unit 5 family inet
set interfaces lo0 unit 5 family inet address 172.16.1.5/32
set interfaces lo0 unit 105 family inet address 172.16.100.5/32
set protocols mpls interface ge-1/2/0.14
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 172.16.1.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 172.16.1.2
set protocols bgp group ibgp neighbor 172.16.1.4
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.14
set protocols ldp interface ge-1/2/0.14
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.5
set routing-instances vpn-1 interface ge-1/2/1.21
set routing-instances vpn-1 interface lo0.105
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.105 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.21
set routing-instances vpn-1 protocols pim rp static address 172.16.100.2
set routing-instances vpn-1 protocols pim interface ge-1/2/1.21 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 172.16.1.5
set routing-options autonomous-system 1001
```

Device R6

```
set interfaces ge-1/2/0 unit 18 family inet address 10.1.1.18/30
set interfaces ge-1/2/0 unit 18 family mpls
set interfaces lo0 unit 6 family inet address 172.16.1.6/32
set protocols sap listen 233.1.1.1
set protocols ospf area 0.0.0.0 interface lo0.6 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.18
set protocols pim rp static address 172.16.100.2
set protocols pim interface all
set routing-options router-id 172.16.1.6
```

Device R7

```
set interfaces ge-1/2/0 unit 22 family inet address 10.1.1.22/30
set interfaces ge-1/2/0 unit 22 family mpls
set interfaces lo0 unit 7 family inet address 172.16.1.7/32
set protocols ospf area 0.0.0.0 interface lo0.7 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.22
set protocols pim rp static address 172.16.100.2
set protocols pim interface all
set routing-options router-id 172.16.1.7
```

Configuring Device R4

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Device R4:

1. Configure the interfaces.

```
[edit interfaces]
user@R4# set ge-1/2/0 unit 10 family inet address 10.1.1.10/30
user@R4# set ge-1/2/0 unit 10 family mpls

user@R4# set ge-1/2/1 unit 17 family inet address 10.1.1.17/30
user@R4# set ge-1/2/1 unit 17 family mpls

user@R4# set vt-1/2/0 unit 4 family inet

user@R4# set lo0 unit 4 family inet address 172.16.1.4/32
user@R4# set lo0 unit 104 family inet address 172.16.100.4/32
```

2. Configure MPLS and the signaling protocols on the interfaces.

```
[edit protocols]
user@R4# set mpls interface all
user@R4# set mpls interface ge-1/2/0.10
user@R4# set rsvp interface all aggregate
user@R4# set ldp interface ge-1/2/0.10
user@R4# set ldp p2mp
```

3. Configure BGP.

   The BGP configuration enables BGP route flap damping for the **inet-mvpn** address family. The BGP configuration also imports into the routing table the routing policy called **dampPolicy**. This policy is applied to neighbor PE Device R2.

```
[edit protocols bgp group ibgp]
user@R4# set type internal
user@R4# set local-address 172.16.1.4
user@R4# set family inet-vpn unicast
user@R4# set family inet-vpn any
user@R4# set family inet-mvpn signaling damping
user@R4# set neighbor 172.16.1.2 import dampPolicy
user@R4# set neighbor 172.16.1.5
```

4. Configure an interior gateway protocol.

```
[edit protocols ospf]
user@R4# set traffic-engineering

[edit protocols ospf area 0.0.0.0]
```

```
user@R4# set interface all
user@R4# set interface lo0.4 passive
user@R4# set interface ge-1/2/0.10
```

5. Configure a damping policy that uses the **nlri-route-type** match condition to damp only MVPN route types 3, 4, and 5.

```
[edit policy-options policy-statement dampPolicy term term1]
user@R4# set from family inet-mvpn
user@R4# set from nlri-route-type 3
user@R4# set from nlri-route-type 4
user@R4# set from nlri-route-type 5
user@R4# set then accept
```

6. Configure the **damping** policy to disable BGP route flap damping.

   The **no-damp** policy (**damping no-damp disable**) causes any damping state that is present in the routing table to be deleted. The **then damping no-damp** statement applies the **no-damp** policy as an action and has no **from** match conditions. Therefore, all routes that are not matched by **term1** are matched by this term, with the result that all other MVPN route types are not damped.

```
[edit policy-options policy-statement dampPolicy]
user@R4# set then damping no-damp
user@R4# set then accept

[edit policy-options]
user@R4# set damping no-damp disable
```

7. Configure the **parent_vpn_routes** to accept all other BGP routes that are not from the **inet-mvpn** address family.

   This policy is applied as an OSPF export policy in the routing instance.

```
[edit policy-options policy-statement parent_vpn_routes]
user@R4# set from protocol bgp
user@R4# set then accept
```

8. Configure the VPN routing and forwarding (VRF) instance.

```
[edit routing-instances vpn-1]
user@R4# set instance-type vrf
user@R4# set interface vt-1/2/0.4
user@R4# set interface ge-1/2/1.17
user@R4# set interface lo0.104
user@R4# set route-distinguisher 100:100
user@R4# set vrf-target target:1:1
user@R4# set protocols ospf export parent_vpn_routes
user@R4# set protocols ospf area 0.0.0.0 interface lo0.104 passive
user@R4# set protocols ospf area 0.0.0.0 interface ge-1/2/1.17
user@R4# set protocols pim rp static address 172.16.100.2
user@R4# set protocols pim interface ge-1/2/1.17 mode sparse
user@R4# set protocols mvpn
```

9. Configure the router ID and the autonomous system (AS) number.

```
[edit routing-options]
user@R4# set router-id 172.16.1.4
user@R4# set autonomous-system 1001
```

10. If you are done configuring the device, commit the configuration.

```
user@R4# commit
```

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R4# show interfaces
ge-1/2/0 {
    unit 10 {
        family inet {
            address 10.1.1.10/30;
        }
        family mpls;
    }
}
ge-1/2/1 {
    unit 17 {
        family inet {
            address 10.1.1.17/30;
        }
        family mpls;
    }
}
vt-1/2/0 {
    unit 4 {
        family inet;
    }
}
lo0 {
    unit 4 {
        family inet {
            address 172.16.1.4/32;
        }
    }
    unit 104 {
        family inet {
            address 172.16.100.4/32;
        }
    }
}

user@R4# show protocols
rsvp {
```

```
    interface all {
      aggregate;
    }
  }
  mpls {
    interface all;
    interface ge-1/2/0.10;
  }
  bgp {
    group ibgp {
      type internal;
      local-address 172.16.1.4;
      family inet-vpn {
        unicast;
        any;
      }
      family inet-mvpn {
        signaling {
          damping;
        }
      }
      neighbor 172.16.1.2 {
        import dampPolicy;
      }
      neighbor 172.16.1.5;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
      interface lo0.4 {
        passive;
      }
      interface ge-1/2/0.10;
    }
  }
  ldp {
    interface ge-1/2/0.10;
    p2mp;
  }

  user@R4# show policy-options
  policy-statement dampPolicy {
    term term1 {
      from {
        family inet-mvpn;
        nlri-route-type [ 3 4 5 ];
      }
      then accept;
    }
    then {
      damping no-damp;
      accept;
    }
  }
```

```
policy-statement parent_vpn_routes {
   from protocol bgp;
   then accept;
}
damping no-damp {
   disable;
}
```

```
user@R4# show routing-instances
vpn-1 {
   instance-type vrf;
   interface vt-1/2/0.4;
   interface ge-1/2/1.17;
   interface lo0.104;
   route-distinguisher 100:100;
   vrf-target target:1:1;
   protocols {
      ospf {
         export parent_vpn_routes;
         area 0.0.0.0 {
            interface lo0.104 {
               passive;
            }
            interface ge-1/2/1.17;
         }
      }
      pim {
         rp {
            static {
               address 172.16.100.2;
            }
         }
         interface ge-1/2/1.17 {
            mode sparse;
         }
      }
      mvpn;
   }
}
```

```
user@R4# show routing-optons
router-id 172.16.1.4;
autonomous-system 1001;
```

Verification

Confirm that the configuration is working properly.

- Verifying That Route Flap Damping Is Disabled on page 646
- Verifying Route Flap Damping on page 647

Verifying That Route Flap Damping Is Disabled

Purpose Verify the presence of the **no-damp** policy, which disables damping for MVPN route types other than 3, 4, and 5.

Action From operational mode, enter the **show policy damping** command.

```
user@R4> show policy damping
Default damping information:
  Halflife: 15 minutes
  Reuse merit: 750 Suppress/cutoff merit: 3000
  Maximum suppress time: 60 minutes
  Computed values:
    Merit ceiling: 12110
    Maximum decay: 6193
Damping information for "no-damp":
 Damping disabled
```

Meaning The output shows that the default damping parameters are in effect and that the **no-damp** policy is also in effect for the specified route types.

Verifying Route Flap Damping

Purpose Check whether BGP routes have been damped.

Action From operational mode, enter the **show bgp summary** command.

```
user@R4> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0
                       6          6          0          0          0          0
bgp.l3vpn.2
                       0          0          0          0          0          0
bgp.mvpn.0
                       2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
172.16.1.2               1001       3159       3155       0       0    23:43:47
 Establ
  bgp.l3vpn.0: 3/3/3/0
  bgp.l3vpn.2: 0/0/0/0
  bgp.mvpn.0: 1/1/1/0
  vpn-1.inet.0: 3/3/3/0
  vpn-1.mvpn.0: 1/1/1/0
172.16.1.5               1001       3157       3154       0       0    23:43:40
 Establ
  bgp.l3vpn.0: 3/3/3/0
  bgp.l3vpn.2: 0/0/0/0
  bgp.mvpn.0: 1/1/1/0
  vpn-1.inet.0: 3/3/3/0
  vpn-1.mvpn.0: 1/1/1/0
```

Meaning The Damp State field shows that zero routes in the bgp.mvpn.0 routing table have been damped. Further down, the last number in the State field shows that zero routes have been damped for BGP peer 172.16.1.2.

See Also

- *Understanding Damping Parameters*
- *Using Routing Policies to Damp BGP Route Flapping*
- *Example: Configuring BGP Route Flap Damping Parameters*

Example: Configuring MBGP Multicast VPN Topology Variations

This section describes how to configure multicast virtual private networks (MVPNs) using multiprotocol BGP (MBGP) (next-generation MVPNs).

Requirements

To implement multiprotocol BGP-based multicast VPNs, auto-RP, bootstrap router (BSR) RP, and PIM dense mode you need JUNOS Release 9.2 or later.

To implement multiprotocol BGP-based multicast VPNs, sender-only sites, and receiver-only sites you need JUNOS Release 8.4 or later.

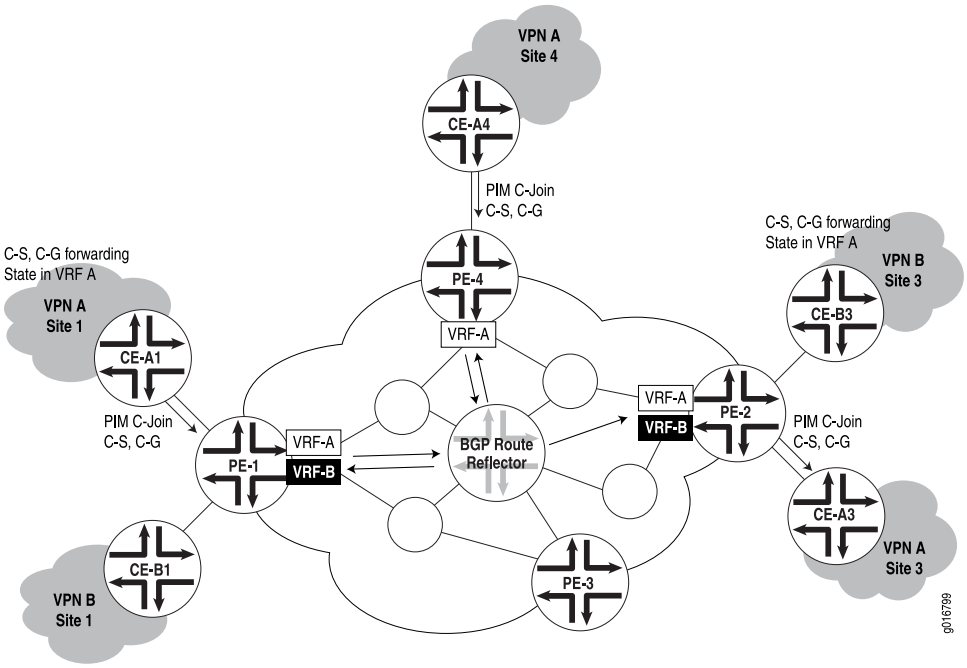
Overview and Topology

You can configure PIM auto-RP, bootstrap router (BSR) RP, PIM dense mode, and mtrace for next generation multicast VPN networks. Auto-RP uses PIM dense mode to propagate control messages and establish RP mapping. You can configure an auto-RP node in one of three different modes: discovery mode, announce mode, and mapping mode. BSR is the IETF standard for RP establishment. A selected router in a network acts as a BSR, which selects a unique RP for different group ranges. BSR messages are flooded using the data tunnel between PE routers. When you enable PIM dense mode, data packets are forwarded to all interfaces except the incoming interface. Unlike PIM sparse mode, where explicit joins are required for data packets to be transmitted downstream, data packets are flooded to all routers in the routing instance in PIM dense mode.

This section shows you how to configure a MVPN using MBGP. If you have multicast VPNs based on draft-rosen, they will continue to work as before and are not affected by the configuration of MVPNs using MBGP.

The network configuration used for most of the examples in this section is shown in Figure 105 on page 649.

Figure 105: MBGP MVPN Topology Variations Diagram



In the figure, two VPNs, VPN A and VPN B, are serviced by the same provider at several sites, two of which have CE routers for both VPN A and VPN B (site 2 is not shown). The PE routers are shown with VRF tables for the VPN CEs for which they have routing information. It is important to note that no multicast protocols are required between the PE routers on the network. The multicast routing information is carried by MBGP between the PE routers. There may be one or more BGP route reflectors in the network. Both VPNs operate independently and are configured separately.

Both the PE and CE routers run PIM sparse mode and maintain forwarding state information about customer source (C-S) and customer group (C-G) multicast components. CE routers still send a customer's PIM join messages (PIM C-Join) from CE to PE, and from PE to CE, as shown in the figure. But on the provider's backbone network, all multicast information is carried by MBGP. The only addition over and above the unicast VPN configuration normally used is the use of a special provider tunnel (**provider-tunnel**) for carrying PIM sparse mode message content between provider nodes on the network.

There are several scenarios for MVPN configuration using MBGP, depending on whether a customer site has senders (sources) of multicast traffic, has receivers of multicast traffic, or a mixture of senders and receivers. MVPNs can be:

- A full mesh (each MVPN site has both senders and receivers)
- A mixture of sender-only and receiver-only sites
- A mixture of sender-only, receiver-only, and sender-receiver sites
- A hub and spoke (two interfaces between hub PE and hub CE, and all spokes are sender-receiver sites)

Each type of MVPN differs more in the configuration VPN statements than the provider tunnel configuration. For information about configuring VPNs, see the *Junos OS VPNs Library for Routing Devices*.

Configuring Full Mesh MBGP MVPNs

This example describes how to configure a full mesh MBGP MVPN:

Configuration Steps

Step-by-Step Procedure

In this example, PE-1 connects to VPN A and VPN B at site 1, PE-4 connects to VPN A at site 4, and PE-2 connects to VPN B at site 3. To configure a full mesh MVPN for VPN A and VPN B, perform the following steps:

1. Configure PE-1 (both VPN A and VPN B at site 1):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn;
    }
    route-distinguisher 65535:0;
    vrf-target target:1:1;
  }
  VPN-B {
    instance-type vrf;
    interface ge-0/3/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn;
    }
    route-distinguisher 65535:1;
    vrf-target target:1:2;
  }
```

2. Configure PE-4 (VPN A at site 4):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-1/0/0.0;
    provider-tunnel {
```

```
        pim-asm {
          group-address 224.1.1.1;
        }
      }
      protocols {
        mvpn;
      }
      route-distinguisher 65535:4;
      vrf-target target:1:1;
    }
```

3. Configure PE-2 (VPN B at site 3):

```
[edit]
routing-instances {
  VPN-B {
    instance-type vrf;
    interface ge-1/3/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn;
    }
    route-distinguisher 65535:3;
    vrf-target target:1:2;
  }
```

Configuring Sender-Only and Receiver-Only Sites Using PIM ASM Provider Tunnels

This example describes how to configure an MBGP MVPN with a mixture of sender-only and receiver-only sites using PIM-ASM provider tunnels.

Configuration Steps

Step-by-Step Procedure

In this example, PE-1 connects to VPN A (sender-only) and VPN B (receiver-only) at site 1, PE-4 connects to VPN A (receiver-only) at site 4, and PE-2 connects to VPN A (receiver-only) and VPN B (sender-only) at site 3.

To configure an MVPN for a mixture of sender-only and receiver-only sites on VPN A and VPN B, perform the following steps:

1. Configure PE-1 (VPN A sender-only and VPN B receiver-only at site 1):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    provider-tunnel {
      pim-asm {
```

```
            group-address 224.1.1.1;
          }
        }
        protocols {
          mvpn {
            sender-site;
            route-target {
              export-target unicast;
              import-target target target:1:4;
            }
          }
        }
        route-distinguisher 65535:0;
        vrf-target target:1:1;
        routing-options {
          auto-export;
        }
      }
      VPN-B {
        instance-type vrf;
        interface ge-0/3/0.0;
        provider-tunnel {
          pim-asm {
            group-address 224.1.1.2;
          }
        }
        protocols {
          mvpn {
            receiver-site;
            route-target {
              export-target target target:1:5;
              import-target unicast;
            }
          }
        }
        route-distinguisher 65535:1;
        vrf-target target:1:2;
        routing-options {
          auto-export;
        }
      }
```

2. Configure PE-4 (VPN A receiver-only at site 4):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-1/0/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
```

```
    mvpn {
      receiver-site;
      route-target {
        export-target target target:1:4;
        import-target unicast;
      }
    }
  }
  route-distinguisher 65535:2;
  vrf-target target:1:1;
  routing-options {
    auto-export;
  }
}
```

3. Configure PE-2 (VPN A receiver-only and VPN B sender-only at site 3):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-2/0/1.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn {
        receiver-site;
        route-target {
          export-target target target:1:4;
          import-target unicast;
        }
      }
    }
    route-distinguisher 65535:3;
    vrf-target target:1:1;
    routing-options {
      auto-export;
    }
  }
  VPN-B {
    instance-type vrf;
    interface ge–1/3/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn {
        sender-site;
        route-target {
          export-target unicast
```

```
            import-target target target:1:5;
          }
        }
      }
      route-distinguisher 65535:4;
      vrf-target target:1:2;
      routing-options {
        auto-export;
      }
    }
```

Configuring Sender-Only, Receiver-Only, and Sender-Receiver MVPN Sites

This example describes how to configure an MBGP MVPN with a mixture of sender-only, receiver-only, and sender-receiver sites.

Configuration Steps

Step-by-Step Procedure

In this example, PE-1 connects to VPN A (sender-receiver) and VPN B (receiver-only) at site 1, PE-4 connects to VPN A (receiver-only) at site 4, and PE-2 connects to VPN A (sender-only) and VPN B (sender-only) at site 3. To configure an MVPN for a mixture of sender-only, receiver-only, and sender-receiver sites for VPN A and VPN B, perform the following steps:

1. Configure PE-1 (VPN A sender-receiver and VPN B receiver-only at site 1):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn {
        route-target {
          export-target unicast target target:1:4;
          import-target unicast target target:1:4 receiver;
        }
      }
    }
    route-distinguisher 65535:0;
    vrf-target target:1:1;
    routing-options {
      auto-export;
    }
  }
  VPN-B {
    instance-type vrf;
    interface ge-0/3/0.0;
```

```
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn {
        receiver-site;
        route-target {
          export-target target target:1:5;
          import-target unicast;
        }
      }
    }
    route-distinguisher 65535:1;
    vrf-target target:1:2;
    routing-options {
      auto-export;
    }
  }
```

2. Configure PE-4 (VPN A receiver-only at site 4):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-1/0/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn {
        receiver-site;
          route-target {
          export-target target target:1:4;
          import-target unicast;
        }
      }
    }
    route-distinguisher 65535:2;
    vrf-target target:1:1;
    routing-options {
      auto-export;
    }
  }
```

3. Configure PE-2 (VPN-A sender-only and VPN-B sender-only at site 3):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
```

```
    interface so-2/0/1.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    protocols {
      mvpn {
        receiver-site;
          route-target {
          export-target target target:1:4;
          import-target unicast;
        }
      }
    }
    route-distinguisher 65535:3;
    vrf-target target:1:1;
    routing-options {
      auto-export;
    }
  }
  VPN-B {
    instance-type vrf;
    interface ge-1/3/0.0;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn {
        sender-site;
        route-target {
          export-target unicast;
          import-target target target:1:5;
        }
      }
    }
    route-distinguisher 65535:4;
    vrf-target target:1:2;
    routing-options {
      auto-export;
    }
  }
```

Configuring Hub-and-Spoke MVPNs

This example describes how to configure an MBGP MVPN in a hub and spoke topology.

Configuration Steps

Step-by-Step Procedure

In this example, which only configures VPN A, PE-1 connects to VPN A (spoke site) at site 1, PE-4 connects to VPN A (hub site) at site 4, and PE-2 connects to VPN A (spoke site) at site 3. Current support is limited to the case where there are two interfaces between the hub site CE and PE. To configure a hub-and-spoke MVPN for VPN A, perform the following steps:

1. Configure PE-1 for VPN A (spoke site):

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    protocols {
      mvpn {
        route-target {
          export-target unicast;
          import-target unicast target target:1:4;
        }
      }
    }
    route-distinguisher 65535:0;
    vrf-target {
      import target:1:1;
      export target:1:3;
    }
    routing-options {
      auto-export;
    }
  }
```

2. Configure PE-4 for VPN A (hub site):

```
[edit]
routing-instances {
  VPN-A-spoke-to-hub {
    instance-type vrf;
    interface so-1/0/0.0; #receives data and joins from the CE
    protocols {
      mvpn {
        receiver-site;
        route-target {
          export-target target target:1:4;
          import-target unicast;
        }
```

```
    }
    ospf {
      export redistribute-vpn; #redistributes VPN routes to CE
      area 0.0.0.0 {
        interface so-1/0/0;
      }
    }
  }
  route-distinguisher 65535:2;
  vrf-target {
    import target:1:3;
  }
  routing-options {
    auto-export;
  }
}
VPN-A-hub-to-spoke {
  instance-type vrf;
  interface so-2/0/0.0;  #receives data and joins from the CE
  provider-tunnel {
    rsvp-te {
      label-switched-path-template {
        default-template;
      }
    }
  }
  protocols {
    mvpn {
      sender-site;
      route-target {
        import-target target target:1:3;
        export-target unicast;
      }
    }
    ospf {
      export redistribute-vpn;  #redistributes VPN routes to CE
      area 0.0.0.0 {
        interface so-2/0/0;
      }
    }
  }
  route-distinguisher 65535:2;
  vrf-target {
    import target:1:1;
  }
  routing-options {
    auto-export;
  }
}
```

3. Configure PE-2 for VPN A (spoke site):

```
[edit]
routing-instances {
  VPN-A {
```

```
    instance-type vrf;
    interface so-2/0/1.0;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    protocols {
      mvpn {
        route–target {
          import-target target target:1:4;
          export-target unicast;
        }
      }
    }
    route-distinguisher 65535:3;
    vrf-target {
      import target:1:1;
      export target:1:3;
    }
    routing-options {
      auto-export;
    }
  }
```

Configuring Nonstop Active Routing for BGP Multicast VPN

BGP multicast virtual private network (MVPN) is a Layer 3 VPN application that is built on top of various unicast and multicast routing protocols such as Protocol Independent Multicast (PIM), BGP, RSVP, and LDP. Enabling nonstop active routing (NSR) for BGP MVPN requires that NSR support is enabled for all these protocols.

The state maintained by MVPN includes MVPN routes, cmcast, provider-tunnel, and forwarding information. BGP MVPN NSR synchronizes this MVPN state between the master and backup Routing Engines. While some of the state on the backup Routing Engine is locally built based on the configuration, most of it is built based on triggers from other protocols that MVPN interacts with. The triggers from these protocols are in turn the result of state replication performed by these modules. This includes route change notifications by unicast protocols, join and prune triggers from PIM, remote MVPN route notification by BGP, and provider-tunnel related notifications from RSVP and LDP.

Configuring NSR and unified in-service software upgrade (ISSU) support to the BGP MVPN protocol provides features such as various provider tunnel types, different MVPN modes (source tree, shared-tree), and PIM features. As a result, at the ingress PE, replication is turned on for dynamic LSPs. Thus, when NSR is configured, the state for dynamic LSPs is also replicated to the backup Routing Engine. After the state is resolved on the backup Routing Engine, RSVP sends required notifications to MVPN.

To enable BGP MVPN NSR support, the advertise-from-main-vpn-tables configuration statement needs to be configured at the **[edit protocols bgp]** hierarchy level.

Nonstop active routing configurations include two Routing Engines that share information so that routing is not interrupted during Routing Engine failover. When NSR is configured on a dual Routing Engine platform, the PIM control state is replicated on both Routing Engines.

This PIM state information includes:

- Neighbor relationships
- Join and prune information
- RP-set information
- Synchronization between routes and next hops and the forwarding state between the two Routing Engines

Junos OS supports NSR in the following PIM scenarios:

- Dense mode
- Sparse mode
- SSM
- Static RP
- Auto-RP (for IPv4 only)
- Bootstrap router
- Embedded RP on the non-RP router (for IPv6 only)
- BFD support
- Draft Rosen multicast VPNs and BGP multicast VPNs
- Policy features such as neighbor policy, bootstrap router export and import policies, scope policy, flow maps, and reverse path forwarding (RPF) check policies

Before you begin:

- Configure the router interfaces. See the *Interfaces Fundamentals for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Configure a multicast group membership protocol (IGMP or MLD). See "Understanding IGMP" on page 25 and "Understanding MLD" on page 51.
- For this feature to work with IPv6, the routing device must be running Junos OS Release 10.4 or later.

To configure nonstop active routing:

1. Because NSR requires you to configure graceful Routing Engine switchover (GRES), to enable GRES, include the **graceful-switchover** statement at the **[edit chassis redundancy]** hierarchy level.

   [edit]


```
user@host# set chassis redundancy graceful-switchover
```

2. Include the **synchronize** statement at the **[edit system]** hierarchy level so that configuration changes are synchronized on both Routing Engines.

```
[edit system]
user@host# set synchronize
user@host# exit
```

3. Configure PIM settings on the DR with sparse mode and version, and static address pointing to the rendezvous points.

```
[edit protocols pim]
user@host# set rp static address address
user@host# set interface interface-name mode sparse
user@host# set interface interface-name version 2
```

For example, to set sparse mode, version 2 and static address:

```
[edit protocols pim]
user@host# set rp static address 10.210.255.202
user@host# set interface fe-0/1/3.0 mode sparse
user@host# set interface fe-0/1/3.0 version 2
```

4. Configure per-packet load balancing on the DR.

```
[edit policy-options policy-statement policy-name]
user@host# set then policy-name per-packet
```

For example, to set load-balance policy:

```
[edit policy-options policy-statement load-balance]
user@host# set then load-balance per-packet
```

5. Apply the load-balance policy on the DR.

```
[edit]
user@host# set routing-options forwarding-table export load-balance
```

6. Configure nonstop active routing on the DR.

```
[edit]
user@host# set routing-options nonstop-routing
user@host# set routing-options router-id address
```

For example, to set nonstop active routing on the designated router with address 10.210.255.201:

```
[edit]
user@host# set routing-options router-id 10.210.255.201
```

See Also

- *Configuring Basic PIM Settings*
- Understanding Nonstop Active Routing for PIM on page 371

Release History Table

Release	Description
15.1X49-D50	Starting in Junos OS Release 15.1X49-D50 and Junos OS Release 17.3R1, the **vrf-table-label** statement allows mapping of the inner label to a specific Virtual Routing and Forwarding (VRF). This mapping allows examination of the encapsulated IP header at an egress VPN router. For SRX Series devices, the **vrf-table-label** statement is currently supported only on physical interfaces. As a workaround, deactivate **vrf-table-label** or use physical interfaces.

Related Documentation

- Example: Configuring MBGP MVPN Extranets on page 669
- Multiprotocol BGP MVPNs Overview on page 526

Configuring MBGP MVPN Wildcards

- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 662
- Configuring a Selective Provider Tunnel Using Wildcards on page 667
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668

Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN

Selective LSPs are also referred to as selective provider tunnels. Selective provider tunnels carry traffic from some multicast groups in a VPN and extend only to the PE routers that have receivers for these groups. You can configure a selective provider tunnel for group prefixes and source prefixes, or you can use wildcards for the group and source, as described in the Internet draft draft-rekhter-mvpn-wildcard-spmsi-01.txt, *Use of Wildcard in S-PMSI Auto-Discovery Routes*.

The following sections describe the scenarios and special considerations when you use wildcards for selective provider tunnels.

- About S-PMSI on page 662
- Scenarios for Using Wildcard S-PMSI on page 663
- Types of Wildcard S-PMSI on page 664
- Differences Between Wildcard S-PMSI and (S,G) S-PMSI on page 664
- Wildcard (*,*) S-PMSI and PIM Dense Mode on page 665
- Wildcard (*,*) S-PMSI and PIM-BSR on page 665
- Wildcard Source and the 0.0.0.0/0 Source Prefix on page 666

About S-PMSI

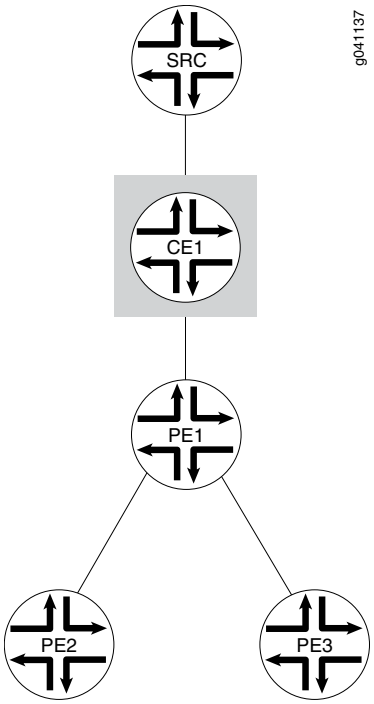
The provider multicast service interface (PMSI) is a BGP tunnel attribute that contains the tunnel ID used by the PE router for transmitting traffic through the core of the provider network. A selective PMSI (S-PMSI) autodiscovery route advertises binding of a given MVPN customer multicast flow to a particular provider tunnel. The S-PMSI autodiscovery route advertised by the ingress PE router contains /32 IPv4 or /128 IPv6 addresses for

the customer source and the customer group derived from the source-tree customer multicast route.

Figure 106 on page 663 shows a simple MVPN topology. The ingress router, PE1, originates the S-PMSI autodiscovery route. The egress routers, PE2 and PE3, have join state as a result of receiving join messages from CE devices that are not shown in the topology. In response to the S-PMSI autodiscovery route advertisement sent by PE1, PE2, and PE3, elect whether or not to join the tunnel based on the join state. The selective provider tunnel configuration is configured in a VRF instance on PE1.

NOTE: The MVPN mode configuration (RPT-SPT or SPT-only) is configured on all three PE routers for all VRFs that make up the VPN. If you omit the MVPN mode configuration, the default mode is SPT-only.

Figure 106: Simple MVPN Topology



Scenarios for Using Wildcard S-PMSI

A wildcard S-PMSI has the source or the group (or both the source and the group) field set to the wildcard value of 0.0.0.0/0 and advertises binding of multiple customer multicast flows to a single provider tunnel in a single S-PMSI autodiscovery route.

The scenarios under which you might configure a wildcard S-PMSI are as follows:

- When the customer multicast flows are PIM-SM in ASM-mode flows. In this case, a PE router connected to an MVPN customer's site that contains the customer's RP (C-RP) could bind all the customer multicast flows traveling along a customer's RPT tree to a single provider tunnel.
- When a PE router is connected to an MVPN customer's site that contains multiple sources, all sending to the same group.
- When the customer multicast flows are PIM-bidirectional flows. In this case, a PE router could bind to a single provider tunnel all the customer multicast flows for the same group that have been originated within the sites of a given MVPN connected to that PE, and advertise such binding in a single S-PMSI autodiscovery route.
- When the customer multicast flows are PIM-SM in SSM-mode flows. In this case, a PE router could bind to a single provider tunnel all the customer multicast flows coming from a given source located in a site connected to that PE router.
- When you want to carry in the provider tunnel all the customer multicast flows originated within the sites of a given MVPN connected to a given PE router.

Types of Wildcard S-PMSI

The following types of wildcard S-PMSI are supported:

- A (*,G) S-PMSI matches all customer multicast routes that have the group address. The customer source address in the customer multicast route can be any address, including 0.0.0.0/0 for shared-tree customer multicast routes. A (*, C-G) S-PMSI autodiscovery route is advertised with the source field set to 0 and the source address length set to 0. The multicast group address for the S-PMSI autodiscovery route is derived from the customer multicast joins.
- A (*,*) S-PMSI matches all customer multicast routes. Any customer source address and any customer group address in a customer multicast route can be bound to the (*,*) S-PMSI. The S-PMSI autodiscovery route is advertised with the source address and length set to 0 and the group address and length set 0. The remaining fields in the S-PMSI autodiscovery route follow the same rule as (C-S, C-G) S-PMSI, as described in section 12.1 of the BGP-MVPN draft (draft-ietf-l3vpn-2547bis-mcast-bgp-00.txt).

Differences Between Wildcard S-PMSI and (S,G) S-PMSI

For dynamic provider tunnels, each customer multicast stream is bound to a separate provider tunnel, and each tunnel is advertised by a separate S-PMSI autodiscovery route. For static LSPs, multiple customer multicast flows are bound to a single provider tunnel by having multiple S-PMSI autodiscovery routes advertise the same provider tunnel.

When you configure a wildcard (*,G) or (*,*) S-PMSI, one or more matching customer multicast routes share a single S-PMSI. All customer multicast routes that have a matching source and group address are bound to the same (*,G) or (*,*) S-PMSI and share the same tunnel. The (*,G) or (*,*) S-PMSI is established when the first matching remote customer multicast join message is received in the ingress PE router, and deleted when the last remote customer multicast join is withdrawn from the ingress PE router. Sharing a single S-PMSI autodiscovery route improves control plane scalability.

Wildcard (*,*) S-PMSI and PIM Dense Mode

For (S,G) and (*,G) S-PMSI autodiscovery routes in PIM dense mode (PIM-DM), all downstream PE routers receive PIM-DM traffic. If a downstream PE router does not have receivers that are interested in the group address, the PE router instantiates prune state and stops receiving traffic from the tunnel.

Now consider what happens for (*,*) S-PMSI autodiscovery routes. If the PIM-DM traffic is not bound by a longer matching (S,G) or (*,G) S-PMSI, it is bound to the (*,*) S-PMSI. As is always true for dense mode, PIM-DM traffic is flooded to downstream PE routers over the provider tunnel regardless of the customer multicast join state. Because there is no group information in the (*,*) S-PMSI autodiscovery route, egress PE routers join a (*,*) S-PMSI tunnel if there is any configuration on the egress PE router indicating interest in PIM-DM traffic.

Interest in PIM-DM traffic is indicated if the egress PE router has one of the following configurations in the VRF instance that corresponds to the instance that imports the S-PMSI autodiscovery route:

- At least one interface is configured in dense mode at the **[edit routing-instances *instance-name* protocols pim interface]** hierarchy level.
- At least one group is configured as a dense-mode group at the **[edit routing-instances *instance-name* protocols pim dense-groups *group-address*]** hierarchy level.

Wildcard (*,*) S-PMSI and PIM-BSR

For (S,G) and (*,G) S-PMSI autodiscovery routes in PIM bootstrap router (PIM-BSR) mode, an ingress PE router floods the PIM bootstrap message (BSM) packets over the provider tunnel to all egress PE routers. An egress PE router does not join the tunnel unless the message has the ALL-PIM-ROUTERS group. If the message has this group, the egress PE router joins the tunnel, regardless of the join state. The group field in the message determines the presence or absence of the ALL-PIM-ROUTERS address.

Now consider what would happen for (*,*) S-PMSI autodiscovery routes used with PIM-BSR mode. If the PIM BSM packets are not bound by a longer matching (S,G) or (*,G) S-PMSI, they are bound to the (*,*) S-PMSI. As is always true for PIM-BSR, BSM packets are flooded to downstream PE routers over the provider tunnel to the ALL-PIM-ROUTERS destination group. Because there is no group information in the (*,*) S-PMSI autodiscovery route, egress PE routers always join a (*,*) S-PMSI tunnel. Unlike PIM-DM, the egress PE routers might have no configuration suggesting use of PIM-BSR as the RP discovery mechanism in the VRF instance. To prevent all egress PE routers from always joining the (*,*) S-PMSI tunnel, the (*,*) wildcard group configuration must be ignored.

This means that if you configure PIM-BSR, a wildcard-group S-PMSI can be configured for all other group addresses. The (*,*) S-PMSI is not used for PIM-BSR traffic. Either a matching (*,G) or (S,G) S-PMSI (where the group address is the ALL-PIM-ROUTERS group) or an inclusive provider tunnel is needed to transmit data over the provider core. For PIM-BSR, the longest-match lookup is (S,G), (*,G), and the inclusive provider tunnel, in that order. If you do not configure an inclusive tunnel for the routing instance, you must

configure a (*,G) or (S,G) selective tunnel. Otherwise, the data is dropped. This is because PIM-BSR functions like PIM-DM, in that traffic is flooded to downstream PE routers over the provider tunnel regardless of the customer multicast join state. However, unlike PIM-DM, the egress PE routers might have no configuration to indicate interest or noninterest in PIM-BSR traffic.

Wildcard Source and the 0.0.0.0/0 Source Prefix

You can configure a 0.0.0.0/0 source prefix and a wildcard source under the same group prefix in a selective provider tunnel. For example, the configuration might look as follows:

```
routing-instances {
  vpna {
    provider-tunnel {
      selective {
        group 203.0.113.0/24 {
          source 0.0.0.0/0 {
            rsvp-te {
              label-switched-path-template {
                sptnl3;
              }
            }
          }
          wildcard-source {
            rsvp-te {
              label-switched-path-template {
                sptnl2;
              }
              static-lsp point-to-multipoint-lsp-name;
            }
            threshold-rate kbps;
          }
        }
      }
    }
  }
}
```

The functions of the **source 0.0.0.0/0** and **wildcard-source** configuration statements are different. The 0.0.0.0/0 source prefix only matches (C-S, C-G) customer multicast join messages and triggers (C-S, C-G) S-PMSI autodiscovery routes derived from the customer multicast address. Because all (C-S, C-G) join messages are matched by the 0.0.0.0/0 source prefix in the matching group, the wildcard source S-PMSI is used only for (*,C-G) customer multicast join messages. In the absence of a configured 0.0.0.0/0 source prefix, the wildcard source matches (C-S, C-G) and (*,C-G) customer multicast join messages. In the example, a join message for (10.0.1.0/24, 203.0.113.0/24) is bound to **sptnl3**. A join message for (*, 203.0.113.0/24) is bound to **sptnl2**.

See Also

- Configuring a Selective Provider Tunnel Using Wildcards on page 667
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668
- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*

- *Configuring Shared-Tree Data Distribution Across Provider Cores for Providers of MBGP MVPNs*

Configuring a Selective Provider Tunnel Using Wildcards

When you configure a selective provider tunnel for MBGP MVPNs (also referred to as next-generation Layer 3 multicast VPNs), you can use wildcards for the multicast group and source address prefixes. Using wildcards enables a PE router to use a single route to advertise the binding of multiple multicast streams of a given MVPN customer to a single provider's tunnel, as described in http://tools.ietf.org/html/draft-rekhter-mvpn-wildcard-spmsi-00 .

Sharing a single route improves control plane scalability because it reduces the number of S-PMSI autodiscovery routes.

To configure a selective provider tunnel using wildcards:

1. Configure a wildcard group matching any group IPv4 address and a wildcard source for (*,*) join messages.

   ```
   [edit routing-instances vpna provider-tunnel selective]
   user@router# set wildcard-group-inet wildcard-source
   ```

2. Configure a wildcard group matching any group IPv6 address and a wildcard source for (*,*) join messages.

   ```
   [edit routing-instances vpna provider-tunnel selective]
   user@router# set wildcard-group-inet6 wildcard-source
   ```

3. Configure an IP prefix of a multicast group and a wildcard source for (*,G) join messages.

   ```
   [edit routing-instances vpna provider-tunnel selective]
   user@router# set group 203.0.113/24 wildcard-source
   ```

4. Map the IPv4 join messages to a selective provider tunnel.

   ```
   [edit routing-instances vpna provider-tunnel selective wildcard-group-inet
     wildcard-source]
   user@router# set rsvp-te (Routing Instances Provider Tunnel Selective)
     label-switched-path-template provider-tunnel1
   ```

5. Map the IPv6 join messages to a selective provider tunnel.

   ```
   [edit routing-instances vpna provider-tunnel selective wildcard-group-inet6
     wildcard-source]
   user@router# set rsvp-te (Routing Instances Provider Tunnel Selective)
     label-switched-path-template provider-tunnel2
   ```

6. Map the (*,203.0.113/24) join messages to a selective provider tunnel.

```
[edit routing-instances vpna provider-tunnel selective group 203.0.113/24
  wildcard-source]
user@router# set rsvp-te (Routing Instances Provider Tunnel Selective)
  label-switched-path-template provider-tunnel3
```

See Also

- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 662
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668
- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*
- *Configuring Shared-Tree Data Distribution Across Provider Cores for Providers of MBGP MVPNs*

Example: Configuring Selective Provider Tunnels Using Wildcards

With the (*,G) and (*,*) S-PMSI, a customer multicast join message can match more than one S-PMSI. In this case, a customer multicast join message is bound to the longest matching S-PMSI. The longest match is a (S,G) S-PMSI, followed by a (*,G) S-PMSI and a (*,*) S-PMSI, in that order.

Consider the following configuration:

```
routing-instances {
  vpna {
    provider-tunnel {
      selective {
        wildcard-group-inet {
          wildcard-source {
            rsvp-te {
              label-switched-path-template {
                sptnl1;
              }
            }
          }
        }
        group 203.0.113.0/24 {
          wildcard-source {
            rsvp-te {
              label-switched-path-template {
                sptnl2;
              }
            }
          }
          source 10.1.1/24 {
            rsvp-te {
              label-switched-path-template {
                sptnl3;
              }
            }
          }
        }
      }
```

```
        }
    }
}
```

For this configuration, the longest-match rule works as follows:

- A customer multicast (10.1.1.1, 203.0.113.1) join message is bound to the sptnl3 S-PMSI autodiscovery route.
- A customer multicast (10.2.1.1, 203.0.113.1) join message is bound to the sptnl2 S-PMSI autodiscovery route.
- A customer multicast (10.1.1.1, 203.1.113.1) join message is bound to the sptnl1 S-PMSI autodiscovery route.

When more than one customer multicast route is bound to the same wildcard S-PMSI, only one S-PMSI autodiscovery route is created. An egress PE router always uses the same matching rules as the ingress PE router that advertises the S-PMSI autodiscovery route. This ensures consistent customer multicast mapping on the ingress and the egress PE routers.

See Also

- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 662
- Configuring a Selective Provider Tunnel Using Wildcards on page 667

Related Documentation

- Example: Configuring MBGP MVPN Extranets on page 669
- Configuring Multiprotocol BGP Multicast VPNs on page 584
- Multiprotocol BGP MVPNs Overview on page 526

Example: Configuring MBGP MVPN Extranets

- Understanding MBGP Multicast VPN Extranets on page 669
- MBGP Multicast VPN Extranets Configuration Guidelines on page 671
- Example: Configuring MBGP Multicast VPN Extranets on page 671

Understanding MBGP Multicast VPN Extranets

A multicast VPN (MVPN) extranet enables service providers to forward IP multicast traffic originating in one VPN routing and forwarding (VRF) instance to receivers in a different VRF instance. This capability is also know as *overlapping* MVPNs.

The MVPN extranet feature supports the following traffic flows:

- A receiver in one VRF can receive multicast traffic from a source connected to a different router in a different VRF.
- A receiver in one VRF can receive multicast traffic from a source connected to the same router in a different VRF.

- A receiver in one VRF can receive multicast traffic from a source connected to a different router in the same VRF.
- A receiver in one VRF can be prevented from receiving multicast traffic from a specific source in a different VRF.

MBGP Multicast VPN Extranets Application

An MVPN extranet is useful in the following applications.

Mergers and Data Sharing

An MVPN extranet is useful when there are business partnerships between different enterprise VPN customers that require them to be able to communicate with one another. For example, a wholesale company might want to broadcast inventory to its contractors and resellers. An MVPN extranet is also useful when companies merge and one set of VPN sites needs to receive content from another VPN. The enterprises involved in the merger are different VPN customers from the service provider point of view. The MVPN extranet makes the connectivity possible.

Video Distribution

Another use for MVPN extranets is video multicast distribution from a video headend to receiving sites. Sites within a given multicast VPN might be in different organizations. The receivers can subscribe to content from a specific content provider.

The PE routers on the MVPN provider network learn about the sources and receivers using MVPN mechanisms. These PE routers can use selective trees as the multicast distribution mechanism in the backbone. The network carries traffic belonging only to a specified set of one or more multicast groups, from one or more multicast VPNs. As a result, this model facilitates the distribution of content from multiple providers on a selective basis if desired.

Financial Services

A third use for MVPN extranets is enterprise and financial services infrastructures. The delivery of financial data, such as financial market updates, stock ticker values, and financial TV channels, is an example of an application that must deliver the same data stream to hundreds and potentially thousands of end users. The content distribution mechanisms largely rely on multicast within the financial provider network. In this case, there could also be an extensive multicast topology within brokerage firms and banks networks to enable further distribution of content and for trading applications. Financial service providers require traffic separation between customers accessing the content, and MVPN extranets provide this separation.

See Also

MBGP Multicast VPN Extranets Configuration Guidelines

When configuring MVPN extranets, keep the following in mind:

- If there is more than one VRF routing instance on a provider edge (PE) router that has receivers interested in receiving multicast traffic from the same source, virtual tunnel (VT) interfaces must be configured on all instances.
- For auto-RP operation, the mapping agent must be configured on at least two PEs in the extranet network.
- For asymmetrically configured extranets using auto-RP, when one VRF instance is the only instance that imports routes from all other extranet instances, the mapping agent must be configured in the VRF that can receive all RP discovery messages from all VRF instances, and mapping-agent election should be disabled.
- For bootstrap router (BSR) operation, the candidate and elected BSRs can be on PE, CE, or C routers. The PE router that connects the BSR to the MVPN extranets must have configured provider tunnels or other physical interfaces configured in the routing instance. The only case not supported is when the BSR is on a CE or C router connected to a PE routing instance that is part of an extranet but does not have configured provider tunnels and does not have any other interfaces besides the one connecting to the CE router.
- RSVP-TE point-to-multipoint LSPs must be used for the provider tunnels.
- PIM dense mode is not supported in the MVPN extranets VRF instances.

See Also

Example: Configuring MBGP Multicast VPN Extranets

This example provides a step-by-step procedure to configure multicast VPN extranets using static rendezvous points. It is organized in the following sections:

Requirements

This example uses the following hardware and software components:

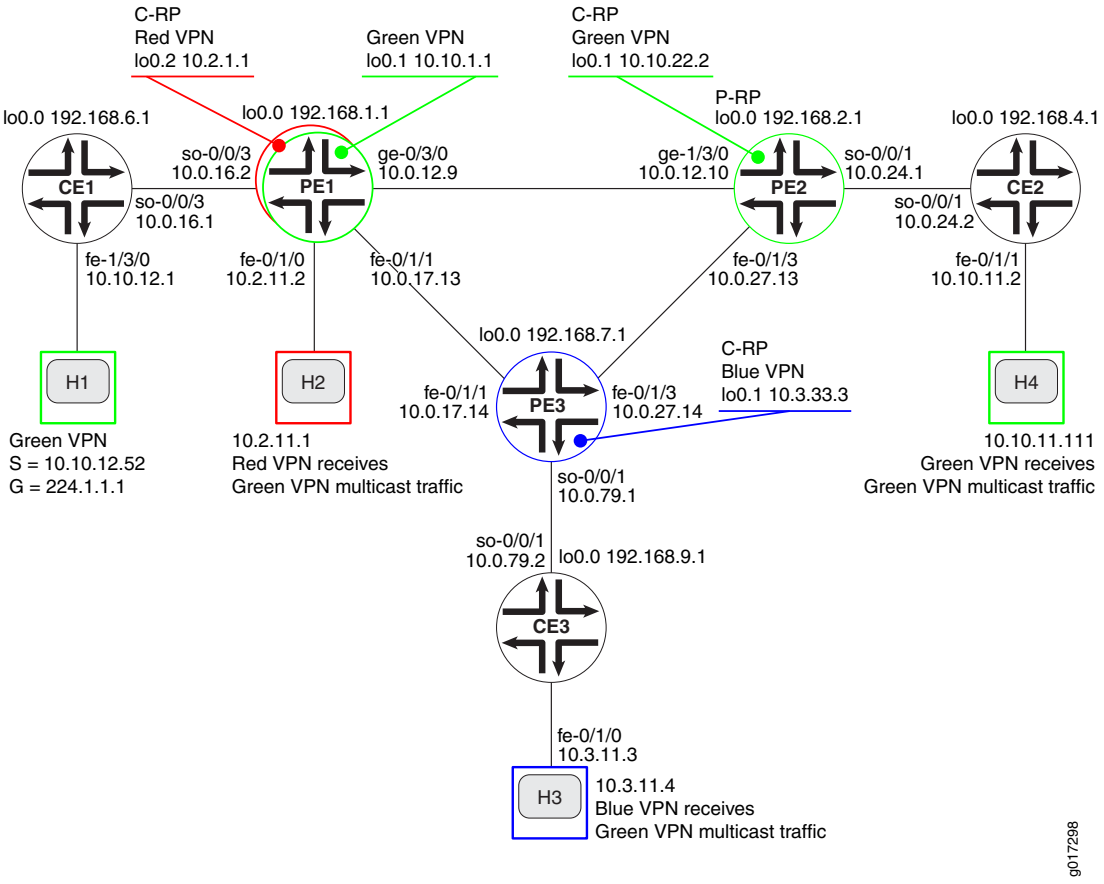
- Junos OS Release 9.5 or later
- Six M Series, T Series, TX Series, or MX Series Juniper routers
- One adaptive services PIC or MultiServices PIC in each of the M Series or T Series routers acting as PE routers
- One host system capable of sending multicast traffic and supporting the Internet Group Management Protocol (IGMP)
- Three host systems capable of receiving multicast traffic and supporting IGMP

Overview and Topology

In the network topology shown in Figure 107 on page 672:

- Host H1 is the source for group 244.1.1.1 in the green VPN.
- The multicast traffic originating at source H1 can be received by host H4 connected to router CE2 in the green VPN.
- The multicast traffic originating at source H1 can be received by host H3 connected to router CE3 in the blue VPN.
- The multicast traffic originating at source H1 can be received by host H2 directly connected to router PE1 in the red VPN.
- Any host can be a sender site or receiver site.

Figure 107: MVPN Extranets Topology Diagram



Configuration

NOTE: In any configuration session, it is good practice to verify periodically that the configuration can be committed using the commit check command.

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE2** identifies the customer edge 2 (CE2) router
- **PE2** identifies the provider edge 2 (PE2) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router

Configuring multicast VPN extranets, involves the following tasks:

Configuring Interfaces

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

1. On each router, configure an IP address on the loopback logical interface 0 (**lo0.0**).

   user@CE1# set interfaces lo0 unit 0 family inet address 192.168.6.1/32 primary

   user@PE1# set interfaces lo0 unit 0 family inet address 192.168.1.1/32 primary

   user@PE2# set interfaces lo0 unit 0 family inet address 192.168.2.1/32 primary

   user@CE2# set interfaces lo0 unit 0 family inet address 192.168.4.1/32 primary

   user@PE3# set interfaces lo0 unit 0 family inet address 192.168.7.1/32 primary

```
user@CE3# set interfaces lo0 unit 0 family inet address 192.168.9.1/32 primary
```

Use the **show interfaces terse** command to verify that the correct IP address is configured on the loopback interface.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet and Gigabit Ethernet interfaces. Specify the **inet** address family type.

```
user@CE1# set interfaces fe-1/3/0 unit 0 family inet address 10.10.12.1/24

user@PE1# set interfaces fe-0/1/0 unit 0 description "to H2"
user@PE1# set interfaces fe-0/1/0 unit 0 family inet address 10.2.11.2/30
user@PE1# set interfaces fe-0/1/1 unit 0 description "to PE3 fe-0/1/1.0"
user@PE1# set interfaces fe-0/1/1 unit 0 family inet address 10.0.17.13/30
user@PE1# set interfaces ge-0/3/0 unit 0 family inet address 10.0.12.9/30

user@PE2# set interfaces fe-0/1/3 unit 0 description "to PE3 fe-0/1/3.0"
user@PE2# set interfaces fe-0/1/3 unit 0 family inet address 10.0.27.13/30
user@PE2# set interfaces ge-1/3/0 unit 0 description "to PE1 ge-0/3/0.0"
user@PE2# set interfaces ge-1/3/0 unit 0 family inet address 10.0.12.10/30

user@CE2# set interfaces fe-0/1/1 unit 0 description "to H4"
user@CE2# set interfaces fe-0/1/1 unit 0 family inet address 10.10.11.2/24

user@PE3# set interfaces fe-0/1/1 unit 0 description "to PE1 fe-0/1/1.0"
user@PE3# set interfaces fe-0/1/1 unit 0 family inet address 10.0.17.14/30
user@PE3# set interfaces fe-0/1/3 unit 0 description "to PE2 fe-0/1/3.0"
user@PE3# set interfaces fe-0/1/3 unit 0 family inet address 10.0.27.14/30

user@CE3# set interfaces fe-0/1/0 unit 0 description "to H3"
user@CE3# set interfaces fe-0/1/0 unit 0 family inet address 10.3.11.3/24
```

Use the **show interfaces terse** command to verify that the correct IP address and address family type are configured on the interfaces.

3. On the PE and CE routers, configure the SONET interfaces. Specify the **inet** address family type, and local IP address.

```
user@CE1# set interfaces so-0/0/3 unit 0 description "to PE1 so-0/0/3.0;"
user@CE1# set interfaces so-0/0/3 unit 0 family inet address 10.0.16.1/30

user@PE1# set interfaces so-0/0/3 unit 0 description "to CE1 so-0/0/3.0"
user@PE1# set interfaces so-0/0/3 unit 0 family inet address 10.0.16.2/30

user@PE2# set interfaces so-0/0/1 unit 0 description "to CE2 so-0/0/1:0.0"
user@PE2# set interfaces so-0/0/1 unit 0 family inet address 10.0.24.1/30

user@CE2# set interfaces so-0/0/1 unit 0 description "to PE2 so-0/0/1"
user@CE2# set interfaces so-0/0/1 unit 0 family inet address 10.0.24.2/30

user@PE3# set interfaces so-0/0/1 unit 0 description "to CE3 so-0/0/1.0"
user@PE3# set interfaces so-0/0/1 unit 0 family inet address 10.0.79.1/30
```

```
user@CE3# set interfaces so-0/0/1 unit 0 description "to PE3 so-0/0/1"
user@CE3# set interfaces so-0/0/1 unit 0 family inet address 10.0.79.2/30
```

Use the **show configuration interfaces** command to verify that the correct IP address and address family type are configured on the interfaces.

4. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

5. Use the **ping** command to verify unicast connectivity between each:
   - CE router and the attached host
   - CE router and the directly attached interface on the PE router
   - PE router and the directly attached interfaces on the other PE routers

Configuring an IGP in the Core

Step-by-Step Procedure

On the PE routers, configure an interior gateway protocol such as OSPF or IS-IS. This example shows how to configure OSPF.

1. Specify the **lo0.0** and SONET core-facing logical interfaces.

```
user@PE1# set protocols ospf area 0.0.0.0 interface ge-0/3/0.0 metric 100
user@PE1# set protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 100
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@PE1# set protocols ospf area 0.0.0.0 interface fxp0.0 disable

user@PE2# set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@PE2# set protocols ospf area 0.0.0.0 interface ge-1/3/0.0 metric 100
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@PE2# set protocols ospf area 0.0.0.0 interface fxp0.0 disable

user@PE3# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@PE3# set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@PE3# set protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 100
user@PE3# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

2. On the PE routers, configure a router ID.

```
user@PE1# set routing-options router-id 192.168.1.1

user@PE2# set routing-options router-id 192.168.2.1

user@PE3# set routing-options router-id 192.168.7.1
```

Use the **show ospf overview** and **show configuration protocols ospf** commands to verify that the correct interfaces have been configured for the OSPF protocol.

3. On the PE routers, configure OSPF traffic engineering support. Enabling traffic engineering extensions supports the Constrained Shortest Path First algorithm, which is needed to support Resource Reservation Protocol - Traffic Engineering (RSVP-TE) point-to-multipoint label-switched paths (LSPs). If you are configuring IS-IS, traffic engineering is supported without any additional configuration.

   user@PE1# **set protocols ospf traffic-engineering**

   user@PE2# **set protocols ospf traffic-engineering**

   user@PE3# **set protocols ospf traffic-engineering**

   Use the **show ospf overview** and **show configuration protocols ospf** commands to verify that traffic engineering support is enabled for the OSPF protocol.

4. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

5. On the PE routers, verify that the OSPF neighbors form adjacencies.

```
user@PE1> show ospf neighbors
Address          Interface              State     ID               Pri  Dead
10.0.17.14       fe-0/1/1.0             Full      192.168.7.1      128  32
10.0.12.10       ge-0/3/0.0             Full      192.168.2.1      128  33
```

   Verify that the neighbor state with the other two PE routers is **Full**.

Configuring BGP in the Core

Step-by-Step Procedure

1. On the PE routers, configure BGP. Configure the BGP local autonomous system number.

   user@PE1# **set routing-options autonomous-system 65000**

   user@PE2# **set routing-options autonomous-system 65000**

   user@PE3# **set routing-options autonomous-system 65000**

2. Configure the BGP peer groups. Configure the local address as the **lo0.0** address on the router. The neighbor addresses are the **lo0.0** addresses of the other PE routers.

The **unicast** statement enables the router to use BGP to advertise network layer reachability information (NLRI). The **signaling** statement enables the router to use BGP as the signaling protocol for the VPN.

```
user@PE1# set protocols bgp group group-mvpn type internal
user@PE1# set protocols bgp group group-mvpn local-address 192.168.1.1
user@PE1# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE1# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE1# set protocols bgp group group-mvpn neighbor 192.168.2.1
user@PE1# set protocols bgp group group-mvpn neighbor 192.168.7.1

user@PE2# set protocols bgp group group-mvpn type internal
user@PE2# set protocols bgp group group-mvpn local-address 192.168.2.1
user@PE2# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE2# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE2# set protocols bgp group group-mvpn neighbor 192.168.1.1
user@PE2# set protocols bgp group group-mvpn neighbor 192.168.7.1

user@PE3# set protocols bgp group group-mvpn type internal
user@PE3# set protocols bgp group group-mvpn local-address 192.168.7.1
user@PE3# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE3# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE3# set protocols bgp group group-mvpn neighbor 192.168.1.1
user@PE3# set protocols bgp group group-mvpn neighbor 192.168.2.1
```

3. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

4. On the PE routers, verify that the BGP neighbors form a peer session.

```
user@PE1> show bgp group
Group Type: Internal    AS: 65000                 Local AS: 65000
  Name: group-mvpn      Index: 0                  Flags: Export Eval
  Holdtime: 0
  Total peers: 2        Established: 2
  192.168.2.1+54883
  192.168.7.1+58933
  bgp.l3vpn.0: 0/0/0/0
  bgp.mvpn.0: 0/0/0/0

Groups: 1 Peers: 2   External: 0   Internal: 2   Down peers: 0  Flaps: 0
Table        Tot Paths Act Paths Suppressed  History Damp State  Pending
bgp.l3vpn.0          0          0         0        0         0       0
bgp.mvpn.0           0          0         0        0         0       0
```

Verify that the peer state for the other two PE routers is **Established** and that the **lo0.0** addresses of the other PE routers are shown as peers.

Configuring LDP

Step-by-Step Procedure

1. On the PE routers, configure LDP to support unicast traffic. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces between the PE routers. Also configure LDP specifying the **lo0.0** interface. As a best practice, disable LDP on the **fxp0** interface.

```
user@PE1# set protocols ldp deaggregate
user@PE1# set protocols ldp interface fe-0/1/1.0
user@PE1# set protocols ldp interface ge-0/3/0.0
user@PE1# set protocols ldp interface fxp0.0 disable
user@PE1# set protocols ldp interface lo0.0

user@PE2# set protocols ldp deaggregate
user@PE2# set protocols ldp interface fe-0/1/3.0
user@PE2# set protocols ldp interface ge-1/3/0.0
user@PE2# set protocols ldp interface fxp0.0 disable
user@PE2# set protocols ldp interface lo0.0

user@PE3# set protocols ldp deaggregate
user@PE3# set protocols ldp interface fe-0/1/1.0
user@PE3# set protocols ldp interface fe-0/1/3.0
user@PE3# set protocols ldp interface fxp0.0 disable
user@PE3# set protocols ldp interface lo0.0
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the PE routers, use the **show ldp route** command to verify the LDP route.

```
user@PE1> show ldp route
Destination         Next-hop intf/lsp               Next-hop address
 10.0.12.8/30       ge-0/3/0.0
 10.0.12.9/32
 10.0.17.12/30      fe-0/1/1.0
 10.0.17.13/32
 10.0.27.12/30      fe-0/1/1.0                      10.0.17.14
                    ge-0/3/0.0                      10.0.12.10
 192.168.1.1/32     lo0.0
 192.168.2.1/32     ge-0/3/0.0                      10.0.12.10
 192.168.7.1/32     fe-0/1/1.0                      10.0.17.14
 224.0.0.5/32
 224.0.0.22/32
```

Verify that a next-hop interface and next-hop address have been established for each remote destination in the core network. Notice that local destinations do not

have next-hop interfaces, and remote destinations outside the core do not have next-hop addresses.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, configure RSVP. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces that participate in the LSP. Also specify the **lo0.0** interface. As a best practice, disable RSVP on the **fxp0** interface.

   ```
   user@PE1# set protocols rsvp interface ge-0/3/0.0
   user@PE1# set protocols rsvp interface fe-0/1/1.0
   user@PE1# set protocols rsvp interface lo0.0
   user@PE1# set protocols rsvp interface fxp0.0 disable

   user@PE2# set protocols rsvp interface fe-0/1/3.0
   user@PE2# set protocols rsvp interface ge-1/3/0.0
   user@PE2# set protocols rsvp interface lo0.0
   user@PE2# set protocols rsvp interface fxp0.0 disable

   user@PE3# set protocols rsvp interface fe-0/1/3.0
   user@PE3# set protocols rsvp interface fe-0/1/1.0
   user@PE3# set protocols rsvp interface lo0.0
   user@PE3# set protocols rsvp interface fxp0.0 disable
   ```

2. On the PE routers, commit the configuration:

   ```
   user@host> commit check

   configuration check succeeds

   user@host> commit

   commit complete
   ```

   Verify these steps using the **show configuration protocols rsvp** command. You can verify the operation of RSVP only after the LSP is established.

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure MPLS. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces that participate in the LSP. As a best practice, disable MPLS on the fxp0 interface.

   ```
   user@PE1# set protocols mpls interface ge-0/3/0.0
   user@PE1# set protocols mpls interface fe-0/1/1.0
   user@PE1# set protocols mpls interface fxp0.0 disable

   user@PE2# set protocols mpls interface fe-0/1/3.0
   user@PE2# set protocols mpls interface ge-1/3/0.0
   user@PE2# set protocols mpls interface fxp0.0 disable
   ```

```
user@PE3# set protocols mpls interface fe-0/1/3.0
user@PE3# set protocols mpls interface fe-0/1/1.0
user@PE3# set protocols mpls interface fxp0.0 disable
```

Use the **show configuration protocols mpls** command to verify that the core-facing Fast Ethernet and Gigabit Ethernet interfaces are configured for MPLS.

2. On the PE routers, configure the core-facing interfaces associated with the LSP. Specify the **mpls** address family type.

```
user@PE1# set interfaces fe-0/1/1 unit 0 family mpls
user@PE1# set interfaces ge-0/3/0 unit 0 family mpls

user@PE2# set interfaces fe-0/1/3 unit 0 family mpls
user@PE2# set interfaces ge-1/3/0 unit 0 family mpls

user@PE3# set interfaces fe-0/1/3 unit 0 family mpls
user@PE3# set interfaces fe-0/1/1 unit 0 family mpls
```

Use the **show mpls interface** command to verify that the core-facing interfaces have the MPLS address family configured.

3. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

You can verify the operation of MPLS after the LSP is established.

Configuring the VRF Routing Instances

Step-by-Step Procedure

1. On Router PE1 , configure the routing instance for the green and red VPNs. Specify the **vrf** instance type and specify the customer-facing SONET interfaces.

   Configure a virtual tunnel (VT) interface on all MVPN routing instances on each PE where hosts in different instances need to receive multicast traffic from the same source.

```
user@PE1# set routing-instances green instance-type vrf
user@PE1# set routing-instances green interface so-0/0/3.0
user@PE1# set routing-instances green interface vt-1/2/0.1 multicast
user@PE1# set routing-instances green interface lo0.1

user@PE1# set routing-instances red instance-type vrf
user@PE1# set routing-instances red interface fe-0/1/0.0
user@PE1# set routing-instances red interface vt-1/2/0.2
user@PE1# set routing-instances red interface lo0.2
```

Use the **show configuration routing-instances green** and **show configuration routing-instances red** commands to verify that the virtual tunnel interfaces have been correctly configured.

2. On Router PE2 , configure the routing instance for the green VPN. Specify the **vrf** instance type and specify the customer-facing SONET interfaces.

```
user@PE2# set routing-instances green instance-type vrf
user@PE2# set routing-instances green interface so-0/0/1.0
user@PE2# set routing-instances green interface vt-1/2/0.1
user@PE2# set routing-instances green interface lo0.1
```

Use the **show configuration routing-instances green** command.

3. On Router PE3, configure the routing instance for the blue VPN. Specify the **vrf** instance type and specify the customer-facing SONET interfaces.

```
user@PE3# set routing-instances blue instance-type vrf
user@PE3# set routing-instances blue interface so-0/0/1.0
user@PE3# set routing-instances blue interface vt-1/2/0.3
user@PE3# set routing-instances blue interface lo0.1
```

Use the **show configuration routing-instances blue** command to verify that the instance type has been configured correctly and that the correct interfaces have been configured in the routing instance.

4. On Router PE1, configure a route distinguisher for the green and red routing instances. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes.

TIP: To help in troubleshooting, this example shows how to configure the route distinguisher to match the router ID. This allows you to associate a route with the router that advertised it.

```
user@PE1# set routing-instances green route-distinguisher 192.168.1.1:1
user@PE1# set routing-instances red route-distinguisher 192.168.1.1:2
```

5. On Router PE2, configure a route distinguisher for the green routing instance.

```
user@PE2# set routing-instances green route-distinguisher 192.168.2.1:1
```

6. On Router PE3, configure a route distinguisher for the blue routing instance.

```
user@PE3# set routing-instances blue route-distinguisher 192.168.7.1:3
```

7. On the PE routers, configure the VPN routing instance for multicast support.

```
user@PE1# set routing-instances green protocols mvpn
user@PE1# set routing-instances red protocols mvpn
```

```
user@PE2# set routing-instances green protocols mvpn

user@PE3# set routing-instances blue protocols mvpn
```

Use the **show configuration routing-instance** command to verify that the route distinguisher is configured correctly and that the MVPN Protocol is enabled in the routing instance.

8. On the PE routers, configure an IP address on additional loopback logical interfaces. These logical interfaces are used as the loopback addresses for the VPNs.

```
user@PE1# set interfaces lo0 unit 1 description "green VRF loopback"
user@PE1# set interfaces lo0 unit 1 family inet address 10.10.1.1/32
user@PE1# set interfaces lo0 unit 2 description "red VRF loopback"
user@PE1# set interfaces lo0 unit 2 family inet address 10.2.1.1/32

user@PE2# set interfaces lo0 unit 1 description "green VRF loopback"
user@PE2# set interfaces lo0 unit 1 family inet address 10.10.22.2/32

user@PE3# set interfaces lo0 unit 1 description "blue VRF loopback"
user@PE3# set interfaces lo0 unit 1 family inet address 10.3.33.3/32
```

Use the **show interfaces terse** command to verify that the loopback logical interfaces are correctly configured.

9. On the PE routers, configure virtual tunnel interfaces. These interfaces are used in VRF instances where multicast traffic arriving on a provider tunnel needs to be forwarded to multiple VPNs.

```
user@PE1# set interfaces vt-1/2/0 unit 1 description "green VRF multicast vt"
user@PE1# set interfaces vt-1/2/0 unit 1 family inet
user@PE1# set interfaces vt-1/2/0 unit 2 description "red VRF unicast and multicast
  vt"
user@PE1# set interfaces vt-1/2/0 unit 2 family inet
user@PE1# set interfaces vt-1/2/0 unit 3 description "blue VRF multicast vt"
user@PE1# set interfaces vt-1/2/0 unit 3 family inet

user@PE2# set interfaces vt-1/2/0 unit 1 description "green VRF unicast and
  multicast vt"
user@PE2# set interfaces vt-1/2/0 unit 1 family inet
user@PE2# set interfaces vt-1/2/0 unit 3 description "blue VRF unicast and multicast
  vt"
user@PE2# set interfaces vt-1/2/0 unit 3 family inet

user@PE3# set interfaces vt-1/2/0 unit 3 description "blue VRF unicast and multicast
  vt"
user@PE3# set interfaces vt-1/2/0 unit 3 family inet
```

Use the **show interfaces terse** command to verify that the virtual tunnel interfaces have the correct address family type configured.

10. On the PE routers, configure the provider tunnel.

```
user@PE1# set routing-instances green provider-tunnel rsvp-te
  label-switched-path-template default-template
user@PE1# set routing-instances red provider-tunnel rsvp-te
  label-switched-path-template default-template

user@PE2# set routing-instances green provider-tunnel rsvp-te
  label-switched-path-template default-template

user@PE3# set routing-instances blue provider-tunnel rsvp-te
  label-switched-path-template default-template
```

Use the **show configuration routing-instance** command to verify that the provider tunnel is configured to use the default LSP template.

NOTE: You cannot commit the configuration for the VRF instance until you configure the VRF target in the next section.

Configuring MVPN Extranet Policy

Step-by-Step Procedure

1. On the PE routers, define the VPN community name for the route targets for each VPN. The community names are used in the VPN import and export policies.

```
user@PE1# set policy-options community green-com members target:65000:1
user@PE1# set policy-options community red-com members target:65000:2
user@PE1# set policy-options community blue-com members target:65000:3

user@PE2# set policy-options community green-com members target:65000:1
user@PE2# set policy-options community red-com members target:65000:2
user@PE2# set policy-options community blue-com members target:65000:3

user@PE3# set policy-options community green-com members target:65000:1
user@PE3# set policy-options community red-com members target:65000:2
user@PE3# set policy-options community blue-com members target:65000:3
```

   Use the **show policy-options** command to verify that the correct VPN community name and route target are configured.

2. On the PE routers, configure the VPN import policy. Include the community name of the route targets that you want to accept. Do not include the community name of the route targets that you do not want to accept. For example, omit the community name for routes from the VPN of a multicast sender from which you do not want to receive multicast traffic.

```
user@PE1# set policy-options policy-statement green-red-blue-import term t1 from
  community green-com
user@PE1# set policy-options policy-statement green-red-blue-import term t1 from
  community red-com
user@PE1# set policy-options policy-statement green-red-blue-import term t1 from
  community blue-com
```

```
user@PE1# set policy-options policy-statement green-red-blue-import term t1 then
  accept
user@PE1# set policy-options policy-statement green-red-blue-import term t2 then
  reject

user@PE2# set policy-options policy-statement green-red-blue-import term t1 from
  community green-com
user@PE2# set policy-options policy-statement green-red-blue-import term t1 from
  community red-com
user@PE2# set policy-options policy-statement green-red-blue-import term t1 from
  community blue-com
user@PE2# set policy-options policy-statement green-red-blue-import term t1 then
  accept
user@PE2# set policy-options policy-statement green-red-blue-import term t2 then
  reject

user@PE3# set policy-options policy-statement green-red-blue-import term t1 from
  community green-com
user@PE3# set policy-options policy-statement green-red-blue-import term t1 from
  community red-com
user@PE3# set policy-options policy-statement green-red-blue-import term t1 from
  community blue-com
user@PE3# set policy-options policy-statement green-red-blue-import term t1 then
  accept
user@PE3# set policy-options policy-statement green-red-blue-import term t2 then
  reject
```

Use the **show policy green-red-blue-import** command to verify that the VPN import policy is correctly configured.

3. On the PE routers, apply the VRF import policy. In this example, the policy is defined in a **policy-statement** policy, and target communities are defined under the **[edit policy-options]** hierarchy level.

```
user@PE1# set routing-instances green vrf-import green-red-blue-import
user@PE1# set routing-instances red vrf-import green-red-blue-import

user@PE2# set routing-instances green vrf-import green-red-blue-import

user@PE3# set routing-instances blue vrf-import green-red-blue-import
```

Use the **show configuration routing-instances** command to verify that the correct VRF import policy has been applied.

4. On the PE routers, configure VRF export targets. The **vrf-target** statement and **export** option cause the routes being advertised to be labeled with the target community.

For Router PE3, the **vrf-target** statement is included without specifying the **export** option. If you do not specify the **import** or **export** options, default VRF import and export policies are generated that accept imported routes and tag exported routes with the specified target community.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@PE1# set routing-instances green vrf-target export target:65000:1
user@PE1# set routing-instances red vrf-target export target:65000:2

user@PE2# set routing-instances green vrf-target export target:65000:1

user@PE3# set routing-instances blue vrf-target target:65000:3
```

Use the **show configuration routing-instances** command to verify that the correct VRF export targets have been configured.

5. On the PE routers, configure automatic exporting of routes between VRF instances. When you include the **auto-export** statement, the **vrf-import** and **vrf-export** policies are compared across all VRF instances. If there is a common route target community between the instances, the routes are shared. In this example, the **auto-export** statement must be included under all instances that need to send traffic to and receive traffic from another instance located on the same router.

```
user@PE1# set routing-instances green routing-options auto-export
user@PE1# set routing-instances red routing-options auto-export

user@PE2# set routing-instances green routing-options auto-export

user@PE3# set routing-instances blue routing-options auto-export
```

6. On the PE routers, configure the load balance policy statement. While load balancing leads to better utilization of the available links, it is not required for MVPN extranets. It is included here as a best practice.

```
user@PE1# set policy-options policy-statement load-balance then load-balance
  per-packet

user@PE2# set policy-options policy-statement load-balance then load-balance
  per-packet

user@PE3# set policy-options policy-statement load-balance then load-balance
  per-packet
```

Use the **show policy-options** command to verify that the load balance policy statement has been correctly configured.

7. On the PE routers, apply the load balance policy.

```
user@PE1# set routing-options forwarding-table export load-balance

user@PE2# set routing-options forwarding-table export load-balance
```

```
user@PE3# set routing-options forwarding-table export load-balance
```

8. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

9. On the PE routers, use the **show rsvp neighbor** command to verify that the RSVP neighbors are established.

```
user@PE1> show rsvp neighbor
RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.0.17.14            5  1/0       43:52        9   293/293  247
10.0.12.10            0  1/0       50:15        9   336/336  140
```

Verify that the other PE routers are listed as RSVP neighbors.

10. On the PE routers, display the MPLS LSPs.

```
user@PE1> show mpls lsp p2mp
Ingress LSP: 2 sessions
P2MP name: 192.168.1.1:1:mvpn:green, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
192.168.2.1     192.168.1.1     Up     0 *
192.168.2.1:192.168.1.1:1:mvpn:green
192.168.7.1     192.168.1.1     Up     0 *
192.168.7.1:192.168.1.1:1:mvpn:green
P2MP name: 192.168.1.1:2:mvpn:red, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
192.168.2.1     192.168.1.1     Up     0 *
192.168.2.1:192.168.1.1:2:mvpn:red
192.168.7.1     192.168.1.1     Up     0 *
192.168.7.1:192.168.1.1:2:mvpn:red
Total 4 displayed, Up 4, Down 0

Egress LSP: 2 sessions
P2MP name: 192.168.2.1:1:mvpn:green, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.1     192.168.2.1     Up       0  1 SE  299888        3
192.168.1.1:192.168.2.1:1:mvpn:green
P2MP name: 192.168.7.1:3:mvpn:blue, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.1     192.168.7.1     Up       0  1 SE  299872        3
192.168.1.1:192.168.7.1:3:mvpn:blue
Total 2 displayed, Up 2, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

In this display from Router PE1, notice that there are two ingress LSPs for the green VPN and two for the red VPN configured on this router. Verify that the state of each

ingress LSP is **up**. Also notice that there is one egress LSP for each of the green and blue VPNs. Verify that the state of each egress LSP is **up**.

TIP: The LSP name displayed in the **show mpls lsp p2mp** command output can be used in the **ping mpls rsvp** <*lsp-name*> **multipath** command.

Configuring CE-PE BGP

Step-by-Step Procedure

1. On the PE routers, configure the BGP export policy. The BGP export policy is used to allow static routes and routes that originated from directly attached interfaces to be exported to BGP.

```
user@PE1# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@PE1# set policy-options policy-statement BGP-export term t1 then accept
user@PE1# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@PE1# set policy-options policy-statement BGP-export term t2 then accept

user@PE2# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@PE2# set policy-options policy-statement BGP-export term t1 then accept
user@PE2# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@PE2# set policy-options policy-statement BGP-export term t2 then accept

user@PE3# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@PE3# set policy-options policy-statement BGP-export term t1 then accept
user@PE3# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@PE3# set policy-options policy-statement BGP-export term t2 then accept
```

   Use the **show policy BGP-export** command to verify that the BGP export policy is correctly configured.

2. On the PE routers, configure the CE to PE BGP session. Use the IP address of the SONET interface as the neighbor address. Specify the autonomous system number for the VPN network of the attached CE router.

```
user@PE1# set routing-instances green protocols bgp group PE-CE export
  BGP-export
user@PE1# set routing-instances green protocols bgp group PE-CE neighbor 10.0.16.1
  peer-as 65001

user@PE2# set routing-instances green protocols bgp group PE-CE export
  BGP-export
user@PE2# set routing-instances green protocols bgp group PE-CE neighbor
  10.0.24.2 peer-as 65009
```

```
user@PE3# set routing-instances blue protocols bgp group PE-CE export BGP-export
user@PE3# set routing-instances blue protocols bgp group PE-CE neighbor 10.0.79.2
  peer-as 65003
```

3. On the CE routers, configure the BGP local autonomous system number.

```
user@CE1# set routing-options autonomous-system 65001

user@CE2# set routing-options autonomous-system 65009

user@CE3# set routing-options autonomous-system 65003
```

4. On the CE routers, configure the BGP export policy. The BGP export policy is used to allow static routes and routes that originated from directly attached interfaces to be exported to BGP.

```
user@CE1# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@CE1# set policy-options policy-statement BGP-export term t1 then accept
user@CE1# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@CE1# set policy-options policy-statement BGP-export term t2 then accept

user@CE2# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@CE2# set policy-options policy-statement BGP-export term t1 then accept
user@CE2# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@CE2# set policy-options policy-statement BGP-export term t2 then accept

user@CE3# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@CE3# set policy-options policy-statement BGP-export term t1 then accept
user@CE3# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@CE3# set policy-options policy-statement BGP-export term t2 then accept
```

Use the **show policy BGP-export** command to verify that the BGP export policy is correctly configured.

5. On the CE routers, configure the CE-to-PE BGP session. Use the IP address of the SONET interface as the neighbor address. Specify the autonomous system number of the core network. Apply the BGP export policy.

```
user@CE1# set protocols bgp group PE-CE export BGP-export
user@CE1# set protocols bgp group PE-CE neighbor 10.0.16.2 peer-as 65000

user@CE2# set protocols bgp group PE-CE export BGP-export
user@CE2# set protocols bgp group PE-CE neighbor 10.0.24.1 peer-as 65000

user@CE3# set protocols bgp group PE-CE export BGP-export
user@CE3# set protocols bgp group PE-CE neighbor 10.0.79.1 peer-as 65000
```

6. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

7. On the PE routers, use the **show bgp group pe-ce** command to verify that the BGP neighbors form a peer session.

```
user@PE1> show bgp group pe-ce
Group Type: External                                  Local AS: 65000
  Name: PE-CE           Index: 1                      Flags: <>
  Export: [ BGP-export ]
  Holdtime: 0
  Total peers: 1        Established: 1
  10.0.16.1+60500
  green.inet.0: 2/3/3/0
```

Verify that the peer state for the CE routers is **Established** and that the IP address configured on the peer SONET interface is shown as the peer.

Configuring PIM on the PE Routers

Step-by-Step Procedure

1. On the PE routers, enable an instance of PIM in each VPN. Configure the **lo0.1**, **lo0.2**, and customer-facing SONET and Fast Ethernet interfaces. Specify the mode as **sparse**.

```
user@PE1# set routing-instances green protocols pim interface lo0.1 mode sparse
user@PE1# set routing-instances green protocols pim interface so-0/0/3.0 mode
  sparse
user@PE1# set routing-instances red protocols pim interface lo0.2 mode sparse
user@PE1# set routing-instances red protocols pim interface fe-0/1/0.0 mode
  sparse

user@PE2# set routing-instances green protocols pim interface lo0.1 mode sparse
user@PE2# set routing-instances green protocols pim interface so-0/0/1.0 mode
  sparse

user@PE3# set routing-instances blue protocols pim interface lo0.1 mode sparse
user@PE3# set routing-instances blue protocols pim interface so-0/0/1.0 mode
  sparse
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit
```

```
commit complete
```

3. On the PE routers, use the **show pim interfaces instance green** command and substitute the appropriate VRF instance name to verify that the PIM interfaces are **up**.

```
user@PE1> show pim interfaces instance green
Instance: PIM.green

Name               Stat Mode       IP V State NbrCnt JoinCnt DR address
lo0.1              Up   Sparse      4 2 DR         0       0 10.10.1.1
lsi.0              Up   SparseDense 4 2 P2P        0       0
pe-1/2/0.32769     Up   Sparse      4 2 P2P        0       0
so-0/0/3.0         Up   Sparse      4 2 P2P        1       2
vt-1/2/0.1         Up   SparseDense 4 2 P2P        0       0
lsi.0              Up   SparseDense 6 2 P2P        0       0
```

Also notice that the normal mode for the virtual tunnel interface and label-switched interface is **SparseDense**.

Configuring PIM on the CE Routers

Step-by-Step Procedure

1. On the CE routers, configure the customer-facing and core-facing interfaces for PIM. Specify the mode as **sparse**.

```
user@CE1# set protocols pim interface fe-1/3/0.0 mode sparse
user@CE1# set protocols pim interface so-0/0/3.0 mode sparse

user@CE2# set protocols pim interface fe-0/1/1.0 mode sparse
user@CE2# set protocols pim interface so-0/0/1.0 mode sparse

user@CE3# set protocols pim interface fe-0/1/0.0 mode sparse
user@CE3# set protocols pim interface so-0/0/1.0 mode sparse
```

Use the **show pim interfaces** command to verify that the PIM interfaces have been configured to use sparse mode.

2. On the CE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the CE routers, use the **show pim interfaces** command to verify that the PIM interface status is **up**.

```
user@CE1> show pim interfaces
Instance: PIM.master

Name               Stat Mode       IP V State NbrCnt JoinCnt DR address
```

```
fe-1/3/0.0          Up   Sparse      4 2 DR        0       0 10.10.12.1
pe-1/2/0.32769      Up   Sparse      4 2 P2P       0       0
so-0/0/3.0          Up   Sparse      4 2 P2P       1       1
```

Configuring the Rendezvous Points

Step-by-Step Procedure

1. Configure Router PE1 to be the rendezvous point for the red VPN instance of PIM. Specify the local **lo0.2** address.

   user@PE1# set routing-instances red protocols pim rp local address 10.2.1.1

2. Configure Router PE2 to be the rendezvous point for the green VPN instance of PIM. Specify the **lo0.1** address of Router PE2.

   user@PE2# set routing-instances green protocols pim rp local address 10.10.22.2

3. Configure Router PE3 to be the rendezvous point for the blue VPN instance of PIM. Specify the local **lo0.1**.

   user@PE3# set routing-instances blue protocols pim rp local address 10.3.33.3

4. On the PE1, CE1, and CE2 routers, configure the static rendezvous point for the green VPN instance of PIM. Specify the **lo0.1** address of Router PE2.

   user@PE1# set routing-instances green protocols pim rp static address 10.10.22.2

   user@CE1# set protocols pim rp static address 10.10.22.2

   user@CE2# set protocols pim rp static address 10.10.22.2

5. On Router CE3, configure the static rendezvous point for the blue VPN instance of PIM. Specify the **lo0.1** address of Router PE3.

   user@CE3# set protocols pim rp static address 10.3.33.3

6. On the CE routers, commit the configuration:

   ```
   user@host> commit check

   configuration check succeeds

   user@host> commit

   commit complete
   ```

7. On the PE routers, use the **show pim rps instance <*instance-name*>** command and substitute the appropriate VRF instance name to verify that the RPs have been correctly configured.

   ```
   user@PE1> show pim rps instance <instance-name>
   Instance: PIM.green
   Address family INET
   ```

```
RP address                Type        Holdtime Timeout Groups Group prefixes
10.10.22.2                static             0    None      1 224.0.0.0/4

Address family INET6
```

Verify that the correct IP address is shown as the RP.

8. On the CE routers, use the **show pim rps** command to verify that the RP has been correctly configured.

```
user@CE1> show pim rps
Instance: PIM.master
Address family INET
RP address                Type        Holdtime Timeout Groups Group prefixes
10.10.22.2                static             0    None      1 224.0.0.0/4

Address family INET6
```

Verify that the correct IP address is shown as the RP.

9. On Router PE1, use the **show route table green.mvpn.0 | find 1** command to verify that the type-1 routes have been received from the PE2 and PE3 routers.

```
user@PE1> show route table green.mvpn.0 | find 1
green.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:192.168.1.1:1:192.168.1.1/240
                   *[MVPN/70] 03:38:09, metric2 1
                      Indirect
1:192.168.1.1:2:192.168.1.1/240
                   *[MVPN/70] 03:38:05, metric2 1
                      Indirect
1:192.168.2.1:1:192.168.2.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
1:192.168.7.1:3:192.168.7.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.7.1
                      AS path: I
                    > to 10.0.17.14 via fe-0/1/1.0
```

10. On Router PE1, use the **show route table green.mvpn.0 | find 5** command to verify that the type-5 routes have been received from Router PE2.

A designated router (DR) sends periodic join messages and prune messages toward a group-specific rendezvous point (RP) for each group for which it has active members. When a PIM router learns about a source, it originates a Multicast Source Discovery Protocol (MSDP) source-address message if it is the DR on the upstream interface. If an MBGP MVPN is also configured, the PE device originates a type-5 MVPN route.

```
user@PE1> show route table green.mvpn.0 | find 5
5:192.168.2.1:1:32:10.10.12.52:32:224.1.1.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
```

```
                    AS path: I
                  > to 10.0.12.10 via ge-0/3/0.0
```

11. On Router PE1, use the **show route table green.mvpn.0 | find 7** command to verify that the type-7 routes have been received from Router PE2.

```
user@PE1> show route table green.mvpn.0 | find 7
7:192.168.1.1:1:65000:32:10.10.12.52:32:224.1.1.1/240
                   *[MVPN/70] 03:22:47, metric2 1
                      Multicast (IPv4)
                    [PIM/105] 03:34:18
                      Multicast (IPv4)
                    [BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
```

12. On Router PE1, use the **show route advertising-protocol bgp 192.168.2.1 table green.mvpn.0 detail** command to verify that the routes advertised by Router PE2 use the PMSI attribute set to RSVP-TE.

```
user@PE1> show route advertising-protocol bgp 192.168.2.1 table green.mvpn.0 detail
green.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
* 1:192.168.1.1:1:192.168.1.1/240 (1 entry, 1 announced)
 BGP group group-mvpn type Internal
     Route Distinguisher: 192.168.1.1:1
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:65000:1
     PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[192.168.1.1:0:56822:192.168.1.1]
```

Testing MVPN Extranets

Step-by-Step Procedure

1. Start the multicast receiver device connected to Router CE2.
2. Start the multicast sender device connected to Router CE1.
3. Verify that the receiver receives the multicast stream.
4. On Router PE1, display the provider tunnel to multicast group mapping by using the **show mvpn c-multicast** command.

```
user@PE1> show mvpn c-multicast
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: green
  C-mcast IPv4 (S:G)            Ptnl                              St
  10.10.12.52/32:224.1.1.1/32   RSVP-TE P2MP:192.168.1.1, 56822,192.168.1.1         RM
```

```
  0.0.0.0/0:239.255.255.250/32
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: red
  C-mcast IPv4 (S:G)            Ptnl                              St
  10.10.12.52/32:224.1.1.1/32                                        DS
  0.0.0.0/0:224.1.1.1/32
```

5. On Router PE2, use the **show route table green.mvpn.0 | find 6** command to verify that the type-6 routes have been created as a result of receiving PIM join messages.

```
user@PE2> show route table green.mvpn.0 | find 6
6:192.168.2.1:1:65000:32:10.10.22.2:32:224.1.1.1/240
                   *[PIM/105] 04:01:23
                      Multicast (IPv4)
6:192.168.2.1:1:65000:32:10.10.22.2:32:239.255.255.250/240
                   *[PIM/105] 22:39:46
                      Multicast (IPv4)
```

NOTE: The multicast address 239.255.255.250 shown in the preceding step is not related to this example. This address is sent by some host machines.

6. Start the multicast receiver device connected to Router CE3.

7. Verify that the receiver is receiving the multicast stream.

8. On Router PE2, use the **show route table green.mvpn.0 | find 6** command to verify that the type-6 routes have been created as a result of receiving PIM join messages from the multicast receiver device connected to Router CE3.

```
user@PE2> show route table green.mvpn.0 | find 6
6:192.168.2.1:1:65000:32:10.10.22.2:32:239.255.255.250/240
                   *[PIM/105] 06:43:39
                      Multicast (IPv4)
```

9. Start the multicast receiver device directly connected to Router PE1.

10. Verify that the receiver is receiving the multicast stream.

11. On Router PE1, use the **show route table green.mvpn.0 | find 6** command to verify that the type-6 routes have been created as a result of receiving PIM join messages from the directly connected multicast receiver device.

```
user@PE1> show route table green.mvpn.0 | find 6
6:192.168.1.1:2:65000:32:10.2.1.1:32:224.1.1.1/240
                   *[PIM/105] 00:02:32
                      Multicast (IPv4)
6:192.168.1.1:2:65000:32:10.2.1.1:32:239.255.255.250/240
                   *[PIM/105] 00:05:49
                      Multicast (IPv4)
```

NOTE: The multicast address 255.255.255.250 shown in the step above is not related to this example.

Results

The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router CE1 follows.

Router CE1

```
interfaces {
   so-0/0/3 {
      unit 0 {
         description "to PE1 so-0/0/3.0";
         family inet {
            address 10.0.16.1/30;
         }
      }
   }
   fe-1/3/0 {
      unit 0 {
         family inet {
            address 10.10.12.1/24;
         }
      }
   }
   lo0 {
      unit 0 {
         description "CE1 Loopback";
         family inet {
            address 192.168.6.1/32 {
               primary;
            }
            address 127.0.0.1/32;
         }
      }
   }
}
routing-options {
   autonomous-system 65001;
   router-id 192.168.6.1;
   forwarding-table {
      export load-balance;
```

```
    }
  }
  protocols {
    bgp {
      group PE-CE {
        export BGP-export;
        neighbor 10.0.16.2 {
          peer-as 65000;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.10.22.2;
        }
      }
      interface fe-1/3/0.0 {
        mode sparse;
      }
      interface so-0/0/3.0 {
        mode sparse;
      }
    }
  }
  policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement load-balance {
      then {
        load-balance per-packet;
      }
    }
  }
```

The relevant sample configuration for Router PE1 follows.

Router PE1

```
interfaces {
  so-0/0/3 {
    unit 0 {
      description "to CE1 so-0/0/3.0";
      family inet {
        address 10.0.16.2/30;
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
```

```
        description "to H2";
        family inet {
          address 10.2.11.2/30;
        }
      }
    }
      fe-0/1/1 {
        unit 0 {
          description "to PE3 fe-0/1/1.0";
          family inet {
            address 10.0.17.13/30;
          }
          family mpls;
        }
      }
    ge-0/3/0 {
      unit 0 {
        description "to PE2 ge-1/3/0.0";
        family inet {
          address 10.0.12.9/30;
        }
        family mpls;
      }
    }
    vt-1/2/0 {
      unit 1 {
        description "green VRF multicast vt";
        family inet;
      }
      unit 2 {
        description "red VRF unicast and multicast vt";
        family inet;
      }
      unit 3 {
        description "blue VRF multicast vt";
        family inet;
      }
    }
    lo0 {
      unit 0 {
        description "PE1 Loopback";
        family inet {
          address 192.168.1.1/32 {
            primary;
          }
          address 127.0.0.1/32;
        }
      }
      unit 1 {
        description "green VRF loopback";
        family inet {
          address 10.10.1.1/32;
        }
      }
      unit 2 {
        description "red VRF loopback";
```

```
        family inet {
          address 10.2.1.1/32;
        }
      }
    }
}
routing-options {
  autonomous-system 65000;
  router-id 192.168.1.1;
  forwarding-table {
    export load-balance;
  }
}
protocols {
  rsvp {
    interface ge-0/3/0.0;
    interface fe-0/1/1.0;
    interface lo0.0;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    interface ge-0/3/0.0;
    interface fe-0/1/1.0;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group group-mvpn {
      type internal;
      local-address 192.168.1.1;
      family inet-vpn {
        unicast;
      }
      family inet-mvpn {
        signaling;
      }
      neighbor 192.168.2.1;
      neighbor 192.168.7.1;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface ge-0/3/0.0 {
        metric 100;
      }
      interface fe-0/1/1.0 {
        metric 100;
      }
      interface lo0.0 {
        passive;
      }
      interface fxp0.0 {
```

```
            disable;
          }
        }
      }
      ldp {
        deaggregate;
        interface ge-0/3/0.0;
        interface fe-0/1/1.0;
        interface fxp0.0 {
          disable;
        }
        interface lo0.0;
      }
    }
    policy-options {
      policy-statement BGP-export {
        term t1 {
          from protocol direct;
          then accept;
        }
        term t2 {
          from protocol static;
          then accept;
        }
      }
      policy-statement green-red-blue-import {
        term t1 {
          from community [ green-com red-com blue-com ];
          then accept;
        }
        term t2 {
          then reject;
        }
      }
      policy-statement load-balance {
        then {
          load-balance per-packet;
        }
      }
      community green-com members target:65000:1;
      community red-com members target:65000:2;
      community blue-com members target:65000:3;
    }
    routing-instances {
      green {
        instance-type vrf;
        interface so-0/0/3.0;
        interface vt-1/2/0.1 {
          multicast;
        }
        interface lo0.1;
        route-distinguisher 192.168.1.1:1;
        provider-tunnel {
          rsvp-te {
            label-switched-path-template {
              default-template;
```

```
        }
      }
    }
    vrf-import green-red-blue-import;
    vrf-target export target:65000:1;
    vrf-table-label;
    routing-options {
      auto-export;
    }
    protocols {
      bgp {
        group PE-CE {
          export BGP-export;
          neighbor 10.0.16.1 {
            peer-as 65001;
          }
        }
      }
      pim {
        rp {
          static {
            address 10.10.22.2;
          }
        }
        interface so-0/0/3.0 {
          mode sparse;
        }
        interface lo0.1 {a
        mode sparse;
      }
    }
    mvpn;
  }
  red {
    instance-type vrf;
    interface fe-0/1/0.0;
    interface vt-1/2/0.2;
    interface lo0.2;
    route-distinguisher 192.168.1.1:2;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    vrf-import green-red-blue-import;
    vrf-target export target:65000:2;
    routing-options {
      auto-export;
    }
    protocols {
      pim {
        rp {
          local {
            address 10.2.1.1;
```

```
            }
          }
          interface fe-0/1/0.0 {
            mode sparse;
          }
          interface lo0.2 {
            mode sparse;
          }
        }
        mvpn;
      }
    }
  }
```

The relevant sample configuration for Router PE2 follows.

Router PE2

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to CE2 so-0/0/1:0.0";
      family inet {
        address 10.0.24.1/30;
      }
    }
  }
  fe-0/1/3 {
    unit 0 {
      description "to PE3 fe-0/1/3.0";
      family inet {
        address 10.0.27.13/30;
      }
      family mpls;
    }
    vt-1/2/0 {
      unit 1 {
        description "green VRF unicast and multicast vt";
        family inet;
      }
      unit 3 {
        description "blue VRF unicast and multicast vt";
        family inet;
      }
    }
  }
  ge-1/3/0 {
    unit 0 {
      description "to PE1 ge-0/3/0.0";
      family inet {
        address 10.0.12.10/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      description "PE2 Loopback";
```

```
      family inet {
        address 192.168.2.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
    unit 1 {
      description "green VRF loopback";
      family inet {
        address 10.10.22.2/32;
      }
    }
  }
  routing-options {
    router-id 192.168.2.1;
    autonomous-system 65000;
    forwarding-table {
      export load-balance;
    }
  }
  protocols {
    rsvp {
      interface fe-0/1/3.0;
      interface ge-1/3/0.0;
      interface lo0.0;
      interface fxp0.0 {
        disable;
      }
    }
    mpls {
      interface fe-0/1/3.0;
      interface ge-1/3/0.0;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group group-mvpn {
        type internal;
        local-address 192.168.2.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.1.1;
        neighbor 192.168.7.1;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface fe-0/1/3.0 {
          metric 100;
```

```
        }
        interface ge-1/3/0.0 {
          metric 100;
        }
        interface lo0.0 {
          passive;
        }
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      deaggregate;
      interface fe-0/1/3.0;
      interface ge-1/3/0.0;
      interface fxp0.0 {
        disable;
      }
      interface lo0.0;
    }
}
policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement green-red-blue-import {
      term t1 {
        from community [ green-com red-com blue-com ];
        then accept;
      }
      term t2 {
        then reject;
      }
    }
    policy-statement load-balance {
      then {
        load-balance per-packet;
      }
    }
    community green-com members target:65000:1;
    community red-com members target:65000:2;
    community blue-com members target:65000:3;
}
routing-instances {
    green {
      instance-type vrf;
      interface so-0/0/1.0;
      interface vt-1/2/0.1;
```

```
        interface lo0.1;
        route-distinguisher 192.168.2.1:1;
        provider-tunnel {
          rsvp-te {
            label-switched-path-template {
              default-template;
            }
          }
        }
        vrf-import green-red-blue-import;
        vrf-target export target:65000:1;
        routing-options {
          auto-export;
        }
        protocols {
          bgp {
            group PE-CE {
              export BGP-export;
              neighbor 10.0.24.2 {
                peer-as 65009;
              }
            }
          }
          pim {
            rp {
              local {
                address 10.10.22.2;
              }
            }
            interface so-0/0/1.0 {
              mode sparse;
            }
            interface lo0.1 {
              mode sparse;
            }
          }
          mvpn;
        }
      }
    }
  }
```

The relevant sample configuration for Router CE2 follows.

Router CE2

```
interfaces {
  fe-0/1/1 {
    unit 0 {
      description "to H4";
      family inet {
        address 10.10.11.2/24;
      }
    }
  }
  so-0/0/1 {
    unit 0 {
      description "to PE2 so-0/0/1";
```

```
        family inet {
          address 10.0.24.2/30;
        }
      }
    }
    lo0 {
      unit 0 {
        description "CE2 Loopback";
        family inet {
          address 192.168.4.1/32 {
            primary;
          }
          address 127.0.0.1/32;
        }
      }
    }
}
routing-options {
    router-id 192.168.4.1;
    autonomous-system 65009;
    forwarding-table {
      export load-balance;
    }
}
protocols {
    bgp {
      group PE-CE {
        export BGP-export;
        neighbor 10.0.24.1 {
          peer-as 65000;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.10.22.2;
        }
      }
      interface so-0/0/1.0 {
        mode sparse;
      }
      interface fe-0/1/1.0 {
        mode sparse;
      }
    }
}
policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
```

```
    }
  }
  policy-statement load-balance {
    then {
      load-balance per-packet;
    }
  }
}
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to CE3 so-0/0/1.0";
      family inet {
        address 10.0.79.1/30;
      }
    }
  }
  fe-0/1/1 {
    unit 0 {
      description "to PE1 fe-0/1/1.0";
      family inet {
        address 10.0.17.14/30;
      }
      family mpls;
    }
  }
  fe-0/1/3 {
    unit 0 {
      description "to PE2 fe-0/1/3.0";
      family inet {
        address 10.0.27.14/30;
      }
      family mpls;
    }
  }
  vt-1/2/0 {
    unit 3 {
      description "blue VRF unicast and multicast vt";
      family inet;
    }
  }
  lo0 {
    unit 0 {
      description "PE3 Loopback";
      family inet {
        address 192.168.7.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
    unit 1 {
      description "blue VRF loopback";
```

```
        family inet {
          address 10.3.33.3/32;
        }
      }
    }
}
routing-options {
  router-id 192.168.7.1;
  autonomous-system 65000;
  forwarding-table {
    export load-balance;
  }
}
protocols {
  rsvp {
    interface fe-0/1/3.0;
    interface fe-0/1/1.0;
    interface lo0.0;
    interface fxp0.0 {
      disable;
    }
  }
  mpls {
    interface fe-0/1/3.0;
    interface fe-0/1/1.0;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group group-mvpn {
      type internal;
      local-address 192.168.7.1;
      family inet-vpn {
        unicast;
      }
      family inet-mvpn {
        signaling;
      }
      neighbor 192.168.1.1;
      neighbor 192.168.2.1;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface fe-0/1/3.0 {
        metric 100;
      }
      interface fe-0/1/1.0 {
        metric 100;
      }
      interface lo0.0 {
        passive;
      }
      interface fxp0.0 {
```

```
            disable;
          }
        }
      }
      ldp {
        deaggregate;
        interface fe-0/1/3.0;
        interface fe-0/1/1.0;
        interface fxp0.0 {
          disable;
        }
        interface lo0.0;
      }
}
policy-options {
    policy-statement BGP-export {
        term t1 {
            from protocol direct;
            then accept;
        }
        term t2 {
            from protocol static;
            then accept;
        }
    }
    policy-statement green-red-blue-import {
        term t1 {
            from community [ green-com red-com blue-com ];
            then accept;
        }
        term t2 {
            then reject;
        }
    }
    policy-statement load-balance {
        then {
            load-balance per-packet;
        }
    }
    community green-com members target:65000:1;
    community red-com members target:65000:2;
    community blue-com members target:65000:3;
}
routing-instances {
    blue {
        instance-type vrf;
        interface vt-1/2/0.3;
        interface so-0/0/1.0;
        interface lo0.1;
        route-distinguisher 192.168.7.1:3;
        provider-tunnel {
            rsvp-te {
                label-switched-path-template {
                    default-template;
                }
            }
```

```
    }
    vrf-import green-red-blue-import;
    vrf-target target:65000:3;
    routing-options {
      auto-export;
    }
    protocols {
      bgp {
        group PE-CE {
          export BGP-export;
          neighbor 10.0.79.2 {
            peer-as 65003;
          }
        }
      }
      pim {
        rp {
          local {
            address 10.3.33.3;
          }
        }
        interface so-0/0/1.0 {
          mode sparse;
        }
        interface lo0.1 {
          mode sparse;
        }
      }
      mvpn ;
    }
  }
}
```

The relevant sample configuration for Router CE3 follows.

Router CE3

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to PE3";
      family inet {
        address 10.0.79.2/30;
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
      description "to H3";
      family inet {
        address 10.3.11.3/24;
      }
    }
  }
  lo0 {
    unit 0 {
      description "CE3 loopback";
      family inet {
```

```
            address 192.168.9.1/32 {
              primary;
            }
            address 127.0.0.1/32;
          }
        }
      }
    }
    routing-options {
      router-id 192.168.9.1;
      autonomous-system 65003;
      forwarding-table {
        export load-balance;
      }
    }
    protocols {
      bgp {
        group PE-CE {
          export BGP-export;
          neighbor 10.0.79.1 {
            peer-as 65000;
          }
        }
      }
      pim {
        rp {
          static {
            address 10.3.33.3;
          }
        }
        interface so-0/0/1.0 {
          mode sparse;
        }
        interface fe-0/1/0.0 {
          mode sparse;
        }
      }
    }
    policy-options {
      policy-statement BGP-export {
        term t1 {
          from protocol direct;
          then accept;
        }
        term t2 {
          from protocol static;
          then accept;
        }
      }
      policy-statement load-balance {
        then {
          load-balance per-packet;
        }
      }
    }
```

See Also

Related Documentation

- Configuring Multiprotocol BGP Multicast VPNs on page 584
- Multiprotocol BGP MVPNs Overview on page 526

Understanding Redundant Virtual Tunnel Interfaces in MBGP MVPNs

In multiprotocol BGP (MBGP) multicast VPNs (MVPNs), VT interfaces are needed for multicast traffic on routing devices that function as combined provider edge (PE) and provider core (P) routers to optimize bandwidth usage on core links. VT interfaces prevent traffic replication when a P router also acts as a PE router (an exit point for multicast traffic).

Starting in Junos OS Release 12.3, you can configure up to eight VT interfaces in a routing instance, thus providing Tunnel PIC redundancy inside the same multicast VPN routing instance. When the active VT interface fails, the secondary one takes over, and you can continue managing multicast traffic with no duplication.

Redundant VT interfaces are supported with RSVP point-to-multipoint provider tunnels as well as multicast LDP provider tunnels. This feature also works for extranets.

You can configure one of the VT interfaces to be the primary interface. If a VT interface is configured as the primary, it becomes the next hop that is used for traffic coming in from the core on the label-switched path (LSP) into the routing instance. When a VT interface is configured to be primary and the VT interface is used for both unicast and multicast traffic, only the multicast traffic is affected.

If no VT interface is configured to be the primary or if the primary VT interface is unusable, one of the usable configured VT interfaces is chosen to be the next hop that is used for traffic coming in from the core on the LSP into the routing instance. If the VT interface in use goes down for any reason, another usable configured VT interface in the routing instance is chosen. When the VT interface in use changes, all multicast routes in the instance also switch their reverse-path forwarding (RPF) interface to the new VT interface to allow the traffic to be received.

To realize the full benefit of redundancy, we recommend that when you configure multiple VT interfaces, at least one of the VT interfaces be on a different Tunnel PIC from the other VT interfaces. However, Junos OS does not enforce this.

Release History Table

Release	Description
12.3	Starting in Junos OS Release 12.3, you can configure up to eight VT interfaces in a routing instance, thus providing Tunnel PIC redundancy inside the same multicast VPN routing instance.

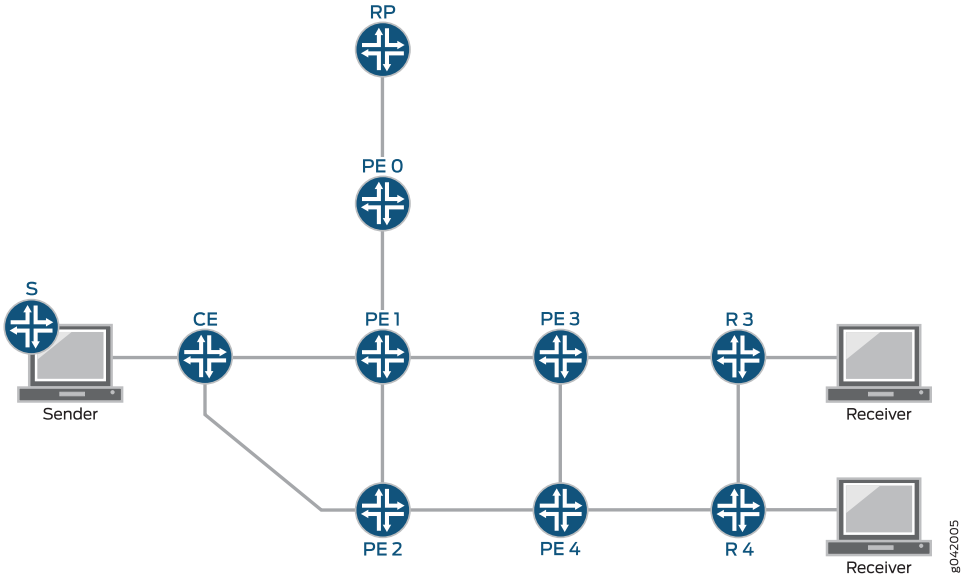
Related Documentation

- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 744

Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels

In a BGP multicast VPN (MVPN) (also called a multiprotocol BGP next-generation multicast VPN), sender-based reverse-path forwarding (RPF) helps to prevent multiple provider edge (PE) routers from sending traffic into the core, thus preventing duplicate traffic being sent to a customer. In the following diagram, sender-based RPF configured on egress Device PE3 and Device PE4 prevents duplicate traffic from being sent to the customers.

Figure 108: Sender-Based RPF



Sender-based RPF is supported on MX Series platforms with MPC line cards. As a prerequisite, the router must be set to **network-services enhanced-ip** mode.

Sender-based RPF (and hot-root standby) are supported only for MPLS BGP MVPNs with RSVP point-to-multipoint provider tunnels. Both SPT-only and SPT-RPT MVPN modes are supported.

Sender-based RPF does not work when point-to-multipoint provider tunnels are used with label-switched interfaces (LSI). Junos OS only allocates a single LSI label for each VRF, and uses this label for all point-to-multipoint tunnels. Therefore, the label that the egress receives does not indicate the sending PE router. LSI labels currently cannot scale to create a unique label for each point-to-multipoint tunnel. As such, virtual tunnel interfaces (vt) must be used for sender-based RPF functionality with point-to-multipoint provider tunnels.

Optionally, LSI interfaces can continue to be used for unicast purposes, and virtual tunnel interfaces can be configured to be used for multicast only.

In general, it is important to avoid (or recover from) having multiple PE routers send duplicate traffic into the core because this can result in duplicate traffic being sent to

the customer. The sender-based RPF has a use case that is limited to BGP MVPNs. The use-case scope is limited for the following reasons:

- A traditional RPF check for native PIM is based on the incoming interface. This RPF check prevents loops but does not prevent multiple forwarders on a LAN. The traditional RPF has been used because current multicast protocols either avoid duplicates on a LAN or have data-driven events to resolve the duplicates once they are detected.
- In PIM sparse mode, duplicates can occur on a LAN in normal protocol operation. The protocol has a data-driven mechanism (PIM assert messages) to detect duplication when it happens and resolve it.
- In PIM bidirectional mode, a designated forwarder (DF) election is performed on all LANs to avoid duplication.
- Draft Rosen MVPNs use the PIM assert mechanism because with Draft Rosen MVPNs the core network is analogous to a LAN.

Sender-based RPF is a solution to be used in conjunction with BGP MVPNs because BGP MVPNs use an alternative to data-driven-event solutions and bidirectional mode DF election. This is so, because, for one thing, the core network is not exactly a LAN. In an MVPN scenario, it is possible to determine which PE router has sent the traffic. Junos OS uses this information to only forward the traffic if it is sent from the correct PE router. With sender-based RPF, the RPF check is enhanced to check whether data arrived on the correct incoming virtual tunnel (vt-) interface and that the data was sent from the correct upstream PE router.

More specifically, the data must arrive with the correct MPLS label in the outer header used to encapsulate data through the core. The label identifies the tunnel and, if the tunnel is point-to-multipoint, the upstream PE router.

Sender-based RPF is not a replacement for single-forwarder election, but is a complementary feature. Configuring a higher primary loopback address (or router ID) on one PE device (PE1) than on another (PE2) ensures that PE1 is the single-forwarder election winner. The unicast-umh-election statement causes the unicast route preference to determine the single-forwarder election. If single-forwarder election is not used or if it is not sufficient to prevent duplicates in the core, sender-based RPF is recommended.

For RSVP point-to-multipoint provider tunnels, the transport label identifies the sending PE router because it is a requirement that penultimate hop popping (PHP) is disabled when using point-to-multipoint provider tunnels with MVPNs. PHP is disabled by default when you configure the MVPN protocol in a routing instance. The label identifies the tunnel, and (because the RSVP-TE tunnel is point-to-multipoint) the sending PE router.

The sender-based RPF mechanism is described in RFC 6513, *Multicast in MPLS/BGP IP VPNs* in section 9.1.1.

NOTE: The hot-root standby technique described in Internet draft draft-morin-l3vpn-mvpn-fast-failover-05 *Multicast VPN fast upstream failover* is an egress PE router functionality in which the egress PE router sends source-tree c-multicast join message to both a primary and a backup upstream PE router. This allows multiple copies of the traffic to flow through the provider core to the egress PE router. Sender-based RPF and hot-root standby can be used together to support *live-live* BGP MVPN traffic. This is a multicast-over-MPLS scheme for carrying mission-critical professional broadcast TV and IPTV traffic. A key requirement for many of these deployments is to have full redundancy of network equipment, including the ingress and egress PE routers. In some cases, a live-live approach is required, meaning that two duplicate traffic flows are sent across the network following diverse paths. When this technique is combined with sender-based forwarding, the two live flows of traffic are received at the egress PE router, and the egress PE router forwards a single stream to the customer network. Any failure in the network can be repaired locally at the egress PE router. For more information about hot-root standby, see hot-root-standby.

Sender-based RPF prevents duplicates from being sent to the customer even if there is duplication in the provider network. Duplication could exist in the provider because of a hot-root standby configuration or if the single-forwarder election is not sufficient to prevent duplicates. Single-forwarder election is used to prevent duplicates to the core network, while sender-based RPF prevents duplicates to the customer even if there are duplicates in the core. There are cases in which single-forwarder election cannot prevent duplicate traffic from arriving at the egress PE router. One example of this (outlined in section 9.3.1 of RFC 6513) is when PIM sparse mode is configured in the customer network and the MVPN is in RPT-SPT mode with an I-PMSI.

Determining the Upstream PE Router

After Junos OS chooses the ingress PE router, the sender-based RPF decision determines whether the correct ingress PE router is selected. As described in RFC 6513, section 9.1.1, an egress PE router, PE1, chooses a specific upstream PE router, for given (C-S,C-G). When PE1 receives a (C-S,C-G) packet from a PMSI, it might be able to identify the PE router that transmitted the packet onto the PMSI. If that transmitter is other than the PE router selected by PE1 as the upstream PE router, PE1 can drop the packet. This means that the PE router detects a duplicate, but the duplicate is not forwarded.

When an egress PE router generates a type 7 C-multicast route, it uses the VRF route import extended community carried in the VPN-IP route toward the source to construct the route target carried by the C-multicast route. This route target results in the C-multicast route being sent to the upstream PE router, and being imported into the correct VRF on the upstream PE router. The egress PE router programs the forwarding entry to only accept traffic from this PE router, and only on a particular tunnel rooted at that PE router.

When an egress PE router generates a type 6 C-multicast route, it uses the VRF route import extended community carried in the VPN-IP route toward the rendezvous point (RP) to construct the route target carried by the C-multicast route.

This route target results in the C-multicast route being sent to the upstream PE router and being imported into the correct VRF on the upstream PE router. The egress PE router programs the forwarding entry to accept traffic from this PE router only, and only on a particular tunnel rooted at that PE router. However, if some other PE routers have switched to SPT mode for (C-S, C-G) and have sent source active (SA) autodiscovery (A-D) routes (type 5 routes), and if the egress PE router only has (C-*, C-G) state, the upstream PE router for (C-S, C-G) is not the PE router toward the RP to which it sent a type 6 route, but the PE router that originates a SA A-D route for (C-S, C-G). The traffic for (C-S, C-G) might be carried over a I-PMSI or S-PMSI, depending on how it was advertised by the upstream PE router.

Additionally, when an egress PE router has only the (C-*, C-G) state and does not have the (C-S, C-G) state, the egress PE router might be receiving (C-S, C-G) type 5 SA routes from multiple PE routers, and chooses the best one, as follows: For every received (C-S, C-G) SA route, the egress PE router finds in its upstream multicast hop (UMH) route-candidate set for C-S a route with the same route distinguisher (RD). Among all such found routes the PE router selects the UMH route (based on the UMH selection). The best (C-S, C-G) SA route is the one whose RD is the same as of the selected UMH route.

When an egress PE router has only the (C-*, C-G) state and does not have the (C-S, C-G) state, and if later the egress PE router creates the (C-S, C-G) state (for example, as a result of receiving a PIM join (C-S, C-G) message from one of its customer edge [CE] neighbors), the upstream PE router for that (C-S, C-G) is not necessarily going to be the same PE router that originated the already-selected best SA A-D route for (C-S, C-G). It is possible to have a situation in which the PE router that originated the best SA A-D route for (C-S, C-G) carries the (C-S, C-G) over an I-PMSI, while some other PE router, that is also connected to the site that contains C-S, carries (C-S,C-G) over an S-PMSI. In this case, the downstream PE router would not join the S-PMSI, but continue to receive (C-S, C-G) over the I-PMSI, because the UMH route for C-S is the one that has been advertised by the PE router that carries (C-S, C-G) over the I-PMSI. This is expected behavior.

The egress PE router determines the sender of a (C-S, C-G) type 5 SA A-D route by finding in its UMH route-candidate set for C-S a route whose RD is the same as in the SA A-D route. The VRF route import extended community of the found route contains the IP address of the sender of the SA A-D route.

Related Documentation

- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- unicast-umh-election on page 1364

Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels

This example shows how to configure sender-based reverse-path forwarding (RPF) in a BGP multicast VPN (MVPN). Sender-based RPF helps to prevent multiple provider edge (PE) routers from sending traffic into the core, thus preventing duplicate traffic being sent to a customer.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Sender-based RPF is supported on MX Series platforms with MPC line cards. As a prerequisite, the router must be set to **network-services enhanced-ip** mode.

Sender-based RPF is supported only for MPLS BGP MVPNs with RSVP-TE point-to-multipoint provider tunnels. Both SPT-only and SPT-RPT MVPN modes are supported.

Sender-based RPF does not work when point-to-multipoint provider tunnels are used with label-switched interfaces (LSI). Junos OS only allocates a single LSI label for each VRF, and uses this label for all point-to-multipoint tunnels. Therefore, the label that the egress receives does not indicate the sending PE router. LSI labels currently cannot scale to create a unique label for each point-to-multipoint tunnel. As such, virtual tunnel interfaces (vt) must be used for sender-based RPF functionality with point-to-multipoint provider tunnels.

This example requires Junos OS Release 14.2 or later on the PE router that has sender-based RPF enabled.

Overview

This example shows a single autonomous system (intra-AS scenario) in which one source sends multicast traffic (group 224.1.1.1) into the VPN (VRF instance vpn-1). Two receivers subscribe to the group. They are connected to Device CE2 and Device CE3, respectively. RSVP point-to-multipoint LSPs with inclusive provider tunnels are set up among the PE routers. PIM (C-PIM) is configured on the PE-CE links.

For MPLS, the signaling control protocol used here is LDP. Optionally, you can use RSVP to signal both point-to-point and point-to-multipoint tunnels.

OSPF is used for interior gateway protocol (IGP) connectivity, though IS-IS is also a supported option. If you use OSPF, you must enable OSPF traffic engineering.

For testing purposes, routers are used to simulate the source and the receivers. Device PE2 and Device PE3 are configured to statically join the 224.1.1.1 group by using the **set protocols igmp interface *interface-name* static group 224.1.1.1** command. In the case when a real multicast receiver host is not available, as in this example, this static IGMP configuration is useful. On the CE devices attached to the receivers, to make them listen to the multicast group address, the example uses **set protocols sap listen 224.1.1.1**. A ping command is used to send multicast traffic into the BGP MBPN.

Sender-based RPF is enabled on Device PE2, as follows:

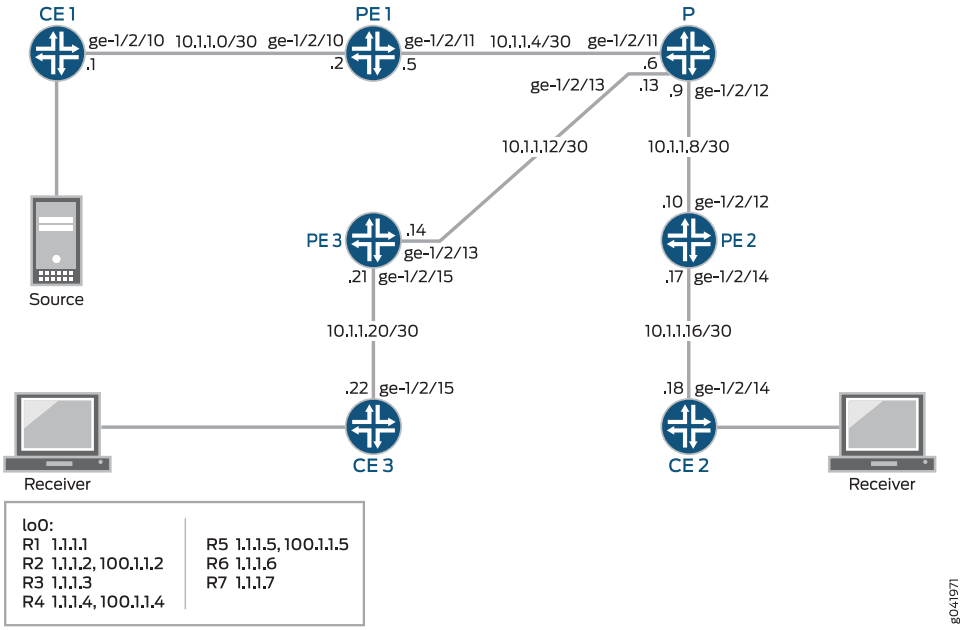
```
[routing-instances vpn-1 protocols mvpn]
user@PE2# set sender-based-rpf
```

You can optionally configure **hot-root-standby** with **sender-based-rpf**.

Topology

Figure 109 on page 717 shows the sample network.

Figure 109: Sender-Based RPF in a BGP MVPN



"Set Commands for All Devices in the Topology" on page 717 shows the configuration for all of the devices in Figure 109 on page 717.

The section "Configuring Device PE2" on page 721 describes the steps on Device PE2.

Set Commands for All Devices in the Topology

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device CE1
```
set interfaces ge-1/2/10 unit 0 family inet address 10.1.1.1/30
set interfaces ge-1/2/10 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/10.0
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.1
```

Device CE2
```
set interfaces ge-1/2/14 unit 0 family inet address 10.1.1.18/30
set interfaces ge-1/2/14 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.6/32
set protocols sap listen 224.1.1.1
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/14.0
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.6
```

Device CE3
```
set interfaces ge-1/2/15 unit 0 family inet address 10.1.1.22/30
set interfaces ge-1/2/15 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.7/32
set protocols sap listen 224.1.1.1
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/15.0
set protocols pim rp static address 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.7
```

Device P
```
set interfaces ge-1/2/11 unit 0 family inet address 10.1.1.6/30
set interfaces ge-1/2/11 unit 0 family mpls
set interfaces ge-1/2/12 unit 0 family inet address 10.1.1.9/30
set interfaces ge-1/2/12 unit 0 family mpls
set interfaces ge-1/2/13 unit 0 family inet address 10.1.1.13/30
set interfaces ge-1/2/13 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.3/32
set protocols rsvp interface all
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls interface ge-1/2/11.0
set protocols mpls interface ge-1/2/12.0
set protocols mpls interface ge-1/2/13.0
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/11.0
set protocols ospf area 0.0.0.0 interface ge-1/2/12.0
set protocols ospf area 0.0.0.0 interface ge-1/2/13.0
set protocols ldp interface ge-1/2/11.0
set protocols ldp interface ge-1/2/12.0
set protocols ldp interface ge-1/2/13.0
set protocols ldp p2mp
set routing-options router-id 1.1.1.3
```

Device PE1
```
set interfaces ge-1/2/10 unit 0 family inet address 10.1.1.2/30
```

```
set interfaces ge-1/2/10 unit 0 family mpls
set interfaces ge-1/2/11 unit 0 family inet address 10.1.1.5/30
set interfaces ge-1/2/11 unit 0 family mpls
set interfaces vt-1/2/10 unit 2 family inet
set interfaces lo0 unit 0 family inet address 1.1.1.2/32
set interfaces lo0 unit 102 family inet address 100.1.1.2/32
set protocols rsvp interface ge-1/2/11.0
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls label-switched-path p2mp-template template
set protocols mpls label-switched-path p2mp-template p2mp
set protocols mpls interface ge-1/2/11.0
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.2
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.4
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/11.0
set protocols ldp interface ge-1/2/11.0
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface ge-1/2/10.0
set routing-instances vpn-1 interface vt-1/2/10.2
set routing-instances vpn-1 interface lo0.102
set routing-instances vpn-1 provider-tunnel rsvp-te label-switched-path-template
  p2mp-template
set routing-instances vpn-1 provider-tunnel selective group 225.0.1.0/24 source 0.0.0.0/0
  rsvp-te label-switched-path-template p2mp-template
set routing-instances vpn-1 provider-tunnel selective group 225.0.1.0/24 source 0.0.0.0/0
  threshold-rate 0
set routing-instances vpn-1 vrf-target target:100:10
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.102 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/10.0
set routing-instances vpn-1 protocols pim rp local address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/10.0 mode sparse
set routing-instances vpn-1 protocols mvpn mvpn-mode rpt-spt
set routing-options router-id 1.1.1.2
set routing-options route-distinguisher-id 1.1.1.2
set routing-options autonomous-system 1001
```

Device PE2

```
set interfaces ge-1/2/12 unit 0 family inet address 10.1.1.10/30
set interfaces ge-1/2/12 unit 0 family mpls
set interfaces ge-1/2/14 unit 0 family inet address 10.1.1.17/30
set interfaces ge-1/2/14 unit 0 family mpls
set interfaces vt-1/2/10 unit 4 family inet
set interfaces lo0 unit 0 family inet address 1.1.1.4/32
set interfaces lo0 unit 104 family inet address 100.1.1.4/32
set protocols igmp interface ge-1/2/14.0 static group 224.1.1.1
set protocols rsvp interface ge-1/2/12.0
```

```
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls label-switched-path p2mp-template template
set protocols mpls label-switched-path p2mp-template p2mp
set protocols mpls interface ge-1/2/12.0
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.4
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/12.0
set protocols ldp interface ge-1/2/12.0
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/10.4
set routing-instances vpn-1 interface ge-1/2/14.0
set routing-instances vpn-1 interface lo0.104
set routing-instances vpn-1 provider-tunnel rsvp-te label-switched-path-template
    p2mp-template
set routing-instances vpn-1 provider-tunnel selective group 225.0.1.0/24 source 0.0.0.0/0
    rsvp-te label-switched-path-template p2mp-template
set routing-instances vpn-1 provider-tunnel selective group 225.0.1.0/24 source 0.0.0.0/0
    threshold-rate 0
set routing-instances vpn-1 vrf-target target:100:10
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.104 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/14.0
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/14.0 mode sparse
set routing-instances vpn-1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn-1 protocols mvpn sender-based-rpf
set routing-instances vpn-1 protocols mvpn hot-root-standby source-tree
set routing-options router-id 1.1.1.4
set routing-options route-distinguisher-id 1.1.1.4
set routing-options autonomous-system 1001
```

Device PE3

```
set interfaces ge-1/2/13 unit 0 family inet address 10.1.1.14/30
set interfaces ge-1/2/13 unit 0 family mpls
set interfaces ge-1/2/15 unit 0 family inet address 10.1.1.21/30
set interfaces ge-1/2/15 unit 0 family mpls
set interfaces vt-1/2/10 unit 5 family inet
set interfaces lo0 unit 0 family inet address 1.1.1.5/32
set interfaces lo0 unit 105 family inet address 100.1.1.5/32
set protocols igmp interface ge-1/2/15.0 static group 224.1.1.1
set protocols rsvp interface ge-1/2/13.0
set protocols mpls traffic-engineering bgp-igp-both-ribs
set protocols mpls label-switched-path p2mp-template template
set protocols mpls label-switched-path p2mp-template p2mp
set protocols mpls interface ge-1/2/13.0
set protocols bgp group ibgp type internal
```

```
set protocols bgp group ibgp local-address 1.1.1.5
set protocols bgp group ibgp family inet unicast
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.4
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/13.0
set protocols ldp interface ge-1/2/13.0
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/10.5
set routing-instances vpn-1 interface ge-1/2/15.0
set routing-instances vpn-1 interface lo0.105
set routing-instances vpn-1 provider-tunnel rsvp-te label-switched-path-template
  p2mp-template
set routing-instances vpn-1 provider-tunnel selective group 225.0.1.0/24 source 0.0.0.0/0
  rsvp-te label-switched-path-template p2mp-template
set routing-instances vpn-1 provider-tunnel selective group 225.0.1.0/24 source 0.0.0.0/0
  threshold-rate 0
set routing-instances vpn-1 vrf-target target:100:10
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.105 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/15.0
set routing-instances vpn-1 protocols pim rp static address 100.1.1.2
set routing-instances vpn-1 protocols pim interface ge-1/2/15.0 mode sparse
set routing-instances vpn-1 protocols mvpn mvpn-mode rpt-spt
set routing-options router-id 1.1.1.5
set routing-options route-distinguisher-id 1.1.1.5
set routing-options autonomous-system 1001
```

Configuring Device PE2

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Device PE2:

1. Enable enhanced IP mode.

```
[edit chassis]
user@PE2# set network-services enhanced-ip
```

2. Configure the device interfaces.

```
[edit interfaces]
user@PE2# set ge-1/2/12 unit 0 family inet address 10.1.1.10/30
user@PE2# set ge-1/2/12 unit 0 family mpls

user@PE2# set ge-1/2/14 unit 0 family inet address 10.1.1.17/30
user@PE2# set ge-1/2/14 unit 0 family mpls
```

```
user@PE2# set vt-1/2/10 unit 4 family inet

user@PE2# set lo0 unit 0 family inet address 1.1.1.4/32
user@PE2# set lo0 unit 104 family inet address 100.1.1.4/32
```

3. Configure IGMP on the interface facing the customer edge.

```
[edit protocols igmp]
user@PE2# set interface ge-1/2/14.0
```

4. (Optional) Force the PE device to join the multicast group with a static configuration.

   Normally, this would happen dynamically in a setup with real sources and receivers.

```
[edit protocols igmp]
user@PE2# set interface ge-1/2/14.0 static group 224.1.1.1
```

5. Configure RSVP on the interfaces facing the provider core.

```
[edit protocols rsvp]
user@PE2# set interface ge-1/2/0.10
```

6. Configure MPLS.

```
[edit protocols mpls]
user@PE2# set traffic-engineering bgp-igp-both-ribs
user@PE2# set label-switched-path p2mp-template template
user@PE2# set label-switched-path p2mp-template p2mp
user@PE2# set interface ge-1/2/12.0
```

7. Configure internal BGP (IBGP) among the PE routers.

```
[edit protocols bgp group ibgp]
user@PE2# set type internal
user@PE2# set local-address 1.1.1.4
user@PE2# set family inet unicast
user@PE2# set family inet-vpn any
user@PE2# set family inet-mvpn signaling
user@PE2# set neighbor 1.1.1.2
user@PE2# set neighbor 1.1.1.5
```

8. Configure an OSPF or IS-IS.

```
[edit protocols ospf]
user@PE2# set traffic-engineering
user@PE2# set area 0.0.0.0 interface lo0.0 passive
user@PE2# set area 0.0.0.0 interface ge-1/2/12.0
```

9. (Optional) Configure LDP.

   RSVP can be used instead for MPLS signaling.

```
[edit protocols bgp group ibgp]
user@PE2# set interface ge-1/2/12.0
```

```
user@PE2# set p2mp
```

10. Configure a routing policy to be used in the VPN.

    The policy is used for exporting the BGP into the PE-CE IGP session.

    ```
    [edit policy-options policy-statement parent_vpn_routes]
    user@PE2# set from protocol bgp
    user@PE2# set then accept
    ```

11. Configure the routing instance.

    ```
    [edit routing-instances vpn-1]
    user@PE2# set instance-type vrf
    user@PE2# set interface vt-1/2/10.4
    user@PE2# set interface ge-1/2/14.0
    user@PE2# set interface lo0.104
    ```

12. Configure the provider tunnel.

    ```
    [edit routing-instances vpn-1 provider-tunnel]
    user@PE2# set rsvp-te label-switched-path-template p2mp-template
    user@PE2# set selective group 225.0.1.0/24 source 0.0.0.0/0 rsvp-te
      label-switched-path-template p2mp-template
    user@PE2# set selective group 225.0.1.0/24 source 0.0.0.0/0 threshold-rate 0
    ```

13. Configure the VRF target.

    In the context of unicast IPv4 routes, choosing **vrf-target** has two implications. First, every locally learned (in this case, direct and static) route at the VRF is exported to BGP with the specified route target (RT). Also, every received inet-vpn BGP route with that RT value is imported into the VRF vpn-1. This has the advantage of a simpler configuration, and the drawback of less flexibility in selecting and modifying the exported and imported routes. It also implies that the VPN is full mesh and all the PE routers get routes from each other, so complex configurations like hub-and-spoke or extranet are not feasible. If any of these features are required, it is necessary to use **vrf-import** and **vrf-export** instead.

    ```
    [edit ]
    user@PE2# set routing-instances vpn-1 vrf-target target:100:10
    ```

14. Configure the PE-CE OSPF session.

    ```
    [edit routing-instances vpn-1 protocols ospf]
    user@PE2# set export parent_vpn_routes
    user@PE2# set area 0.0.0.0 interface lo0.104 passive
    user@PE2# set area 0.0.0.0 interface ge-1/2/14.0
    ```

15. Configure the PE-CE PIM session.

    ```
    [edit routing-instances vpn-1 protocols pim]
    user@PE2# set rp static address 100.1.1.2
    user@PE2# set interface ge-1/2/14.0 mode sparse
    ```

16. Enable the MVPN mode.

    Both **rpt-spt** and **spt-only** are supported with sender-based RPF.

    ```
    [edit routing-instances vpn-1 protocols mvpn]
    user@PE2# set mvpn-mode rpt-spt
    ```

17. Enable sender-based RPF.

    ```
    [edit routing-instances vpn-1 protocols mvpn]
    user@PE2# set sender-based-rpf
    ```

18. Configure the router ID, the router distinguisher, and the AS number.

    ```
    [edit routing-options]
    user@PE2# set router-id 1.1.1.4
    user@PE2# set route-distinguisher-id 1.1.1.4
    user@PE2# set autonomous-system 1001
    ```

Results From configuration mode, confirm your configuration by entering the **show chassis**, **show interfaces**, **show protocols**, **show policy-options**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE2# show chassis
network-services enhanced-ip;

user@PE2# show interfaces
ge-1/2/12 {
  unit 0 {
    family inet {
      address 10.1.1.10/30;
    }
    family mpls;
  }
}
ge-1/2/14 {
  unit 0 {
    family inet {
      address 10.1.1.17/30;
    }
    family mpls;
  }
}
vt-1/2/10 {
  unit 5 {
    family inet;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.5/32;
    }
```

```
    }
    unit 105 {
      family inet {
        address 100.1.1.5/32;
      }
    }
}
user@PE2# show protocols
igmp {
    interface ge-1/2/15.0 {
        static {
            group 224.1.1.1;
        }
    }
}
rsvp {
    interface all;
}
mpls {
    traffic-engineering bgp-igp-both-ribs;
    label-switched-path p2mp-template {
        template;
        p2mp;
    }
    interface ge-1/2/13.0;
}
bgp {
    group ibgp {
        type internal;
        local-address 1.1.1.5;
        family inet {
            unicast;
        }
        family inet-vpn {
            any;
        }
        family inet-mvpn {
            signaling;
        }
        neighbor 1.1.1.2;
        neighbor 1.1.1.4;
    }
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
        interface ge-1/2/13.0;
    }
}
ldp {
    interface ge-1/2/13.0;
    p2mp;
```

```
}

user@PE2# show policy-options
policy-statement parent_vpn_routes {
   from protocol bgp;
   then accept;
}

user@PE2# show routing-instances
vpn-1 {
   instance-type vrf;
   interface vt-1/2/10.5;
   interface ge-1/2/15.0;
   interface lo0.105;
   provider-tunnel {
      rsvp-te {
         label-switched-path-template {
            p2mp-template;
         }
      }
      selective {
         group 225.0.1.0/24 {
            source 0.0.0.0/0 {
               rsvp-te {
                  label-switched-path-template {
                     p2mp-template;
                  }
               }
               threshold-rate 0;
            }
         }
      }
   }
   vrf-target target:100:10;
   protocols {
      ospf {
         export parent_vpn_routes;
         area 0.0.0.0 {
            interface lo0.105 {
               passive;
            }
            interface ge-1/2/15.0;
         }
      }
      pim {
         rp {
            static {
               address 100.1.1.2;
            }
         }
         interface ge-1/2/15.0 {
            mode sparse;
         }
      }
      mvpn {
         mvpn-mode {
            rpt-spt;
```

```
        }
        sender-based-rpf;
      }
    }
  }
```

```
user@PE2# show routing-options
router-id 1.1.1.5;
route-distinguisher-id 1.1.1.5;
autonomous-system 1001;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying Sender-Based RPF on page 727
- Checking the BGP Routes on page 728
- Checking the PIM Joins on the Downstream CE Receiver Devices on page 734
- Checking the PIM Joins on the PE Devices on page 735
- Checking the Multicast Routes on page 737
- Checking the MVPN C-Multicast Routes on page 739
- Checking the Source PE on page 741

Verifying Sender-Based RPF

Purpose Make sure that sender-based RPF is enabled on Device PE2.

Action

```
user@PE2> show mvpn instance vpn-1

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : vpn-1
  MVPN Mode : RPT-SPT
  Sender-Based RPF: Enabled.
  Hot Root Standby: Disabled. Reason: Not enabled by configuration.
  Provider tunnel: I-P-tnl:RSVP-TE P2MP:1.1.1.4, 32647,1.1.1.4
  Neighbor                               Inclusive Provider Tunnel
  1.1.1.2                                RSVP-TE P2MP:1.1.1.2, 15282,1.1.1.2
  1.1.1.5                                RSVP-TE P2MP:1.1.1.5, 8895,1.1.1.5
  C-mcast IPv4 (S:G)            Provider Tunnel                          St
  0.0.0.0/0:224.1.1.1/32        RSVP-TE P2MP:1.1.1.2, 15282,1.1.1.2
  0.0.0.0/0:224.2.127.254/32    RSVP-TE P2MP:1.1.1.2, 15282,1.1.1.2

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET6

Instance : vpn-1
  MVPN Mode : RPT-SPT
  Sender-Based RPF: Enabled.
  Hot Root Standby: Disabled. Reason: Not enabled by configuration.
  Provider tunnel: I-P-tnl:RSVP-TE P2MP:1.1.1.4, 32647,1.1.1.4
```

Checking the BGP Routes

Purpose Make sure the expected BGP routes are being added to the routing tables on the PE devices.

Action `user@PE1> show route protocol bgp`

```
inet.0: 10 destinations, 14 routes (10 active, 0 holddown, 0 hidden)

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

vpn-1.inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.6/32          *[BGP/170] 1d 04:23:24, MED 1, localpref 100, from 1.1.1.4
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299792(top)
1.1.1.7/32          *[BGP/170] 1d 04:23:23, MED 1, localpref 100, from 1.1.1.5
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299776(top)
10.1.1.16/30        *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.4
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299792(top)
10.1.1.20/30        *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.5
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299776(top)
100.1.1.4/32        *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.4
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299792(top)
100.1.1.5/32        *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.5
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299776(top)

vpn-1.inet.1: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)

mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.4:32767:1.1.1.6/32
                    *[BGP/170] 1d 04:23:24, MED 1, localpref 100, from 1.1.1.4
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299792(top)
1.1.1.4:32767:10.1.1.16/30
                    *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.4
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299792(top)
1.1.1.4:32767:100.1.1.4/32
                    *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.4
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299792(top)
1.1.1.5:32767:1.1.1.7/32
                    *[BGP/170] 1d 04:23:23, MED 1, localpref 100, from 1.1.1.5
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299776(top)
1.1.1.5:32767:10.1.1.20/30
                    *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.5
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299776(top)
1.1.1.5:32767:100.1.1.5/32
                    *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.5
                       AS path: I, validation-state: unverified
                     > via ge-1/2/11.0, Push 299776, Push 299776(top)
```

```
bgp.mvpn.0: 5 destinations, 8 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.1.1.4:32767:1.1.1.4/240
                   *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
1:1.1.1.5:32767:1.1.1.5/240
                   *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
6:1.1.1.2:32767:1001:32:100.1.1.2:32:224.1.1.1/240
                   *[BGP/170] 1d 04:17:25, MED 0, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
                    [BGP/170] 1d 04:17:24, MED 0, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
6:1.1.1.2:32767:1001:32:100.1.1.2:32:224.2.127.254/240
                   *[BGP/170] 1d 04:17:25, MED 0, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
                    [BGP/170] 1d 04:17:23, MED 0, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
7:1.1.1.2:32767:1001:32:10.1.1.1:32:224.1.1.1/240
                   *[BGP/170] 20:34:47, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
                    [BGP/170] 20:34:47, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792

vpn-1.mvpn.0: 7 destinations, 13 routes (7 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.1.1.4:32767:1.1.1.4/240
                   *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
1:1.1.1.5:32767:1.1.1.5/240
                   *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
6:1.1.1.2:32767:1001:32:100.1.1.2:32:224.1.1.1/240
                    [BGP/170] 1d 04:17:25, MED 0, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
                    [BGP/170] 1d 04:17:24, MED 0, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
6:1.1.1.2:32767:1001:32:100.1.1.2:32:224.2.127.254/240
                    [BGP/170] 1d 04:17:25, MED 0, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776
                    [BGP/170] 1d 04:17:23, MED 0, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
7:1.1.1.2:32767:1001:32:10.1.1.1:32:224.1.1.1/240
                    [BGP/170] 20:34:47, localpref 100, from 1.1.1.4
```

```
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299792
                    [BGP/170] 20:34:47, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/11.0, Push 299776


user@PE2> show route protocol bgp

inet.0: 10 destinations, 14 routes (10 active, 0 holddown, 0 hidden)

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

vpn-1.inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32          *[BGP/170] 1d 04:23:24, MED 1, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299808(top)
1.1.1.7/32          *[BGP/170] 1d 04:23:20, MED 1, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299776(top)
10.1.1.0/30         *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299808(top)
10.1.1.20/30        *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299776(top)
100.1.1.2/32        *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299808(top)
100.1.1.5/32        *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299776(top)

vpn-1.inet.1: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)

mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.2:32767:1.1.1.1/32
                    *[BGP/170] 1d 04:23:24, MED 1, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299808(top)
1.1.1.2:32767:10.1.1.0/30
                    *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299808(top)
1.1.1.2:32767:100.1.1.2/32
                    *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299808(top)
1.1.1.5:32767:1.1.1.7/32
                    *[BGP/170] 1d 04:23:20, MED 1, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                     > via ge-1/2/12.0, Push 299776, Push 299776(top)
1.1.1.5:32767:10.1.1.20/30
                    *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.5
```

```
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299776, Push 299776(top)
1.1.1.5:32767:100.1.1.5/32
                   *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299776, Push 299776(top)


bgp.mvpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.1.1.2:32767:1.1.1.2/240
                   *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299808
1:1.1.1.5:32767:1.1.1.5/240
                   *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299776
5:1.1.1.2:32767:32:10.1.1.1:32:224.1.1.1/240
                   *[BGP/170] 20:34:47, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299808

vpn-1.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.1.1.2:32767:1.1.1.2/240
                   *[BGP/170] 1d 04:23:24, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299808
1:1.1.1.5:32767:1.1.1.5/240
                   *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.5
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299776
5:1.1.1.2:32767:32:10.1.1.1:32:224.1.1.1/240
                   *[BGP/170] 20:34:47, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/12.0, Push 299808
```

```
user@PE3> show route protocol bgp

inet.0: 10 destinations, 14 routes (10 active, 0 holddown, 0 hidden)

inet.3: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

vpn-1.inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32         *[BGP/170] 1d 04:23:23, MED 1, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299808(top)
1.1.1.6/32         *[BGP/170] 1d 04:23:20, MED 1, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299792(top)
10.1.1.0/30        *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299808(top)
10.1.1.16/30       *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.4
```

```
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299792(top)
100.1.1.2/32        *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299808(top)
100.1.1.4/32        *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299792(top)

vpn-1.inet.1: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)

mpls.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.2:32767:1.1.1.1/32
                    *[BGP/170] 1d 04:23:23, MED 1, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299808(top)
1.1.1.2:32767:10.1.1.0/30
                    *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299808(top)
1.1.1.2:32767:100.1.1.2/32
                    *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299808(top)
1.1.1.4:32767:1.1.1.6/32
                    *[BGP/170] 1d 04:23:20, MED 1, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299792(top)
1.1.1.4:32767:10.1.1.16/30
                    *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299792(top)
1.1.1.4:32767:100.1.1.4/32
                    *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299776, Push 299792(top)

bgp.mvpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.1.1.2:32767:1.1.1.2/240
                    *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299808
1:1.1.1.4:32767:1.1.1.4/240
                    *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299792
5:1.1.1.2:32767:32:10.1.1.1:32:224.1.1.1/240
                    *[BGP/170] 20:34:47, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                    > via ge-1/2/13.0, Push 299808

vpn-1.mvpn.0: 7 destinations, 8 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1.1.1.2:32767:1.1.1.2/240
```

```
                    *[BGP/170] 1d 04:23:23, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/13.0, Push 299808
1:1.1.1.4:32767:1.1.1.4/240
                    *[BGP/170] 1d 04:23:20, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                     > via ge-1/2/13.0, Push 299792
5:1.1.1.2:32767:32:10.1.1.1:32:224.1.1.1/240
                    *[BGP/170] 20:34:47, localpref 100, from 1.1.1.2
                      AS path: I, validation-state: unverified
                     > via ge-1/2/13.0, Push 299808
```

Checking the PIM Joins on the Downstream CE Receiver Devices

Purpose Make sure that the expected join messages are being sent.

Action

```
user@CE2> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-1/2/14.0

Group: 224.2.127.254
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-1/2/14.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
-----


user@CE3> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-1/2/15.0

Group: 224.2.127.254
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-1/2/15.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
-----
```

Meaning Both Device CE2 and Device CE3 send C-Join packets upstream to their neighboring PE routers, their unicast next-hop to reach the C-Source.

Checking the PIM Joins on the PE Devices

Purpose Make sure that the expected join messages are being sent.

Action

```
user@PE1> show pim join instance vpn-1
Instance: PIM.vpn-1 Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 224.1.1.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: ge-1/2/10.0

Group: 224.2.127.254
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream interface: Local


user@PE2> show pim join instance vpn-1
Instance: PIM.vpn-1 Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP

Group: 224.1.1.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream protocol: BGP
    Upstream interface: Through BGP

Group: 224.2.127.254
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP


user@PE3> show pim join instance vpn-1
Instance: PIM.vpn-1 Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.1.1.1
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP

Group: 224.1.1.1
    Source: 10.1.1.1
```

```
    Flags: sparse,spt
    Upstream protocol: BGP
    Upstream interface: Through BGP

Group: 224.2.127.254
    Source: *
    RP: 100.1.1.2
    Flags: sparse,rptree,wildcard
    Upstream protocol: BGP
    Upstream interface: Through BGP
```

Meaning Both Device CE2 and Device CE3 send C-Join packets upstream to their neighboring PE routers, their unicast next-hop to reach the C-Source.

The C-Join state points to BGP as the upstream interface. Actually, there is no PIM neighbor relationship between the PEs. The downstream PE converts the C-PIM (C-S, C-G) state into a Type 7 source-tree join BGP route, and sends it to the upstream PE router toward the C-Source.

Checking the Multicast Routes

Purpose Make sure that the C-Multicast flow is integrated in MVPN vpn-1 and sent by Device PE1 into the provider tunnel.

Action

```
user@PE1> show multicast route instance vpn-1
Instance: vpn-1 Family: INET

Group: 224.1.1.1/32
    Source: *
    Upstream interface: local
    Downstream interface list:
        ge-1/2/11.0

Group: 224.1.1.1
    Source: 10.1.1.1/32
    Upstream interface: ge-1/2/10.0
    Downstream interface list:
        ge-1/2/11.0

Group: 224.2.127.254/32
    Source: *
    Upstream interface: local
    Downstream interface list:
        ge-1/2/11.0


user@PE2> show multicast route instance vpn-1
Instance: vpn-1 Family: INET

Group: 224.1.1.1/32
    Source: *
    Upstream rpf interface list:
        vt-1/2/10.4 (P)
            Sender Id: Label 299840
    Downstream interface list:
        ge-1/2/14.0

Group: 224.1.1.1
    Source: 10.1.1.1/32
    Upstream rpf interface list:
        vt-1/2/10.4 (P)
            Sender Id: Label 299840

Group: 224.2.127.254/32
    Source: *
    Upstream rpf interface list:
        vt-1/2/10.4 (P)
            Sender Id: Label 299840
    Downstream interface list:
        ge-1/2/14.0


user@PE3> show multicast route instance vpn-1

Instance: vpn-1 Family: INET

Group: 224.1.1.1/32
    Source: *
    Upstream interface: vt-1/2/10.5
    Downstream interface list:
        ge-1/2/15.0

Group: 224.1.1.1
    Source: 10.1.1.1/32
    Upstream interface: vt-1/2/10.5
```

```
Group: 224.2.127.254/32
    Source: *
    Upstream interface: vt-1/2/10.5
    Downstream interface list:
        ge-1/2/15.0
```

Meaning The output shows that, unlike the other PE devices, Device PE2 is using sender-based RPF. The output on Device PE2 includes the upstream RPF sender. The Sender Id field is only shown when sender-based RPF is enabled.

Checking the MVPN C-Multicast Routes

Purpose Check the MVPN C-multicast route information,

Action

```
user@PE1> show mvpn c-multicast instance-name vpn-1

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : vpn-1
  MVPN Mode : RPT-SPT
  C-mcast IPv4 (S:G)            Provider Tunnel                        St
  0.0.0.0/0:224.1.1.1/32        RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2         RM
  10.1.1.1/32:224.1.1.1/32      RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2         RM
  0.0.0.0/0:224.2.127.254/32    RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2         RM


...


user@PE2> show mvpn c-multicast instance-name vpn-1
MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : vpn-1
  MVPN Mode : RPT-SPT
  C-mcast IPv4 (S:G)            Provider Tunnel                        St
  0.0.0.0/0:224.1.1.1/32        RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2
  10.1.1.1/32:224.1.1.1/32      RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2
  0.0.0.0/0:224.2.127.254/32    RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2

...


user@PE3> show mvpn c-multicast instance-name vpn-1

MVPN instance:
Legend for provider tunnel
S-    Selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : vpn-1
  MVPN Mode : RPT-SPT
  C-mcast IPv4 (S:G)            Provider Tunnel                        St
  0.0.0.0/0:224.1.1.1/32        RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2
  10.1.1.1/32:224.1.1.1/32      RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2
  0.0.0.0/0:224.2.127.254/32    RSVP-TE P2MP:1.1.1.2, 33314,1.1.1.2

...
```

Meaning The output shows the provider tunnel and label information.

Checking the Source PE

Purpose Check the details of the source PE,

Action `user@PE1> show mvpn c-multicast source-pe`

```
Instance : vpn-1
  MVPN Mode : RPT-SPT
  Family : INET
    C-Multicast route address :0.0.0.0/0:224.1.1.1/32
        MVPN Source-PE1:
                extended-community: no-advertise target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: lo0.102 Index: -1610691384
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: lo0.102 Index: -1610691384
    C-Multicast route address :10.1.1.1/32:224.1.1.1/32
        MVPN Source-PE1:
                extended-community: no-advertise target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: ge-1/2/10.0 Index: -1610691384
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: ge-1/2/10.0 Index: -1610691384
    C-Multicast route address :0.0.0.0/0:224.2.127.254/32
        MVPN Source-PE1:
                extended-community: no-advertise target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: lo0.102 Index: -1610691384
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: lo0.102 Index: -1610691384


user@PE2> show mvpn c-multicast source-pe
Instance : vpn-1
  MVPN Mode : RPT-SPT
  Family : INET
    C-Multicast route address :0.0.0.0/0:224.1.1.1/32
        MVPN Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
    C-Multicast route address :10.1.1.1/32:224.1.1.1/32
        MVPN Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
```



```
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
    C-Multicast route address :0.0.0.0/0:224.2.127.254/32
        MVPN Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)


user@PE3> show mvpn c-multicast source-pe

Instance : vpn-1
  MVPN Mode : RPT-SPT
  Family : INET
    C-Multicast route address :0.0.0.0/0:224.1.1.1/32
        MVPN Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
    C-Multicast route address :10.1.1.1/32:224.1.1.1/32
        MVPN Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
    C-Multicast route address :0.0.0.0/0:224.2.127.254/32
        MVPN Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)
        PIM Source-PE1:
                extended-community: target:1.1.1.2:72
                Route Distinguisher: 1.1.1.2:32767
                Autonomous system number: 1001
                Interface: (Null)

...
```

Meaning The output shows the provider tunnel and label information.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- unicast-umh-election on page 1364

Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs

This example shows how to configure redundant virtual tunnel (VT) interfaces in multiprotocol BGP (MBGP) multicast VPNs (MVPNs). To configure, include multiple VT interfaces in the routing instance and, optionally, apply the **primary** statement to one of the VT interfaces.

Requirements

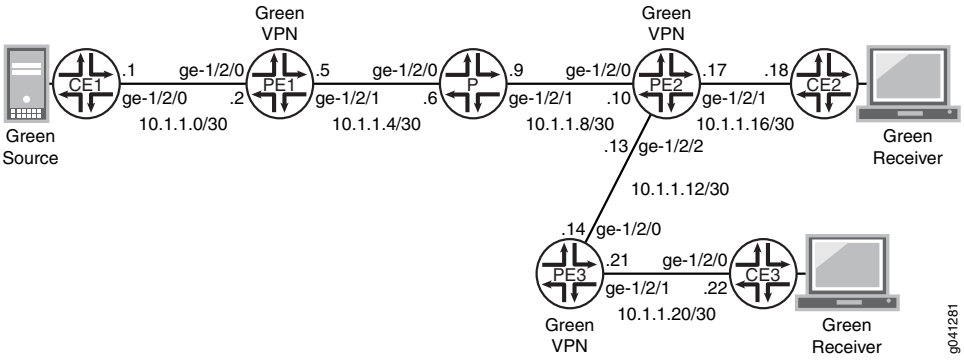
The routing device that has redundant VT interfaces configured must be running Junos OS Release 12.3 or later.

Overview

In this example, Device PE2 has redundant VT interfaces configured in a multicast LDP routing instance, and one of the VT interfaces is assigned to be the primary interface.

Figure 110 on page 744 shows the topology used in this example.

Figure 110: Multiple VT Interfaces in MBGP MVPN Topology



"CLI Quick Configuration" on page 745 shows the configuration for the customer edge (CE), provider (P), and provider edge (PE) devices in Figure 110 on page 744. The section "Step-by-Step Procedure" on page 748 describes the steps on Device PE2.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device CE1

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.1/30
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.0.2.1/24
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set protocols pim rp static address 198.51.100.0
set protocols pim interface all
set routing-options router-id 192.0.2.1
```

Device CE2

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.18/30
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.0.2.6/24
set protocols sap listen 192.168.0.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set protocols pim rp static address 198.51.100.0
set protocols pim interface all
set routing-options router-id 192.0.2.6
```

Device CE3

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.22/30
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.0.2.7/24
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set protocols pim rp static address 198.51.100.0
set protocols pim interface all
set routing-options router-id 192.0.2.7
```

Device P

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.6/30
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-1/2/1 unit 0 family inet address 10.1.1.9/30
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 192.0.2.3/24
set protocols mpls interface ge-1/2/0.0
set protocols mpls interface ge-1/2/1.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set protocols ospf area 0.0.0.0 interface ge-1/2/1.0
set protocols ldp interface ge-1/2/0.0
set protocols ldp interface ge-1/2/1.0
set protocols ldp p2mp
set routing-options router-id 192.0.2.3
```

Device PE1

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.2/30
set interfaces ge-1/2/0 unit 0 family mpls
```

```
set interfaces ge-1/2/1 unit 0 family inet address 10.1.1.5/30
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces vt-1/2/0 unit 2 family inet
set interfaces lo0 unit 0 family inet address 192.0.2.2/24
set interfaces lo0 unit 1 family inet address 198.51.100.0/24
set protocols mpls interface ge-1/2/1.0
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.0.2.2
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols bgp group ibgp neighbor 192.0.2.5
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/1.0
set protocols ldp interface ge-1/2/1.0
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface ge-1/2/0.0
set routing-instances vpn-1 interface vt-1/2/0.2 multicast
set routing-instances vpn-1 interface lo0.1
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 provider-tunnel ldp-p2mp
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.1 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set routing-instances vpn-1 protocols pim rp static address 198.51.100.0
set routing-instances vpn-1 protocols pim interface ge-1/2/0.0 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 192.0.2.2
set routing-options autonomous-system 1001
```

Device PE2

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.10/30
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-1/2/2 unit 0 family inet address 10.1.1.13/30
set interfaces ge-1/2/2 unit 0 family mpls
set interfaces ge-1/2/1 unit 0 family inet address 10.1.1.17/30
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces vt-1/1/0 unit 0 family inet
set interfaces vt-1/2/1 unit 0 family inet
set interfaces lo0 unit 0 family inet address 192.0.2.4/24
set interfaces lo0 unit 1 family inet address 203.0.113.4/24
set protocols mpls interface ge-1/2/0.0
set protocols mpls interface ge-1/2/2.0
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.0.2.4
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 192.0.2.2
set protocols bgp group ibgp neighbor 192.0.2.5
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set protocols ospf area 0.0.0.0 interface ge-1/2/2.0
```

```
set protocols ldp interface ge-1/2/0.0
set protocols ldp interface ge-1/2/2.0
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/1/0.0 multicast
set routing-instances vpn-1 interface vt-1/1/0.0 primary
set routing-instances vpn-1 interface vt-1/2/1.0 multicast
set routing-instances vpn-1 interface ge-1/2/1.0
set routing-instances vpn-1 interface lo0.1
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.1 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.0
set routing-instances vpn-1 protocols pim rp static address 198.51.100.0
set routing-instances vpn-1 protocols pim interface ge-1/2/1.0 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 192.0.2.4
set routing-options autonomous-system 1001
```

Device PE3

```
set interfaces ge-1/2/0 unit 0 family inet address 10.1.1.14/30
set interfaces ge-1/2/0 unit 0 family mpls
set interfaces ge-1/2/1 unit 0 family inet address 10.1.1.21/30
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces vt-1/2/0 unit 5 family inet
set interfaces lo0 unit 0 family inet address 192.0.2.5/24
set interfaces lo0 unit 1 family inet address 203.0.113.5/24
set protocols mpls interface ge-1/2/0.0
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.0.2.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 192.0.2.2
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
set protocols ldp interface ge-1/2/0.0
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.5 multicast
set routing-instances vpn-1 interface ge-1/2/1.0
set routing-instances vpn-1 interface lo0.1
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.1 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.0
set routing-instances vpn-1 protocols pim rp static address 198.51.100.0
set routing-instances vpn-1 protocols pim interface ge-1/2/1.0 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 192.0.2.5
```

```
set routing-options autonomous-system 1001
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure redundant VT interfaces in an MBGP MVPN:

1. Configure the physical interfaces and loopback interfaces.

```
[edit interfaces]
user@PE2# set ge-1/2/0 unit 0 family inet address 10.1.1.10/30
user@PE2# set ge-1/2/0 unit 0 family mpls

user@PE2# set ge-1/2/2 unit 0 family inet address 10.1.1.13/30
user@PE2# set ge-1/2/2 unit 0 family mpls

user@PE2# set ge-1/2/1 unit 0 family inet address 10.1.1.17/30
user@PE2# set ge-1/2/1 unit 0 family mpls

user@PE2# set lo0 unit 0 family inet address 192.0.2.4/24
user@PE2# set lo0 unit 1 family inet address 203.0.113.4/24
```

2. Configure the VT interfaces.

   Each VT interface is configurable under one routing instance.

```
[edit interfaces]
user@PE2# set vt-1/1/0 unit 0 family inet
user@PE2# set vt-1/2/1 unit 0 family inet
```

3. Configure MPLS on the physical interfaces.

```
[edit protocols mpls]
user@PE2# set interface ge-1/2/0.0
user@PE2# set interface ge-1/2/2.0
```

4. Configure BGP.

```
[edit protocols bgp group ibgp]
user@PE2# set type internal
user@PE2# set local-address 192.0.2.4
user@PE2# set family inet-vpn any
user@PE2# set family inet-mvpn signaling
user@PE2# set neighbor 192.0.2.2
user@PE2# set neighbor 192.0.2.5
```

5. Configure an interior gateway protocol.

```
[edit protocols ospf area 0.0.0.0]
user@PE2# set interface lo0.0 passive
user@PE2# set interface ge-1/2/0.0
user@PE2# set interface ge-1/2/2.0
```

6. Configure LDP.

```
[edit protocols ldp]
user@PE2# set interface ge-1/2/0.0
user@PE2# set interface ge-1/2/2.0
user@PE2# set p2mp
```

7. Configure the routing policy.

```
[edit policy-options policy-statement parent_vpn_routes]
user@PE2# set from protocol bgp
user@PE2# set then accept
```

8. Configure the routing instance.

```
[edit routing-instances vpn-1]
user@PE2# set instance-type vrf
user@PE2# set interface ge-1/2/1.0
user@PE2# set interface lo0.1
user@PE2# set route-distinguisher 100:100
user@PE2# set vrf-target target:1:1
user@PE2# set protocols ospf export parent_vpn_routes
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.1 passive
user@PE2# set protocols ospf area 0.0.0.0 interface ge-1/2/1.0
user@PE2# set protocols pim rp static address 198.51.100.0
user@PE2# set protocols pim interface ge-1/2/1.0 mode sparse
user@PE2# set protocols mvpn
```

9. Configure redundant VT interfaces in the routing instance.

   Make vt-1/1/0.0 the primary interface.

```
[edit routing-instances vpn-1]
user@PE2# set interface vt-1/1/0.0 multicast primary
user@PE2# set interface vt-1/2/1.0 multicast
```

10. Configure the router ID and autonomous system (AS) number.

```
[edit routing-options]
user@PE2# set router-id 192.0.2.4
user@PE2# set autonomous-system 1001
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@PE2# show interfaces
ge-1/2/0 {
   unit 0 {
      family inet {
         address 10.1.1.10/30;
      }
```

```
      family mpls;
    }
  }
  ge-1/2/2 {
    unit 0 {
      family inet {
        address 10.1.1.13/30;
      }
      family mpls;
    }
  }
  ge-1/2/1 {
    unit 0 {
      family inet {
        address 10.1.1.17/30;
      }
      family mpls;
    }
  }
  vt-1/1/0 {
    unit 0 {
      family inet;
    }
  }
  vt-1/2/1 {
    unit 0 {
      family inet;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 192.0.2.4/24;
      }
    }
    unit 1 {
      family inet {
        address 203.0.113.4/24;
      }
    }
  }

user@PE2# show protocols
  mpls {
    interface ge-1/2/0.0;
    interface ge-1/2/2.0;
  }
  bgp {
    group ibgp {
      type internal;
      local-address 192.0.2.4;
      family inet-vpn {
        any;
      }
      family inet-mvpn {
        signaling;
```

```
    }
    neighbor 192.0.2.2;
    neighbor 192.0.2.5;
  }
}
ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface ge-1/2/0.0;
    interface ge-1/2/2.0;
  }
}
ldp {
  interface ge-1/2/0.0;
  interface ge-1/2/2.0;
  p2mp;
}
```

```
user@PE2# show policy-options
policy-statement parent_vpn_routes {
  from protocol bgp;
  then accept;
}
```

```
user@PE2# show routing-instances
vpn-1 {
  instance-type vrf;
  interface vt-1/1/0.0 {
    multicast;
    primary;
  }
  interface vt-1/2/1.0 {
    multicast;
  }
  interface ge-1/2/1.0;
  interface lo0.1;
  route-distinguisher 100:100;
  vrf-target target:1:1;
  protocols {
    ospf {
      export parent_vpn_routes;
      area 0.0.0.0 {
        interface lo0.1 {
          passive;
        }
        interface ge-1/2/1.0;
      }
    }
    pim {
      rp {
        static {
          address 198.51.100.0;
        }
      }
      interface ge-1/2/1.0 {
```

```
            mode sparse;
          }
        }
        mvpn;
      }
    }
```

```
user@PE2# show routing-options
router-id 192.0.2.4;
autonomous-system 1001;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

NOTE: The show multicast route extensive instance *instance-name* command also displays the VT interface in the multicast forwarding table when multicast traffic is transmitted across the VPN.

Checking the LSP Route

Purpose Verify that the expected LT interface is assigned to the LDP-learned route.

Action 1. From operational mode, enter the **show route table mpls** command.

```
user@PE2> show route table mpls
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 02:09:36, metric 1
                      Receive
1                  *[MPLS/0] 02:09:36, metric 1
                      Receive
2                  *[MPLS/0] 02:09:36, metric 1
                      Receive
13                 *[MPLS/0] 02:09:36, metric 1
                      Receive
299776             *[LDP/9] 02:09:14, metric 1
                    >    via ge-1/2/0.0, Pop
299776(S=0)        *[LDP/9] 02:09:14, metric 1
                    >    via ge-1/2/0.0, Pop
299792             *[LDP/9] 02:09:09, metric 1
                    >    via ge-1/2/2.0, Pop
299792(S=0)        *[LDP/9] 02:09:09, metric 1
                    >    via ge-1/2/2.0, Pop
299808             *[LDP/9] 02:09:04, metric 1
                    >    via ge-1/2/0.0, Swap 299808
299824             *[VPN/170] 02:08:56
                    >    via ge-1/2/1.0, Pop
299840             *[VPN/170] 02:08:56
                    >    via ge-1/2/1.0, Pop
299856             *[VPN/170] 02:08:56
```

```
                    receive table vpn-1.inet.0, Pop
299872             *[LDP/9] 02:08:54, metric 1
                    >    via vt-1/1/0.0, Pop
                         via ge-1/2/2.0, Swap 299872
```

2. From configuration mode, change the primary VT interface by removing the **primary** statement from the vt-1/1/0.0 interface and adding it to the vt-1/2/1.0 interface.

```
[edit routing-instances vpn-1]
user@PE2# delete interface vt-1/1/0.0 primary
user@PE2# set interface vt-1/2/1.0 primary
user@PE2# commit
```

3. From operational mode, enter the **show route table mpls** command.

```
user@PE2> show route table mpls
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 02:09:36, metric 1
                         Receive
1                  *[MPLS/0] 02:09:36, metric 1
                         Receive
2                  *[MPLS/0] 02:09:36, metric 1
                         Receive
13                 *[MPLS/0] 02:09:36, metric 1
                         Receive
299776             *[LDP/9] 02:09:14, metric 1
                    >    via ge-1/2/0.0, Pop
299776(S=0)        *[LDP/9] 02:09:14, metric 1
                    >    via ge-1/2/0.0, Pop
299792             *[LDP/9] 02:09:09, metric 1
                    >    via ge-1/2/2.0, Pop
299792(S=0)        *[LDP/9] 02:09:09, metric 1
                    >    via ge-1/2/2.0, Pop
299808             *[LDP/9] 02:09:04, metric 1
                    >    via ge-1/2/0.0, Swap 299808
299824             *[VPN/170] 02:08:56
                    >    via ge-1/2/1.0, Pop
299840             *[VPN/170] 02:08:56
                    >    via ge-1/2/1.0, Pop
299856             *[VPN/170] 02:08:56
                      receive table vpn-1.inet.0, Pop
299872             *[LDP/9] 02:08:54, metric 1
                    >    via vt-1/2/1.0, Pop
                         via ge-1/2/2.0, Swap 299872
```

Meaning With the original configuration, the output shows the vt-1/1/0.0 interface. If you change the primary interface to vt-1/2/1.0, the output shows the vt-1/2/1.0 interface.

Related Documentation

- Understanding Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 540

Example: Configuring PIM State Limits

Controlling PIM Resources for Multicast VPNs Overview

A service provider network must protect itself from potential attacks from misconfigured or misbehaving customer edge (CE) devices and their associated VPN routing and forwarding (VRF) routing instances. Misbehaving CE devices can potentially advertise a large number of multicast routes toward a provider edge (PE) device, thereby consuming memory on the PE device and using other system resources in the network that are reserved for routes belonging to other VPNs.

To protect against potential misbehaving CE devices and VRF routing instances for specific multicast VPNs (MVPNs), you can control the following Protocol Independent Multicast (PIM) resources:

- Limit the number of accepted PIM join messages for any-source groups (*,G) and source-specific groups (S,G).

  Note how the device counts the PIM join messages:

  - Each (*,G) counts as one group toward the limit.
  - Each (S,G) counts as one group toward the limit.

- Limit the number of PIM register messages received for a specific VRF routing instance. Use this configuration if the device is configured as a rendezvous point (RP) or has the potential to become an RP. When a source in a multicast network becomes active, the source's designated router (DR) encapsulates multicast data packets into a PIM register message and sends them by means of unicast to the RP router.

  Note how the device counts PIM register messages:

  - Each unique (S,G) join received by the RP counts as one group toward the configured register messages limit.
  - Periodic register messages sent by the DR for existing or already known (S,G) entries do not count toward the configured register messages limit.
  - Register messages are accepted until either the PIM register limit or the PIM join limit (if configured) is exceeded. Once either limit isreached, any new requests are dropped.

- Limit the number of group-to-RP mappings allowed in a specific VRF routing instance. Use this configuration if the device is configured as an RP or has the potential to become an RP. This configuration can apply to devices configured for automatic RP announce and discovery (Auto-RP) or as a PIM bootstrap router. Every multicast device within a PIM domain must be able to map a particular multicast group address to the same RP. Both Auto-RP and the bootstrap router functionality are the mechanisms used to learn the set of group-to-RP mappings. Auto-RP is typically used in a PIM dense-mode deployment, and the bootstrap router is typically used in a PIM sparse-mode deployment.

NOTE: The group-to-RP mappings limit does not apply to static RP or embedded RP configurations.

Some important things to note about how the device counts group-to-RP mappings:

- One group prefix mapped to five RPs counts as five group-to-RP mappings.
- Five distinct group prefixes mapped to one RP count as five group-to-RP mappings.

Once the configured limits are reached, no new PIM join messages, PIM register messages, or group-to-RP mappings are accepted unless one of the following occurs:

- You clear the current PIM join states by using the clear pim join command. If you use this command on an RP configured for PIM register message limits, the register limit count is also restarted because the PIM join messages are unknown by the RP.

  NOTE: On the RP, you can also use the clear pim register command to clear all of the PIM registers. This command is useful if the current PIM register count is greater than the newly configured PIM register limit. After you clear the PIM registers, new PIM register messages are received up to the configured limit.

- The traffic responsible for the excess PIM join messages and PIM register messages stops and is no longer present.
- CAUTION: Never restart any of the software processes unless instructed to do so by a customer support engineer.

  You restart the PIM routing process on the device. This restart clears all of the configured limits but disrupts routing and therefore requires a maintenance window for the change.

System Log Messages for PIM Resources

You can optionally configure a system log warning threshold for each of the PIM resources. With this configuration, you can generate and review system log messages to detect if an excessive number of PIM join messages, PIM register messages, or group-to-RP mappings have been received on the device. The system log warning thresholds are configured per PIM resource and are a percentage of the configured maximum limits of the PIM join messages, PIM register messages, and group-to-RP mappings. You can further specify a log interval for each configured PIM resource, which is the amount of time (in seconds) between the log messages.

The log messages convey when the configured limits have been exceeded, when the configured warning thresholds have been exceeded, and when the configured limits drop below the configured warning threshold. Table 31 on page 756 describes the different types of PIM system messages that you might see depending on your system log warning and log interval configurations.

Table 31: PIM System Log Messages

System Log Message	Definition
RPD_PIM_SG_THRESHOLD_EXCEED	Records when the (S,G)/(*,G) routes exceed the configured warning threshold.
RPD_PIM_REG_THRESH_EXCEED	Records when the PIM registers exceed the configured warning threshold.
RPD_PIM_GRP_RP_MAP_THRES_EXCEED	Records when the group-to-RP mappings exceed the configured warning threshold.
RPD_PIM_SG_LIMIT_EXCEED	Records when the (S,G)/(*,G) routes exceed the configured limit, or when the configured log interval has been met and the routes exceed the configured limit.
RPD_PIM_REGISTER_LIMIT_EXCEED	Records when the PIM registers exceed the configured limit, or when the configured log interval has been met and the registers exceed the configured limit.
RPD_PIM_GRP_RP_MAP_LIMIT_EXCEED	Records when the group-to-RP mappings exceed the configured limit, or when the configured log interval has been met and the mapping exceeds the configured limit.
RPD_PIM_SG_LIMIT_BELOW	Records when the (S,G)/(*,G) routes drop below the configured limit and the configured log interval.
RPD_PIM_REGISTER_LIMIT_BELOW	Records when the PIM registers drop below the configured limit and the configured log interval.
RPD_PIM_GRP_RP_MAP_LIMIT_BELOW	Records when the group-to-RP mappings drop below the configured limit and the configured log interval.

See Also
- Example: Configuring PIM State Limits on page 756

Example: Configuring PIM State Limits

This example shows how to set limits on the Protocol Independent Multicast (PIM) state information so that a service provider network can protect itself from potential attacks from misconfigured or misbehaving customer edge (CE) devices and their associated VPN routing and forwarding (VRF) routing instances.

Requirements

No special configuration beyond device initialization is required before configuring this example.

Overview

In this example, a multiprotocol BGP-based multicast VPN (next-generation MBGP MVPN) is configured with limits on the PIM state resources.

The **sglimit maximum** statement sets a limit for the number of accepted (*,G) and (S,G) PIM join states received for the vpn-1 routing instance.

The **rp register-limit maximum** statement configures a limit for the number of PIM register messages received for the vpn-1 routing instance. You configure this statement on the rendezvuos point (RP) or on all the devices that might become the RP.

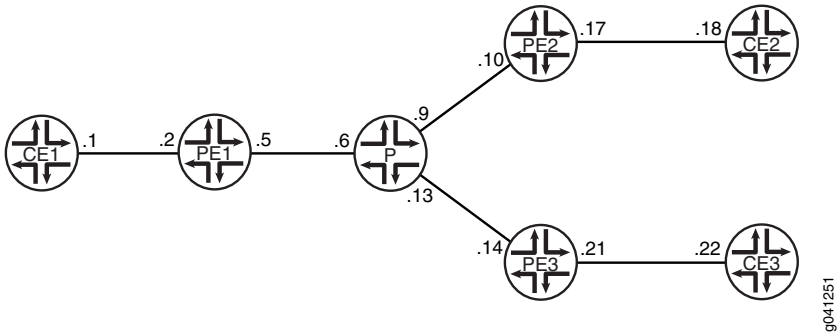
The **group-rp-mapping maximum** statement configures a limit for the number of group-to-RP mappings allowed in the vpn-1 routing instance.

For each configured PIM resource, the **threshold** statement sets a percentage of the maximum limit at which to start generating warning messages in the PIM log file.

For each configured PIM resource, the **log-interval** statement is an amount of time (in seconds) between system log message generation.

Figure 111 on page 757 shows the topology used in this example.

Figure 111: PIM State Limits Topology



"CLI Quick Configuration" on page 757 shows the configuration for all of the devices in Figure 111 on page 757. The section "Step-by-Step Procedure" on page 761 describes the steps on Device PE1.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device CE1

```
set interfaces ge-1/2/0 unit 1 family inet address 10.1.1.1/30
set interfaces ge-1/2/0 unit 1 family mpls
set interfaces lo0 unit 1 family inet address 192.0.2.1/24
set protocols ospf area 0.0.0.0 interface lo0.1 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.1
```

```
set protocols pim rp static address 203.0.113.1
set protocols pim interface all
set routing-options router-id 192.0.2.1
```

Device PE1

```
set interfaces ge-1/2/0 unit 2 family inet address 10.1.1.2/30
set interfaces ge-1/2/0 unit 2 family mpls
set interfaces ge-1/2/1 unit 5 family inet address 10.1.1.5/30
set interfaces ge-1/2/1 unit 5 family mpls
set interfaces vt-1/2/0 unit 2 family inet
set interfaces lo0 unit 2 family inet address 192.0.2.2/24
set interfaces lo0 unit 102 family inet address 203.0.113.1/24
set protocols mpls interface ge-1/2/1.5
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.0.2.2
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols bgp group ibgp neighbor 192.0.2.5
set protocols ospf area 0.0.0.0 interface lo0.2 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/1.5
set protocols ldp interface ge-1/2/1.5
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface ge-1/2/0.2
set routing-instances vpn-1 interface vt-1/2/0.2
set routing-instances vpn-1 interface lo0.102
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 provider-tunnel ldp-p2mp
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.102 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/0.2
set routing-instances vpn-1 protocols pim sglimit family inet maximum 100
set routing-instances vpn-1 protocols pim sglimit family inet threshold 70
set routing-instances vpn-1 protocols pim sglimit family inet log-interval 10
set routing-instances vpn-1 protocols pim rp register-limit family inet maximum 100
set routing-instances vpn-1 protocols pim rp register-limit family inet threshold 80
set routing-instances vpn-1 protocols pim rp register-limit family inet log-interval 10
set routing-instances vpn-1 protocols pim rp group-rp-mapping family inet maximum 100
set routing-instances vpn-1 protocols pim rp group-rp-mapping family inet threshold 80
set routing-instances vpn-1 protocols pim rp group-rp-mapping family inet log-interval
  10
set routing-instances vpn-1 protocols pim rp static address 203.0.113.1
set routing-instances vpn-1 protocols pim interface ge-1/2/0.2 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 192.0.2.2
set routing-options autonomous-system 1001
```

Device P

```
set interfaces ge-1/2/0 unit 6 family inet address 10.1.1.6/30
set interfaces ge-1/2/0 unit 6 family mpls
set interfaces ge-1/2/1 unit 9 family inet address 10.1.1.9/30
set interfaces ge-1/2/1 unit 9 family mpls
```

```
set interfaces ge-1/2/2 unit 13 family inet address 10.1.1.13/30
set interfaces ge-1/2/2 unit 13 family mpls
set interfaces lo0 unit 3 family inet address 192.0.2.3/24
set protocols mpls interface ge-1/2/0.6
set protocols mpls interface ge-1/2/1.9
set protocols mpls interface ge-1/2/2.13
set protocols ospf area 0.0.0.0 interface lo0.3 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.6
set protocols ospf area 0.0.0.0 interface ge-1/2/1.9
set protocols ospf area 0.0.0.0 interface ge-1/2/2.13
set protocols ldp interface ge-1/2/0.6
set protocols ldp interface ge-1/2/1.9
set protocols ldp interface ge-1/2/2.13
set protocols ldp p2mp
set routing-options router-id 192.0.2.3
```

Device PE2

```
set interfaces ge-1/2/0 unit 10 family inet address 10.1.1.10/30
set interfaces ge-1/2/0 unit 10 family mpls
set interfaces ge-1/2/1 unit 17 family inet address 10.1.1.17/30
set interfaces ge-1/2/1 unit 17 family mpls
set interfaces vt-1/2/0 unit 4 family inet
set interfaces lo0 unit 4 family inet address 192.0.2.4/24
set interfaces lo0 unit 104 family inet address 203.0.113.4/24
set protocols mpls interface ge-1/2/0.10
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.0.2.4
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 192.0.2.2
set protocols bgp group ibgp neighbor 192.0.2.5
set protocols ospf area 0.0.0.0 interface lo0.4 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.10
set protocols ldp interface ge-1/2/0.10
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.4
set routing-instances vpn-1 interface ge-1/2/1.17
set routing-instances vpn-1 interface lo0.104
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.104 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.17
set routing-instances vpn-1 protocols pim rp group-rp-mapping family inet maximum 100
set routing-instances vpn-1 protocols pim rp group-rp-mapping family inet threshold 80
set routing-instances vpn-1 protocols pim rp group-rp-mapping family inet log-interval
  10
set routing-instances vpn-1 protocols pim rp static address 203.0.113.1
set routing-instances vpn-1 protocols pim interface ge-1/2/1.17 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 192.0.2.4
set routing-options autonomous-system 1001
```

Device PE3

```
set interfaces ge-1/2/0 unit 14 family inet address 10.1.1.14/30
set interfaces ge-1/2/0 unit 14 family mpls
set interfaces ge-1/2/1 unit 21 family inet address 10.1.1.21/30
set interfaces ge-1/2/1 unit 21 family mpls
set interfaces vt-1/2/0 unit 5 family inet
set interfaces lo0 unit 5 family inet address 192.0.2.5/24
set interfaces lo0 unit 105 family inet address 203.0.113.5/24
set protocols mpls interface ge-1/2/0.14
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 192.0.2.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 192.0.2.2
set protocols bgp group ibgp neighbor 192.0.2.4
set protocols ospf area 0.0.0.0 interface lo0.5 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.14
set protocols ldp interface ge-1/2/0.14
set protocols ldp p2mp
set policy-options policy-statement parent_vpn_routes from protocol bgp
set policy-options policy-statement parent_vpn_routes then accept
set routing-instances vpn-1 instance-type vrf
set routing-instances vpn-1 interface vt-1/2/0.5
set routing-instances vpn-1 interface ge-1/2/1.21
set routing-instances vpn-1 interface lo0.105
set routing-instances vpn-1 route-distinguisher 100:100
set routing-instances vpn-1 vrf-target target:1:1
set routing-instances vpn-1 protocols ospf export parent_vpn_routes
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface lo0.105 passive
set routing-instances vpn-1 protocols ospf area 0.0.0.0 interface ge-1/2/1.21
set routing-instances vpn-1 protocols pim rp static address 203.0.113.1
set routing-instances vpn-1 protocols pim interface ge-1/2/1.21 mode sparse
set routing-instances vpn-1 protocols mvpn
set routing-options router-id 192.0.2.5
set routing-options autonomous-system 1001
```

Device CE2

```
set interfaces ge-1/2/0 unit 18 family inet address 10.1.1.18/30
set interfaces ge-1/2/0 unit 18 family mpls
set interfaces lo0 unit 6 family inet address 192.0.2.6/24
set protocols sap listen 192.168.0.0
set protocols ospf area 0.0.0.0 interface lo0.6 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.18
set protocols pim rp static address 203.0.113.1
set protocols pim interface all
set routing-options router-id 192.0.2.6
```

Device CE3

```
set interfaces ge-1/2/0 unit 22 family inet address 10.1.1.22/30
set interfaces ge-1/2/0 unit 22 family mpls
set interfaces lo0 unit 7 family inet address 192.0.2.7/24
set protocols ospf area 0.0.0.0 interface lo0.7 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.22
set protocols pim rp static address 203.0.113.1
set protocols pim interface all
set routing-options router-id 192.0.2.7
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure PIM state limits:

1. Configure the network interfaces.

```
[edit interfaces]
user@PE1# set ge-1/2/0 unit 2 family inet address 10.1.1.2/30
user@PE1# set ge-1/2/0 unit 2 family mpls

user@PE1# set ge-1/2/1 unit 5 family inet address 10.1.1.5/30
user@PE1# set ge-1/2/1 unit 5 family mpls

user@PE1# set vt-1/2/0 unit 2 family inet

user@PE1# set lo0 unit 2 family inet address 192.0.2.2/24
user@PE1# set lo0 unit 102 family inet address 203.0.113.1/24
```

2. Configure MPLS on the core-facing interface.

```
[edit protocols mpls]
user@PE1# set interface ge-1/2/1.5
```

3. Configure internal BGP (IBGP) on the main router.

   The IBGP neighbors are the other PE devices.

```
[edit protocols bgp group ibgp]
user@PE1# set type internal
user@PE1# set local-address 192.0.2.2
user@PE1# set family inet-vpn any
user@PE1# set family inet-mvpn signaling
user@PE1# set neighbor 192.0.2.4
user@PE1# set neighbor 192.0.2.5
```

4. Configure OSPF on the main router.

```
[edit protocols ospf area 0.0.0.0]
user@PE1# set interface lo0.2 passive
user@PE1# set interface ge-1/2/1.5
```

5. Configure a signaling protocol (RSVP or LDP) on the main router.

```
[edit protocols ldp]
user@PE1# set interface ge-1/2/1.5
user@PE1# set p2mp
```

6. Configure the BGP export policy.

```
[edit policy-options policy-statement parent_vpn_routes]
user@PE1# set from protocol bgp
user@PE1# set then accept
```

7. Configure the routing instance.

   The customer-facing interfaces and the BGP export policy are referenced in the routing instance.

```
[edit routing-instances vpn-1]
user@PE1# set instance-type vrf

user@PE1# set interface ge-1/2/0.2
user@PE1# set interface vt-1/2/0.2
user@PE1# set interface lo0.102

user@PE1# set route-distinguisher 100:100
user@PE1# set provider-tunnel ldp-p2mp
user@PE1# set vrf-target target:1:1

user@PE1# set protocols ospf export parent_vpn_routes
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.102 passive
user@PE1# set protocols ospf area 0.0.0.0 interface ge-1/2/0.2

user@PE1# set protocols pim rp static address 203.0.113.1
user@PE1# set protocols pim interface ge-1/2/0.2 mode sparse

user@PE1# set protocols mvpn
```

8. Configure the PIM state limits.

```
[edit routing-instances vpn-1 protocols pim]
user@PE1# set sglimit family inet maximum 100
user@PE1# set sglimit family inet threshold 70
user@PE1# set sglimit family inet log-interval 10

user@PE1# set rp register-limit family inet maximum 100
user@PE1# set rp register-limit family inet threshold 80
user@PE1# set rp register-limit family inet log-interval 10

user@PE1# set rp group-rp-mapping family inet maximum 100
user@PE1# set rp group-rp-mapping family inet threshold 80
user@PE1# set rp group-rp-mapping family inet log-interval 10
```

9. Configure the router ID and AS number.

```
[edit routing-options]
user@PE1# set router-id 192.0.2.2
user@PE1# set autonomous-system 1001
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, **show routing-instances**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
user@PE1# show interfaces
ge-1/2/0 {
    unit 2 {
        family inet {
            address 10.1.1.2/30;
        }
        family mpls;
    }
}
ge-1/2/1 {
    unit 5 {
        family inet {
            address 10.1.1.5/30;
        }
        family mpls;
    }
}
vt-1/2/0 {
    unit 2 {
        family inet;
    }
}
lo0 {
    unit 2 {
        family inet {
            address 192.0.2.2/24;
        }
    }
    unit 102 {
        family inet {
            address 203.0.113.1/24;
        }
    }
}

user@PE1# show protocols
mpls {
    interface ge-1/2/1.5;
}
bgp {
    group ibgp {
        type internal;
        local-address 192.0.2.2;
        family inet-vpn {
            any;
        }
        family inet-mvpn {
            signaling;
        }
        neighbor 192.0.2.4;
        neighbor 192.0.2.5;
    }
}
ospf {
    area 0.0.0.0 {
        interface lo0.2 {
```

```
        passive;
      }
      interface ge-1/2/1.5;
    }
}
ldp {
    interface ge-1/2/1.5;
    p2mp;
}
```

```
user@PE1# show policy-options
policy-statement parent_vpn_routes {
    from protocol bgp;
    then accept;
}
```

```
user@PE1# show routing-instances
vpn-1 {
    instance-type vrf;
    interface ge-1/2/0.2;
    interface vt-1/2/0.2;
    interface lo0.102;
    route-distinguisher 100:100;
    provider-tunnel {
        ldp-p2mp;
    }
    vrf-target target:1:1;
    protocols {
        ospf {
            export parent_vpn_routes;
            area 0.0.0.0 {
                interface lo0.102 {
                    passive;
                }
                interface ge-1/2/0.2;
            }
        }
        pim {
            sglimit {
                family inet {
                    maximum 100;
                    threshold 70;
                    log-interval 10;
                }
            }
            rp {
                register-limit {
                    family inet {
                        maximum 100;
                        threshold 80;
                        log-interval 10;
                    }
                }
                group-rp-mapping {
                    family inet {
                        maximum 100;
                        threshold 80;
```

```
            log-interval 10;
          }
        }
        static {
          address 203.0.113.1;
        }
      }
      interface ge-1/2/0.2 {
        mode sparse;
      }
    }
    mvpn;
  }
}
```

```
user@PE1# show routing-options
router-id 192.0.2.2;
autonomous-system 1001;
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Monitoring the PIM State Information

Purpose Verify that the counters are set as expected and are not exceeding the configured limits.

Action From operational mode, enter the **show pim statistics** command.

```
user@PE1> show pim statistics instance vpn-1
PIM Message type        Received       Sent  Rx errors
V2 Hello                     393        390          0
...
V4 (S,G) Maximum                            100
V4 (S,G) Accepted                             0
V4 (S,G) Threshold                           70
V4 (S,G) Log Interval                        10
V4 (grp-prefix, RP) Maximum                 100
V4 (grp-prefix, RP) Accepted                  0
V4 (grp-prefix, RP) Threshold                80
V4 (grp-prefix, RP) Log Interval             10
V4 Register Maximum                         100
V4 Register Accepted                          0
V4 Register Threshold                        80
V4 Register Log Interval                     10
```

Meaning The V4 (S,G) Maximum field shows the maximum number of (S,G) IPv4 multicast routes accepted for the VPN routing instance. If this number is met, additional (S,G) entries are not accepted.

The V4 (S,G) Accepted field shows the number of accepted (S,G) IPv4 multicast routes.

The V4 (S,G) Threshold field shows the threshold at which a warning message is logged (percentage of the maximum number of (S,G) IPv4 multicast routes accepted by the device).

The V4 (S,G) Log Interval field shows the time (in seconds) between consecutive log messages.

The V4 (grp-prefix, RP) Maximum field shows the maximum number of group-to-rendezvous point (RP) IPv4 multicast mappings accepted for the VRF routing instance. If this number is met, additional mappings are not accepted.

The V4 (grp-prefix, RP) Accepted field shows the number of accepted group-to-RP IPv4 multicast mappings.

The V4 (grp-prefix, RP) Threshold field shows the threshold at which a warning message is logged (percentage of the maximum number of group-to-RP IPv4 multicast mappings accepted by the device).

The V4 (grp-prefix, RP) Log Interval field shows the time (in seconds) between consecutive log messages.

The V4 Register Maximum field shows the maximum number of IPv4 PIM registers accepted for the VRF routing instance. If this number is met, additional PIM registers are not accepted. You configure the register limits on the RP.

The V4 Register Accepted field shows the number of accepted IPv4 PIM registers.

The V4 Register Threshold field shows the threshold at which a warning message is logged (percentage of the maximum number of IPv4 PIM registers accepted by the device).

The V4 Register Log Interval field shows the time (in seconds) between consecutive log messages.

See Also

- Controlling PIM Resources for Multicast VPNs Overview on page 754

Related Documentation

- Limiting the Number of IGMP Multicast Group Joins on Logical Interfaces on page 45
- Examples: Configuring the Multicast Forwarding Cache on page 928
- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

CHAPTER 22

Configuring PIM Join Load Balancing

Use Case for PIM Join Load Balancing

Large-scale service providers often have to meet the dynamic requirements of rapidly growing, worldwide virtual private network (VPN) markets. Service providers use the VPN infrastructure to deliver sophisticated services, such as video and voice conferencing, over highly secure, resilient networks. These services are usually loss-sensitive or delay-sensitive, and their data packets need to be delivered over a large-scale IP network in real time. The use of IP Multicast bandwidth-conserving technology has enabled service providers to exceed the most stringent service-level agreements (SLAs) and resiliency requirements.

IP multicast enables service providers to optimize network utilization while offering new revenue-generating value-added services, such as voice, video, and collaboration-based applications. IP multicast applications are becoming increasingly popular among enterprises, and as new applications start using multicast to deploy high-bandwidth and mission-critical services, it raises a new set of challenges for deploying IP multicast in the network.

IP multicast applications act as an essential communication protocol to effectively manage bandwidth and to reduce application server load by replicating the traffic on the network when the need arises. IP Protocol Independent Multicast (PIM) is the most important IP multicast routing protocol that is used to communicate between the multicast routers, and is the industry standard for building multicast distribution trees of receiving hosts. The multipath PIM join load-balancing feature in a multicast VPN provides bandwidth efficiency by utilizing unequal paths toward a destination, improves scalability for large service providers, and minimizes service disruption.

The large-scale demands of service providers for IP access require Layer 3 VPN composite next hops along with external and internal BGP (EIBGP) VPN load balancing. The

multipath PIM join load-balancing feature meets the large-scale requirements of enterprises by enabling **l3vpn-composite-nh** to be turned on along with EIBGP load balancing.

When the service provider network does not have the multipath PIM join load-balancing feature enabled on the provider edge (PE) routers, a hash-based algorithm is used to determine the best route to transmit multicast datagrams throughout the network. With hash-based join load balancing, adding new PE routers to the candidate upstream toward the destination results in PIM join messages being redistributed to new upstream paths. If the number of join messages is large, network performance is impacted because join messages are being sent to the new reverse path forwarding (RPF) neighbor and prune messages are being sent to the old RPF neighbor. In next-generation multicast virtual private network (MVPN), this results in multicast data messages being withdrawn from old upstream paths and advertised on new upstream paths, impacting network performance.

Related Documentation

- PIM Join Load Balancing on Multipath MVPN Routes Overview on page 768
- Example: Configuring PIM Join Load Balancing on Draft-Rosen Multicast VPN on page 772
- Example: Configuring PIM Join Load Balancing on Next-Generation Multicast VPN on page 781

PIM Join Load Balancing on Multipath MVPN Routes Overview

A multicast virtual private network (MVPN) is a technology to deploy the multicast service in an existing MPLS/BGP VPN.

The two main MVPN services are:

- Dual PIM MVPNs (also referred to as Draft-Rosen)
- Multiprotocol BGP-based MVPNs (also referred to as next-generation)

Next-generation MVPNs constitute the next evolution after the Draft-Rosen MVPN and provide a simpler solution for administrators who want to configure multicast over Layer 3 VPNs. A Draft-Rosen MVPN uses Protocol Independent Multicast (PIM) for customer multicast (C-multicast) signaling, and a next-generation MVPN uses BGP for C-multicast signaling.

Multipath routing in an MVPN is applied to make data forwarding more robust against network failures and to minimize shared backup capacities when resilience against network failures is required.

By default, PIM join messages are sent toward a source based on the reverse path forwarding (RPF) routing table check. If there is more than one equal-cost path toward the source [S, G] or rendezvous point (RP) [*, G], then one upstream interface is used to send the join messages. The upstream path can be:

- A single active external BGP (EBGP) path when both EBGP and internal BGP (IBGP) paths are present.

- A single active IBGP path when there is no EBGP path present.

With the introduction of the multipath PIM join load-balancing feature, customer PIM (C-PIM) join messages are load-balanced in the following ways:

- In the case of a Draft-Rosen MVPN, unequal EBGP and IBGP paths are utilized.
- In the case of next-generation MVPN:
  - Available IBGP paths are utilized when no EBGP path is present.
  - Available EBGP paths are utilized when both EBGP and IBGP paths are present.

This feature is applicable to IPv4 C-PIM join messages over the Layer 3 MVPN service.

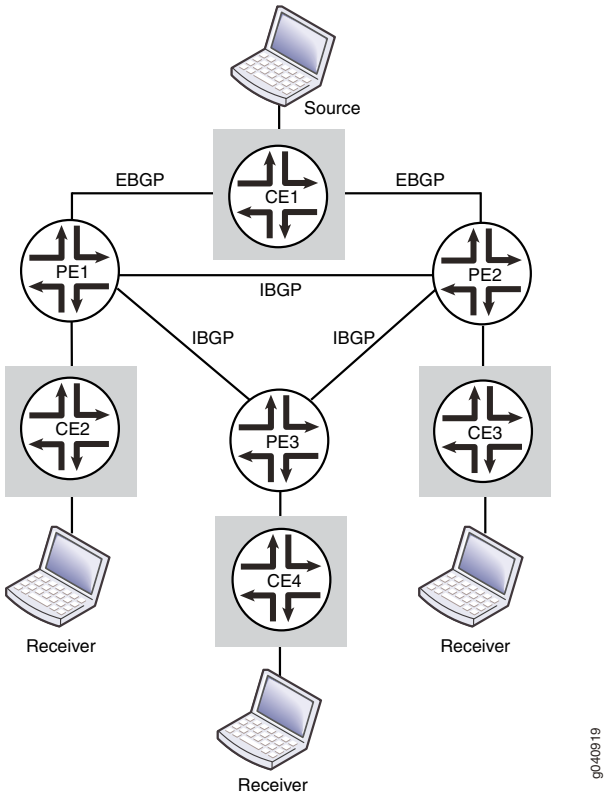
By default, a customer source (C-S) or a customer RP (C-RP) is considered remote if the active **rt_entry** is a secondary route and the primary route is present in a different routing instance. Such determination is being done without taking into consideration the (C-*,G) or (C-S,G) state for which the check is being performed. The multipath PIM join load-balancing feature determines if a source (or RP) is remote by taking into account the associated (C-*,G) or (C-S,G) state.

When the provider network does not have provider edge (PE) routers with the multipath PIM join load-balancing feature enabled, hash-based join load balancing is used. Although the decision to configure this feature does not impact PIM or overall system performance, network performance can be affected temporarily, if the feature is not enabled.

With hash-based join load balancing, adding new PE routers to the candidate upstream toward the C-S or C-RP results in C-PIM join messages being redistributed to new upstream paths. If the number of join messages is large, network performance is impacted because of join messages being sent to the new RPF neighbor and prune messages being sent to the old RPF neighbor. In next-generation MVPN, this results in BGP C-multicast data messages being withdrawn from old upstream paths and advertised on new upstream paths, impacting network performance.

In Figure 112 on page 770, PE1 and PE2 are the upstream PE routers. Router PE1 learns route Source from EBGP and IBGP peers—the customer edge CE1 router and the PE2 router, respectively.

Figure 112: PIM Join Load Balancing



- If the PE routers run the Draft-Rosen MVPN, the PE1 router distributes C-PIM join messages between the EBGP path to the CE1 router and the IBGP path to the PE2 router. The join messages on the IBGP path are sent over a multicast tunnel interface through which the PE routers establish C-PIM adjacency with each other.

  If a PE router loses one or all EBGP paths toward the source (or RP), the C-PIM join messages that were previously using the EBGP path are moved to a multicast tunnel interface, and the RPF neighbor on the multicast tunnel interface is selected based on a hash mechanism.

  On discovering the first EBGP path toward the source (or RP), only new join messages get load-balanced across EBGP and IBGP paths, whereas the existing join messages on the multicast tunnel interface remain unaffected.

- If the PE routers run the next-generation MVPN, the PE1 router sends C-PIM join messages directly to the CE1 router over the EBGP path. There is no C-PIM adjacency between the PE1 and PE2 routers. Router PE3 distributes the C-PIM join messages between the two IBGP paths to PE1 and PE2. The Bytewise-XOR hash algorithm is used to send the C-multicast data according to Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp, *BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs*.

Because the multipath PIM join load-balancing feature in a Draft-Rosen MVPN utilizes unequal EBGP and IBGP paths to the destination, loops can be created when forwarding unicast packets to the destination. To avoid or break such loops:

- Traffic arriving from a core or master instance should not be forwarded back to the core facing interfaces.
- A single multicast tunnel interface should either be selected as the upstream interface or the downstream interface.
- An upstream or downstream multicast tunnel interface should point to a non-multicast tunnel interface.

As a result of the loop avoidance mechanism, join messages arriving from an EBGP path get load-balanced across EIBGP paths as expected, whereas join messages from an IBGP path are constrained to choose the EBGP path only.

In Figure 112 on page 770, if the CE2 host sends unicast data traffic to the CE1 host, the PE1 router could send the multicast flow to the PE2 router over the MPLS core due to traffic load balancing. A data forwarding loop is prevented by ensuring that PE2 does not forward traffic back on the MPLS core because of the load-balancing algorithm.

In the case of C-PIM join messages, assuming that both the CE2 host and the CE3 host are interested in receiving traffic from the source (S, G), and if both PE1 and PE2 choose each other as the RPF neighbor toward the source, then a multicast tree cannot be formed completely. This feature implements mechanisms to prevent such join loops in the multicast control plane in a Draft-Rosen MVPN scenario.

NOTE:

Disruption of multicast traffic or creation of join loops can occur, resulting in a multicast distribution tree (MDT) not being formed properly due to one of the following reasons:

- During a graceful Routing Engine switchover (GRES), the EIBGP path selection for C-PIM join messages can vary, because the upstream interface selection is performed again for the new Routing Engine based on the join messages it receives from the CE and PE neighbors. This can lead to disruption of multicast traffic depending on the number of join messages received and the load on the network at the time of the graceful restart. However, nonstop active routing (NSR) is not supported and has no impact on the multicast traffic in a Draft-Rosen MVPN scenario.
- Any PE router in the provider network is running another vendor's implementation that does not apply the same hashing algorithm implemented in this feature.
- The multipath PIM join load-balancing feature has not been configured properly.

Related Documentation

- Example: Configuring PIM Join Load Balancing on Next-Generation Multicast VPN on page 781

Example: Configuring PIM Join Load Balancing on Draft-Rosen Multicast VPN

This example shows how to configure multipath routing for external and internal virtual private network (VPN) routes with unequal interior gateway protocol (IGP) metrics, and Protocol Independent Multicast (PIM) join load balancing on provider edge (PE) routers running Draft-Rosen multicast VPN (MVPN). This feature allows customer PIM (C-PIM) join messages to be load-balanced across external and internal BGP (EIBGP) upstream paths when the PE router has both external BGP (EBGP) and internal BGP (IBGP) paths toward the source or rendezvous point (RP).

Requirements

This example requires the following hardware and software components:

- Three routers that can be a combination of M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers.
- Junos OS Release 12.1 or later running on all the devices.

Before you begin:

1. Configure the device interfaces.
2. Configure the following routing protocols on all PE routers:
   - OSPF
   - MPLS
   - LDP
   - PIM
   - BGP
3. Configure a multicast VPN.

Overview and Topology

Junos OS Release 12.1 and later support multipath configuration along with PIM join load balancing. This allows C-PIM join messages to be load-balanced across unequal EIBGP routes, if a PE router has EBGP and IBGP paths toward the source (or RP). In previous releases, only the active EBGP path was used to send the join messages. This feature is applicable to IPv4 C-PIM join messages.

During load balancing, if a PE router loses one or more EBGP paths toward the source (or RP), the C-PIM join messages that were previously using the EBGP path are moved to a multicast tunnel interface, and the reverse path forwarding (RPF) neighbor on the multicast tunnel interface is selected based on a hash mechanism.

On discovering the first EBGP path toward the source (or RP), only the new join messages get load-balanced across EIBGP paths, whereas the existing join messages on the multicast tunnel interface remain unaffected.

Though the primary goal for multipath PIM join load balancing is to utilize unequal EIBGP paths for multicast traffic, potential join loops can be avoided if a PE router chooses only the EBGP path when there are one or more join messages for different groups from a remote PE router. If the remote PE router's join message arrives after the PE router has already chosen IBGP as the upstream path, then the potential loops can be broken by changing the selected upstream path to EBGP.

NOTE: During a graceful Routing Engine switchover (GRES), the EIBGP path selection for C-PIM join messages can vary, because the upstream interface selection is performed again for the new Routing Engine based on the join messages it receives from the CE and PE neighbors. This can lead to disruption of multicast traffic depending on the number of join messages received and the load on the network at the time of the graceful restart. However, the nonstop active routing feature is not supported and has no impact on the multicast traffic in a Draft-Rosen MVPN scenario.

In this example, PE1 and PE2 are the upstream PE routers for which the multipath PIM join load-balancing feature is configured. Routers PE1 and PE2 have one EBGP path and one IBGP path each toward the source. The Source and Receiver attached to customer edge (CE) routers are Free BSD hosts.

On PE routers that have EIBGP paths toward the source (or RP), such as PE1 and PE2, PIM join load balancing is performed as follows:

1. The existing join-count-based load balancing is performed such that the algorithm first selects the least loaded C-PIM interface. If there is equal or no load on all the C-PIM interfaces, the join messages get distributed equally across the available upstream interfaces.

   In Figure 113 on page 776, if the PE1 router receives PIM join messages from the CE2 router, and if there is equal or no load on both the EBGP and IBGP paths toward the source, the join messages get load-balanced on the EIBGP paths.

2. If the selected least loaded interface is a multicast tunnel interface, then there can be a potential join loop if the downstream list of the customer join (C-join) message already contains the multicast tunnel interface. In such a case, the least loaded interface among EBGP paths is selected as the upstream interface for the C-join message.

Assuming that the IBGP path is the least loaded, the PE1 router sends the join messages to PE2 using the IBGP path. If PIM join messages from the PE3 router arrive on PE1, then the downstream list of the C-join messages for PE3 already contains a multicast tunnel interface, which can lead to a potential join loop, because both the upstream and downstream interfaces are multicast tunnel interfaces. In this case, PE1 uses only the EBGP path to send the join messages.

3. If the selected least loaded interface is a multicast tunnel interface and the multicast tunnel interface is not present in the downstream list of the C-join messages, the loop prevention mechanism is not necessary. If any PE router has already advertised data multicast distribution tree (MDT) type, length, and values (TLVs), that PE router is selected as the upstream neighbor.

   When the PE1 router sends the join messages to PE2 using the least loaded IBGP path, and if PE3 sends its join messages to PE2, no join loop is created.

4. If no data MDT TLV corresponds to the C-join message, the least loaded neighbor on a multicast tunnel interface is selected as the upstream interface.

On PE routers that have only IBGP paths toward the source (or RP), such as PE3, PIM join load balancing is performed as follows:

1. The PE router only finds a multicast tunnel interface as the RPF interface, and load balancing is done across the C-PIM neighbors on a multicast tunnel interface.

   Router PE3 load-balances PIM join messages received from the CE4 router across the IBGP paths to the PE1 and PE2 routers.

2. If any PE router has already advertised data MDT TLVs corresponding to the C-join messages, that PE router is selected as the RPF neighbor.

For a particular C-multicast flow, at least one of the PE routers having EIBGP paths toward the source (or RP) must use only the EBGP path to avoid or break join loops. As a result of the loop avoidance mechanism, a PE router is constrained to choose among EIBGP paths when a multicast tunnel interface is already present in the downstream list.

In Figure 113 on page 776, assuming that the CE2 host is interested in receiving traffic from the Source and CE2 initiates multiple PIM join messages for different groups (Group 1 with group address 203.0.113.1, and Group 2 with group address 203.0.113.2), the join messages for both groups arrive on the PE1 router.

Router PE1 then equally distributes the join messages between the EIBGP paths toward the Source. Assuming that Group 1 join messages are sent to the CE1 router directly using the EBGP path, and Group 2 join messages are sent to the PE2 router using the IBGP path, PE1 and PE2 become the RPF neighbors for Group 1 and Group 2 join messages, respectively.

When the CE3 router initiates Group 1 and Group 2 PIM join messages, the join messages for both groups arrive on the PE2 router. Router PE2 then equally distributes the join

messages between the EIBGP paths toward the Source. Since PE2 is the RPF neighbor for Group 2 join messages, it sends the Group 2 join messages directly to the CE1 router using the EBGP path. Group 1 join messages are sent to the PE1 router using the IBGP path.

However, if the CE4 router initiates multiple Group 1 and Group 2 PIM join messages, there is no control over how these join messages received on the PE3 router get distributed to reach the Source. The selection of the RPF neighbor by PE3 can affect PIM join load balancing on EIBGP paths.

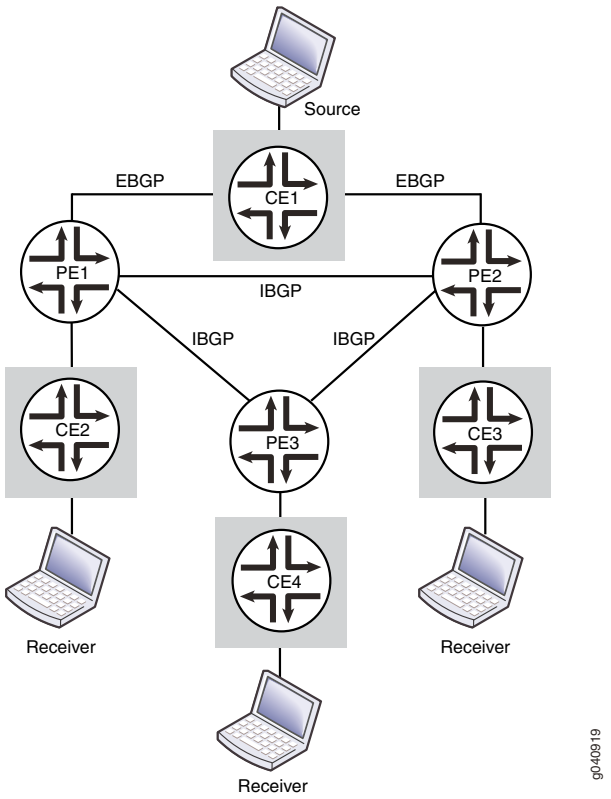
- If PE3 sends Group 1 join messages to PE1 and Group 2 join messages to PE2, there is no change in RPF neighbor. As a result, no join loops are created.
- If PE3 sends Group 1 join messages to PE2 and Group 2 join messages to PE1, there is a change in the RPF neighbor for the different groups resulting in the creation of join loops. To avoid potential join loops, PE1 and PE2 do not consider IBGP paths to send the join messages received from the PE3 router. Instead, the join messages are sent directly to the CE1 router using only the EBGP path.

The loop avoidance mechanism in a Draft-Rosen MVPN has the following limitations:

- Because the timing of arrival of join messages on remote PE routers determines the distribution of join messages, the distribution could be sub-optimal in terms of join count.
- Because join loops cannot be avoided and can occur due to the timing of join messages, the subsequent RPF interface change leads to loss of multicast traffic. This can be avoided by implementing the PIM make-before-break feature.

  The PIM make-before-break feature is an approach to detect and break C-PIM join loops in a Draft-Rosen MVPN. The C-PIM join messages are sent to the new RPF neighbor after establishing the PIM neighbor relationship, but before updating the related multicast forwarding entry. Though the upstream RPF neighbor would have updated its multicast forwarding entry and started sending the multicast traffic downstream, the downstream router does not forward the multicast traffic (because of RPF check failure) until the multicast forwarding entry is updated with the new RPF neighbor. This helps to ensure that the multicast traffic is available on the new path before switching the RPF interface of the multicast forwarding entry.

Figure 113: PIM Join Load Balancing on Draft-Rosen MVPN



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

PE1

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-5/0/4.0
set routing-instances vpn1 interface ge-5/2/0.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 1:1
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 192.0.2.4
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 192.0.2.5 peer-as 3
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 192.0.2.1
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
set routing-instances vpn1 protocols bgp group bgp1 neighbor 192.0.2.2 peer-as 4
set routing-instances vpn1 protocols pim vpn-group-address 198.51.100.1
```

```
set routing-instances vpn1 protocols pim rp static address 10.255.8.168
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols pim join-load-balance
```

PE2

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-2/0/3.0
set routing-instances vpn1 interface ge-4/0/5.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 2:2
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 10.90.10.1
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
set routing-instances vpn1 protocols bgp group bgp1 neighbor 10.90.10.2 peer-as 45
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 10.50.10.2
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.50.10.1 peer-as 4
set routing-instances vpn1 protocols pim vpn-group-address 198.51.100.1
set routing-instances vpn1 protocols pim rp static address 10.255.8.168
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols pim join-load-balance
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*. To configure the PE1 router:

NOTE: Repeat this procedure for every Juniper Networks router in the MVPN domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Configure a VPN routing and forwarding (VRF) instance.

```
[edit routing-instances vpn1]
user@PE1# set instance-type vrf
user@PE1# set interface ge-5/0/4.0
user@PE1# set interface ge-5/2/0.0
user@PE1# set interface lo0.1
user@PE1# set route-distinguisher 1:1
user@PE1# set vrf-target target:1:1
```

2. Enable protocol-independent load balancing for the VRF instance.

```
[edit routing-instances vpn1]
user@PE1# set routing-options multipath vpn-unequal-cost equal-external-internal
```

3. Configure BGP groups and neighbors to enable PE to CE routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set bgp export direct
user@PE1# set bgp group bgp type external
user@PE1# set bgp group bgp local-address 192.0.2.4
user@PE1# set bgp group bgp family inet unicast
user@PE1# set bgp group bgp neighbor 192.0.2.5 peer-as 3
user@PE1# set bgp group bgp1 type external
user@PE1# set bgp group bgp1 local-address 192.0.2.1
user@PE1# set bgp group bgp1 family inet unicast
user@PE1# set bgp group bgp1 neighbor 192.0.2.2 peer-as 4
```

4. Configure PIM to enable PE to CE multicast routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim vpn-group-address 198.51.100.1
user@PE1# set pim rp static address 10.255.8.168
```

5. Enable PIM on all network interfaces.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim interface all
```

6. Enable PIM join load balancing for the VRF instance.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim join-load-balance
```

Results From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
routing-instances {
    vpn1 {
        instance-type vrf;
        interface ge-5/0/4.0;
        interface ge-5/2/0.0;
        interface lo0.1;
        route-distinguisher 1:1;
        vrf-target target:1:1;
        routing-options {
            multipath {
                vpn-unequal-cost equal-external-internal;
            }
        }
        protocols {
            bgp {
                export direct;
                group bgp {
                    type external;
                    local-address 192.0.2.4;
                    family inet {
                        unicast;
                    }
```

```
            neighbor 192.0.2.5 {
              peer-as 3;
            }
          }
          group bgp1 {
            type external;
            local-address 192.0.2.1;
            family inet {
              unicast;
            }
            neighbor 192.0.2.2 {
              peer-as 4;
            }
          }
        }
        pim {
          vpn-group-address 198.51.100.1;
          rp {
            static {
              address 10.255.8.168;
            }
          }
          interface all;
          join-load-balance;
        }
      }
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying PIM Join Load Balancing for Different Groups of Join Messages on page 779

Verifying PIM Join Load Balancing for Different Groups of Join Messages

Purpose Verify PIM join load balancing for the different groups of join messages received on the PE1 router.

Action From operational mode, run the **show pim join instance extensive** command.

```
user@PE1>show pim join instance extensive
 Instance: PIM.vpn1 Family: INET
 R = Rendezvous Point Tree, S = Sparse, W = Wildcard

 Group: 203.0.113.1
     Source: *
     RP: 10.255.8.168
     Flags: sparse,rptree,wildcard
     Upstream interface: ge-5/2/0.1
     Upstream neighbor: 10.10.10.2
     Upstream state: Join to RP
```

```
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207

Group: 203.0.113.2
    Source: *
    RP: 10.255.8.168
    Flags: sparse,rptree,wildcard
    Upstream interface:  mt-5/0/10.32768
    Upstream neighbor: 19.19.19.19
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207

Group: 203.0.113.3
    Source: *
    RP: 10.255.8.168
    Flags: sparse,rptree,wildcard
    Upstream interface: ge-5/2/0.1
    Upstream neighbor: 10.10.10.2
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207

Group: 203.0.113.4
    Source: *
    RP: 10.255.8.168
    Flags: sparse,rptree,wildcard
    Upstream interface:  mt-5/0/10.32768
    Upstream neighbor: 19.19.19.19
    Upstream state: Join to RP
    Downstream neighbors:
        Interface: ge-5/0/4.0
            10.40.10.2 State: Join Flags: SRW Timeout: 207
```

Meaning The output shows how the PE1 router has load-balanced the C-PIM join messages for four different groups.

- For Group 1 (group address: 203.0.113.1) and Group 3 (group address: 203.0.113.3) join messages, the PE1 router has selected the EBGP path toward the CE1 router to send the join messages.
- For Group 2 (group address: 203.0.113.2) and Group 4 (group address: 203.0.113.4) join messages, the PE1 router has selected the IBGP path toward the PE2 router to send the join messages.

Related Documentation

- PIM Join Load Balancing on Multipath MVPN Routes Overview on page 768
- Example: Configuring PIM Join Load Balancing on Next-Generation Multicast VPN on page 781

Example: Configuring PIM Join Load Balancing on Next-Generation Multicast VPN

This example shows how to configure multipath routing for external and internal virtual private network (VPN) routes with unequal interior gateway protocol (IGP) metrics and Protocol Independent Multicast (PIM) join load balancing on provider edge (PE) routers running next-generation multicast VPN (MVPN). This feature allows customer PIM (C-PIM) join messages to be load-balanced across available internal BGP (IBGP) upstream paths when there is no external BGP (EBGP) path present, and across available EBGP upstream paths when external and internal BGP (EIBGP) paths are present toward the source or rendezvous point (RP).

Requirements

This example uses the following hardware and software components:

- Three routers that can be a combination of M Series, MX Series, or T Series routers.
- Junos OS Release 12.1 running on all the devices.

Before you begin:

1. Configure the device interfaces.
2. Configure the following routing protocols on all PE routers:
   - OSPF
   - MPLS
   - LDP
   - PIM
   - BGP
3. Configure a multicast VPN.

Overview and Topology

Junos OS Release 12.1 and later support multipath configuration along with PIM join load balancing. This allows C-PIM join messages to be load-balanced across all available IBGP paths when there are only IBGP paths present, and across all available upstream EBGP paths when EIBGP paths are present toward the source (or RP). Unlike Draft-Rosen MVPN, next-generation MVPN does not utilize unequal EIBGP paths to send C-PIM join messages. This feature is applicable to IPv4 C-PIM join messages.

By default, only one active IBGP path is used to send the C-PIM join messages for a PE router having only IBGP paths toward the source (or RP). When there are EIBGP upstream paths present, only one active EBGP path is used to send the join messages.

In a next-generation MVPN, C-PIM join messages are translated into (or encoded as) BGP customer multicast (C-multicast) MVPN routes and advertised with the BGP MCAST-VPN address family toward the sender PE routers. A PE router originates a C-multicast MVPN route in response to receiving a C-PIM join message through its PE router to customer edge (CE) router interface. The two types of C-multicast MVPN routes are:

- Shared tree join route (C-*, C-G)
  - Originated by receiver PE routers.
  - Originated when a PE router receives a shared tree C-PIM join message through its PE-CE router interface.
- Source tree join route (C-S, C-G)
  - Originated by receiver PE routers.
  - Originated when a PE router receives a source tree C-PIM join message (C-S, C-G), or originated by the PE router that already has a shared tree join route and receives a source active autodiscovery route.

  The upstream path in a next-generation MVPN is selected using the Bytewise-XOR hash algorithm as specified in Internet draft draft-ietf-l3vpn-2547bis-mcast, *Multicast in MPLS/BGP IP VPNs*. The hash algorithm is performed as follows:

  1. The PE routers in the candidate set are numbered from lower to higher IP address, starting from **0**.
  2. A bytewise exclusive-or of all the bytes is performed on the C-root (source) and the C-G (group) address.
  3. The result is taken modulo *n*, where *n* is the number of PE routers in the candidate set. The result is **N**.
  4. **N** represents the IP address of the upstream PE router as numbered in Step 1.

During load balancing, if a PE router with one or more upstream IBGP paths toward the source (or RP) discovers a new IBGP path toward the same source (or RP), the C-PIM join messages distributed among previously existing IBGP paths get redistributed due to the change in the candidate PE router set.

In this example, PE1, PE2, and PE3 are the PE routers that have the multipath PIM join load-balancing feature configured. Router PE1 has two EBGP paths and one IBGP upstream path, PE2 has one EBGP path and one IBGP upstream path, and PE3 has two IBGP upstream paths toward the Source. Router CE4 is the customer edge (CE) router attached to PE3. Source and Receiver are the Free BSD hosts.

On PE routers that have EIBGP paths toward the source (or RP), such as PE1 and PE2, PIM join load balancing is performed as follows:

1. The C-PIM join messages are sent using EBGP paths only. IBGP paths are not used to propagate the join messages.

   In Figure 114 on page 784, the PE1 router distributes the join messages between the two EBGP paths to the CE1 router, and PE2 uses the EBGP path to CE1 to send the join messages.

2. If a PE router loses one or more EBGP paths toward the source (or RP), the RPF neighbor on the multicast tunnel interface is selected based on a hash mechanism.

   On discovering the first EBGP path, only new join messages get load-balanced across available EBGP paths, whereas the existing join messages on the multicast tunnel interface are not redistributed.

   If the EBGP path from the PE2 router to the CE1 router goes down, PE2 sends the join messages to PE1 using the IBGP path. When the EBGP path to CE1 is restored, only new join messages that arrive on PE2 use the restored EBGP path, whereas join messages already sent on the IBGP path are not redistributed.

On PE routers that have only IBGP paths toward the source (or RP), such as the PE3 router, PIM join load balancing is performed as follows:

1. The C-PIM join messages from CE routers get load-balanced only as BGP C-multicast data messages among IBGP paths.

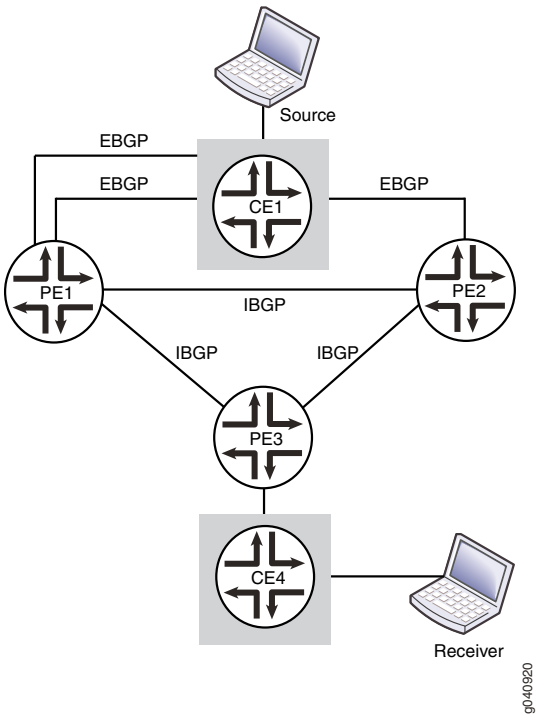
   In Figure 114 on page 784, assuming that the CE4 host is interested in receiving traffic from the Source, and CE4 initiates source join messages for different groups (Group 1 [C-S,C-G1] and Group 2 [C-S,C-G2]), the source join messages arrive on the PE3 router.

   Router PE3 then uses the Bytewise-XOR hash algorithm to select the upstream PE router to send the C-multicast data for each group. The algorithm first numbers the upstream PE routers from lower to higher IP address starting from **0**.

   Assuming that Router PE1 router is numbered **0** and Router PE2 is **1**, and the hash result for Group 1 and Group 2 join messages is **0** and **1**, respectively, the PE3 router selects PE1 as the upstream PE router to send Group 1 join messages, and PE2 as the upstream PE router to send the Group 2 join messages to the Source.

2. The shared join messages for different groups [C-*,C-G] are also treated in a similar way to reach the destination.

Figure 114: PIM Join Load Balancing on Next-Generation MVPN



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

PE1

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-3/0/1.0
set routing-instances vpn1 interface ge-3/3/2.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 1:1
set routing-instances vpn1 provider-tunnel rsvp-te label-switched-path-template
  default-template
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 10.40.10.1
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.40.10.2 peer-as 3
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 10.10.10.1
```

```
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
set routing-instances vpn1 protocols bgp group bgp1 neighbor 10.10.10.2 peer-as 3
set routing-instances vpn1 protocols pim rp static address 10.255.10.119
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols pim join-load-balance
set routing-instances vpn1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn1 protocols mvpn mvpn-join-load-balance bytewise-xor-hash
```

PE2

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-1/0/9.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 2:2
set routing-instances vpn1 provider-tunnel rsvp-te label-switched-path-template
  default-template
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp local-address 10.50.10.2
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.50.10.1 peer-as 3
set routing-instances vpn1 protocols pim rp static address 10.255.10.119
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn1 protocols mvpn mvpn-join-load-balance bytewise-xor-hash
```

PE3

```
set routing-instances vpn1 instance-type vrf
set routing-instances vpn1 interface ge-0/0/8.0
set routing-instances vpn1 interface lo0.1
set routing-instances vpn1 route-distinguisher 3:3
set routing-instances vpn1 provider-tunnel rsvp-te label-switched-path-template
  default-template
set routing-instances vpn1 vrf-target target:1:1
set routing-instances vpn1 vrf-table-label
set routing-instances vpn1 routing-options multipath vpn-unequal-cost
  equal-external-internal
set routing-instances vpn1 routing-options autonomous-system 1
set routing-instances vpn1 protocols bgp export direct
set routing-instances vpn1 protocols bgp group bgp type external
set routing-instances vpn1 protocols bgp group bgp local-address 10.80.10.1
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 10.80.10.2 peer-as 2
set routing-instances vpn1 protocols pim rp static address 10.255.10.119
set routing-instances vpn1 protocols pim interface all
set routing-instances vpn1 protocols mvpn mvpn-mode rpt-spt
set routing-instances vpn1 protocols mvpn mvpn-join-load-balance bytewise-xor-hash
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode*. To configure the PE1 router:

NOTE: Repeat this procedure for every Juniper Networks router in the MVPN domain, after modifying the appropriate interface names, addresses, and any other parameters for each router.

1. Configure a VPN routing forwarding (VRF) routing instance.

```
[edit routing-instances vpn1]
user@PE1# set instance-type vrf
user@PE1# set interface ge-3/0/1.0
user@PE1# set interface ge-3/3/2.0
user@PE1# set interface lo0.1
user@PE1# set route-distinguisher 1:1
user@PE1# set provider-tunnel rsvp-te label-switched-path-template
  default-template
user@PE1# set vrf-target target:1:1
user@PE1# set vrf-table-label
```

2. Enable protocol-independent load balancing for the VRF instance.

```
[edit routing-instances vpn1]
user@PE1# set routing-options multipath vpn-unequal-cost equal-external-internal
```

3. Configure BGP groups and neighbors to enable PE to CE routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set bgp export direct
user@PE1# set bgp group bgp type external
user@PE1# set bgp group bgp local-address 10.40.10.1
user@PE1# set bgp group bgp family inet unicast
user@PE1# set bgp group bgp neighbor 10.40.10.2 peer-as 3
user@PE1# set bgp group bgp1 type external
user@PE1# set bgp group bgp1 local-address 10.10.10.1
user@PE1# set bgp group bgp1 family inet unicast
user@PE1# set bgp group bgp1 neighbor 10.10.10.2 peer-as 3
```

4. Configure PIM to enable PE to CE multicast routing.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim rp static address 10.255.10.119
```

5. Enable PIM on all network interfaces.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim interface all
```

6. Enable PIM join load balancing for the VRF instance.

```
[edit routing-instances vpn1 protocols]
user@PE1# set pim join-load-balance
```

7. Configure the mode for C-PIM join messages to use rendezvous-point trees, and switch to the shortest-path tree after the source is known.

```
[edit routing-instances vpn1 protocols]
user@PE1# set mvpn mvpn-mode rpt-spt
```

8. Configure the VRF instance to use the Bytewise-XOR hash algorithm.

```
[edit routing-instances vpn1 protocols]
user@PE1# set mvpn mvpn-join-load-balance bytewise-xor-hash
```

Results

From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show routing-instances
routing-instances {
  vpn1 {
    instance-type vrf;
    interface ge-3/0/1.0;
    interface ge-3/3/2.0;
    interface lo0.1;
    route-distinguisher 1:1;
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
    }
    vrf-target target:1:1;
    vrf-table-label;
    routing-options {
      multipath {
        vpn-unequal-cost equal-external-internal;
      }
    }
    protocols {
      bgp {
        export direct;
        group bgp {
          type external;
          local-address 10.40.10.1;
          family inet {
            unicast;
          }
          neighbor 10.40.10.2 {
            peer-as 3;
```

```
        }
      }
      group bgp1 {
        type external;
        local-address 10.10.10.1;
        family inet {
          unicast;
        }
        neighbor 10.10.10.2 {
          peer-as 3;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.255.10.119;
        }
      }
      interface all;
      join-load-balance;
    }
    mvpn {
      mvpn-mode {
        rpt-spt;
      }
      mvpn-join-load-balance {
        bytewise-xor-hash;
      }
    }
  }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Verifying MVPN C-Multicast Route Information for Different Groups of Join Messages on page 788

Verifying MVPN C-Multicast Route Information for Different Groups of Join Messages

Purpose Verify MVPN C-multicast route information for different groups of join messages received on the PE3 router.

Action From operational mode, run the **show mvpn c-multicast** command.

```
user@PE3>
MVPN instance:
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel
```

```
Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Family : INET

Instance : vpn1
  MVPN Mode : RPT-SPT
  C-mcast IPv4 (S:G)            Ptnl                              St
  0.0.0.0/0:203.0.113.1/24        RSVP-TE P2MP:10.255.10.2, 5834,10.255.10.2
  192.0.2.2/24:203.0.113.1/24       RSVP-TE P2MP:10.255.10.2, 5834,10.255.10.2
  0.0.0.0/0:203.0.113.2/24        RSVP-TE P2MP:10.255.10.14, 47575,10.255.10.14
  192.0.2.2/24:203.0.113.2/24      RSVP-TE P2MP:10.255.10.14, 47575,10.255.10.14
```

Meaning The output shows how the PE3 router has load-balanced the C-multicast data for the different groups.

- For source join messages (S,G):
  - 192.0.2.2/24:203.0.113.1/24 (S,G1) toward the PE1 router (10.255.10.2 is the loopback address of Router PE1).
  - 192.0.2.2/24:203.0.113.2/24 (S,G2) toward the PE2 router (10.255.10.14 is the loopback address of Router PE2).
- For shared join messages (*,G):
  - 0.0.0.0/0:203.0.113.1/24 (*,G1) toward the PE1 router (10.255.10.2 is the loopback address of Router PE1).
  - 0.0.0.0/0:203.0.113.2/24 (*,G2) toward the PE2 router (10.255.10.14 is the loopback address of Router PE2).

Related Documentation

- PIM Join Load Balancing on Multipath MVPN Routes Overview on page 768

Example: Configuring PIM Make-Before-Break Join Load Balancing

Understanding the PIM Automatic Make-Before-Break Join Load-Balancing Feature

The PIM automatic make-before-break (MBB) join load-balancing feature introduces redistribution of PIM joins on equal-cost multipath (ECMP) links, with minimal disruption of traffic, when an interface is added to an ECMP path.

The existing PIM join load-balancing feature enables distribution of joins across ECMP links. In case of a link failure, the joins are redistributed among the remaining ECMP links, and traffic is lost. The addition of an interface causes no change to this distribution of joins unless the **clear pim join-distribution** command is used to load-balance the existing

joins to the new interface. If the PIM automatic MBB join load-balancing feature is configured, this process takes place automatically.

The feature can be enabled by using the **automatic** statement at the **[edit protocols pim join-load-balance]** hierarchy level. When a new neighbor is available, the time taken to create a path to the neighbor (standby path) can be configured by using the **standby-path-creation-delay** *seconds* statement at the **[edit protocols pim]** hierarchy level. In the absence of this statement, the standby path is created immediately, and the joins are redistributed as soon as the new neighbor is added to the network. For a join to be moved to the standby path in the absence of traffic, the **idle-standby-path-switchover-delay** *seconds* statement is configured at the **[edit protocols pim]** hierarchy level. In the absence of this statement, the join is not moved until traffic is received on the standby path.

```
protocols {
  pim {
    join-load-balance {
      automatic;
    }
    standby-path-creation-delay seconds;
    idle-standby-path-switchover-delay seconds;
  }
}
```

Example: Configuring PIM Make-Before-Break Join Load Balancing

This example shows how to configure the PIM make-before-break (MBB) join load-balancing feature.

Requirements

This example uses the following hardware and software components:

- Three routers that can be a combination of M Series Multiservice Edge Routers (M120 and M320 only), MX Series 3D Universal Edge Routers, or T Series Core Routers (TX Matrix and TX Matrix Plus only).
- Junos OS Release 12.2 or later.

Before you configure the MBB feature, be sure you have:

- Configured the device interfaces.
- Configured an interior gateway protocol (IGP) for both IPv4 and IPv6 routes on the devices (for example, OSPF and OSPFv3).
- Configured multiple ECMP interfaces (logical tunnels) using VLANs on any two routers (for example, Routers R1 and R2).

Overview

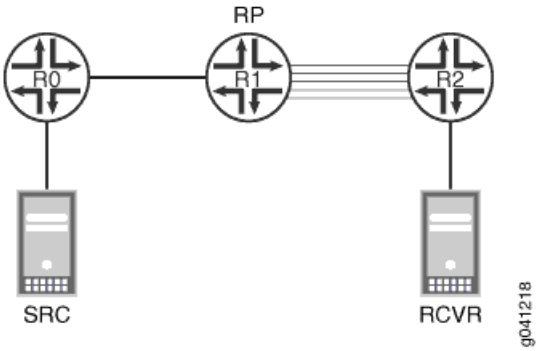
Junos OS provides a PIM automatic MBB join load-balancing feature to ensure that PIM joins are evenly redistributed to all upstream PIM neighbors on an equal-cost multipath (ECMP) path. When an interface is added to an ECMP path, MBB provides a switchover to an alternate path with minimal traffic disruption.

Topology

In this example, three routers are connected in a linear manner between source and receiver. An IGP protocol and PIM sparse mode are configured on all three routers. The source is connected to Router R0, and five interfaces are configured between Routers R1 and R2. The receiver is connected to Router R2, and PIM automatic MBB join load balancing is configured on Router R2.

Figure 115 on page 791 shows the topology used in this example.

Figure 115: Configuring PIM Automatic MBB Join Load Balancing



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router R0 (Source)

```
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim rp static address 10.255.12.34
set protocols pim rp static address abcd::10:255:12:34
```

Router R1 (RP)

```
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim rp local family inet address 10.255.12.34
set protocols pim rp local family inet6 address abcd::10:255:12:34
```

Router R2 (Receiver)

```
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim rp static address 10.255.12.34
set protocols pim rp static address abcd::10:255:12:34
```

```
set protocols mld interface ge-0/0/3 version 1
set protocols mld interface ge-0/0/3 static group ff05::e100:1 group-count 100
set protocols pim join load-balance automatic
set protocols pim standby-path-creation-delay 5
set protocols pim idle-standby-path-switchover-delay 10
```

Configuring PIM MBB Join Load Balancing

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure PIM MBB join load balancing across the setup:

1. Configure PIM sparse mode on all three routers.

```
[edit protocols pim interface all]
user@host# set mode sparse
user@host# set version 2
```

2. Configure Router R1 as the RP.

```
[edit protocols pim rp local]
user@R1# set family inet address 10.255.12.34
user@R1# set family inet6 address abcd::10:255:12:34
```

3. Configure the RP static address on non-RP routers (R0 and R2).

```
[edit protocols pim rp ]
user@host# set static address 10.255.12.34
user@host# set static address abcd::10:255:12:34
```

4. Configure the Multicast Listener Discovery (MLD) group for ECMP interfaces on Router R2.

```
[edit protocols mld interface ge-0/0/3]
user@R2# set version 1
user@R2# set static group ff05::e100:1 group-count 100
```

5. Configure the PIM MBB join load-balancing feature on the receiver router (Router R2).

```
[edit protocols pim]
user@R2# set join load-balance automatic
user@R2# set standby-path-creation-delay 5
user@R2# set idle-standby-path-switchover-delay 10
```

Results

From configuration mode, confirm your configuration by entering the **show protocols** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R0# show protocols
ospf {
    area 0.0.0.0 {
        interface lo0.0;
        interface ge-0/0/3.1;
        interface ge-0/0/3.2;
        interface ge-0/0/3.3;
        interface ge-0/0/3.4;
        interface ge-0/0/3.5;
    }
}
    ospf3 {
        area 0.0.0.0 {
            interface lo0.0;
            interface ge-0/0/3.1;
            interface ge-0/0/3.2;
            interface ge-0/0/3.3;
            interface ge-0/0/3.4;
            interface ge-0/0/3.5;
        }
    }
    pim {
        rp {
            static {
                address 10.255.12.34;
                address abcd::10:255:12:34;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
        interface ge-0/0/3.1;
        interface ge-0/0/3.2;
        interface ge-0/0/3.3;
        interface ge-0/0/3.4;
        interface ge-0/0/3.5;
    }
user@R1# show protocols
ospf {
    area 0.0.0.0 {
        interface lo0.0;
        interface ge-0/0/3.1;
        interface ge-0/0/3.2;
        interface ge-0/0/3.3;
        interface ge-0/0/3.4;
        interface ge-0/0/3.5;
    }
}
    ospf3 {
        area 0.0.0.0 {
            interface lo0.0;
```

```
        interface ge-0/0/3.1;
        interface ge-0/0/3.2;
        interface ge-0/0/3.3;
        interface ge-0/0/3.4;
        interface ge-0/0/3.5;
      }
    }
    pim {
      rp {
        local {
          family inet {
            address 10.255.12.34;
          }
          family inet6 {
            address abcd::10:255:12:34;
          }
        }
      }
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
      interface ge-0/0/3.1;
      interface ge-0/0/3.2;
      interface ge-0/0/3.3;
      interface ge-0/0/3.4;
      interface ge-0/0/3.5;
    }
```

```
user@R2# show protocols
mld {
  interface ge-0/0/3.1 {
    version 1;
    static {
      group ff05::e100:1 {
        group-count 100;
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0;
      interface ge-1/0/7.1;
      interface ge-1/0/7.2;
      interface ge-1/0/7.3;
      interface ge-1/0/7.4;
      interface ge-1/0/7.5;
      interface ge-0/0/3.1;
    }
  }
  ospf3 {
    area 0.0.0.0 {
      interface lo0.0;
```

```
        interface ge-1/0/7.1;
        interface ge-1/0/7.2;
        interface ge-1/0/7.3;
        interface ge-1/0/7.4;
        interface ge-1/0/7.5;
        interface ge-0/0/3.1;
      }
    }
    pim {
      rp {
        static {
          address 10.255.12.34;
          address abcd::10:255:12:34;
        }
      }
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
      interface ge-1/0/7.1;
      interface ge-1/0/7.2;
      interface ge-1/0/7.3;
      interface ge-1/0/7.4;
      interface ge-1/0/7.5;
      interface ge-0/0/3.1;
      join-load-balance {
        automatic;
      }
      standby-path-creation-delay 5;
      idle-standby-path-switchover-delay 10;
    }
```

Verification

Verifying Interface Configuration

Purpose Verify that the configured interfaces are functional.

Action Send 100 (S,G) joins from the receiver to Router R2 . From the operational mode of Router R2, run the **show pim interfaces** command.

```
user@R2> show pim interfaces

Stat = Status, V = Version, NbrCnt = Neighbor Count,
S = Sparse, D = Dense, B = Bidirectional,
DR = Designated Router, P2P = Point-to-point link,
Active = Bidirectional is active, NotCap = Not Bidirectional Capable
Name       Stat Mode IP V State     NbrCnt JoinCnt(sg/*g) DR address
ge-0/0/3.1 Up      S 4  2 DR,NotCap 0      0/0            70.0.0.1
ge-1/0/7.1 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.2
ge-1/0/7.2 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.6
ge-1/0/7.3 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.10
ge-1/0/7.4 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.14
ge-1/0/7.5 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.18
```

The output lists all the interfaces configured for use with the PIM protocol. The **Stat** field indicates the current status of the interface. The **DR address** field lists the configured IP addresses. All the interfaces are operational. If the output does not indicate that the interfaces are operational, reconfigure the interfaces before proceeding.

Meaning All the configured interfaces are functional in the network.

Verifying PIM

Purpose Verify that PIM is operational in the configured network.

Action From operational mode, enter the **show pim statistics** command.

```
user@R2> show pim statistics

PIM Message type        Received       Sent  Rx errors
V2 Hello                    4253       5269          0
V2 Register                    0          0          0
V2 Register Stop               0          0          0
V2 Join Prune                  0       1750          0
V2 Bootstrap                   0          0          0
V2 Assert                      0          0          0
V2 Graft                       0          0          0
V2 Graft Ack                   0          0          0
V2 Candidate RP                0          0          0
V2 State Refresh               0          0          0
V2 DF Election                 0          0          0
V1 Query                       0          0          0
V1 Register                    0          0          0
V1 Register Stop               0          0          0
V1 Join Prune                  0          0          0
V1 RP Reachability             0          0          0
V1 Assert                      0          0          0
V1 Graft                       0          0          0
V1 Graft Ack                   0          0          0
AutoRP Announce                0          0          0
AutoRP Mapping                 0          0          0
AutoRP Unknown type            0
Anycast Register               0          0          0
Anycast Register Stop          0          0          0

Global Statistics

Hello dropped on neighbor policy              0
Unknown type                                  0
V1 Unknown type                               0
Unknown Version                               0
Neighbor unknown                              0
Bad Length                                    0
Bad Checksum                                  0
Bad Receive If                                0
Rx Bad Data                                   0
Rx Intf disabled                              0
Rx V1 Require V2                              0
Rx V2 Require V1                              0
Rx Register not RP                            0
Rx Register no route                          0
Rx Register no decap if                       0
Null Register Timeout                         0
RP Filtered Source                            0
Rx Unknown Reg Stop                           0
Rx Join/Prune no state                        0
Rx Join/Prune on upstream if                  0
Rx Join/Prune for invalid group               0
Rx Join/Prune messages dropped                0
Rx sparse join for dense group                0
Rx Graft/Graft Ack no state                   0
Rx Graft on upstream if                       0
Rx CRP not BSR                                0
Rx BSR when BSR                               0
Anycast Register Stop          0          0          0
```

The **V2 Hello** field lists the number of PIM hello messages sent and received. The **V2 Join Prune** field lists the number of join messages sent before the **join-prune-timeout** value is reached. If both values are nonzero, PIM is functional.

Meaning PIM is operational in the network.

Verifying the PIM Automatic MBB Join Load-Balancing Feature

Purpose Verify that the PIM automatic MBB join load-balancing feature works as configured.

Action To see the effect of the MBB feature on Router R2:

1. Run the **show pim interfaces** operational mode command before disabling an interface.

```
user@R2> show pim interfaces
Stat = Status, V = Version, NbrCnt = Neighbor Count,
S = Sparse, D = Dense, B = Bidirectional,
DR = Designated Router, P2P = Point-to-point link,
Active = Bidirectional is active, NotCap = Not Bidirectional Capable
Name       Stat Mode IP V State     NbrCnt JoinCnt(sg/*g) DR address
ge-0/0/3.1 Up      S 4  2 DR,NotCap 0      0/0            70.0.0.1
ge-1/0/7.1 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.2
ge-1/0/7.2 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.6
ge-1/0/7.3 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.10
ge-1/0/7.4 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.14
ge-1/0/7.5 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.18
```

   The **JoinCnt(sg/*g)** field shows that the 100 joins are equally distributed among the five interfaces.

2. Disable the **ge-1/0/7.5** interface.

```
[edit]
user@R2# set interfaces ge-1/0/7.5 disable
user@R2# commit
```

3. Run the **show pim interfaces** command to check if load balancing of joins is taking place.

```
user@R2> show pim interfaces
Stat = Status, V = Version, NbrCnt = Neighbor Count,
S = Sparse, D = Dense, B = Bidirectional,
DR = Designated Router, P2P = Point-to-point link,
Active = Bidirectional is active, NotCap = Not Bidirectional Capable
Name       Stat Mode IP V State     NbrCnt JoinCnt(sg/*g) DR address
ge-0/0/3.1 Up      S 4  2 DR,NotCap 0      0/0            70.0.0.1
ge-1/0/7.1 Up      S 4  2 DR,NotCap 1      25/0           14.0.0.2
ge-1/0/7.2 Up      S 4  2 DR,NotCap 1      25/0           14.0.0.6
ge-1/0/7.3 Up      S 4  2 DR,NotCap 1      25/0           14.0.0.10
ge-1/0/7.4 Up      S 4  2 DR,NotCap 1      25/0           14.0.0.14
```

   The **JoinCnt(sg/*g)** field shows that the 100 joins are equally redistributed among the four active interfaces.

4. Add the removed interface on Router R2.

```
[edit]
user@R2# delete interfaces ge-1/0/7.5 disable
user@R2# commit
```

5. Run the **show pim interfaces** command to check if load balancing of joins is taking place after enabling the inactive interface.

```
user@R2> show pim interfaces
Stat = Status, V = Version, NbrCnt = Neighbor Count,
S = Sparse, D = Dense, B = Bidirectional,
DR = Designated Router, P2P = Point-to-point link,
Active = Bidirectional is active, NotCap = Not Bidirectional Capable
Name       Stat Mode IP V State     NbrCnt JoinCnt(sg/*g) DR address
ge-0/0/3.1 Up      S 4  2 DR,NotCap 0      0/0            70.0.0.1
ge-1/0/7.1 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.2
ge-1/0/7.2 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.6
ge-1/0/7.3 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.10
ge-1/0/7.4 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.14
ge-1/0/7.5 Up      S 4  2 DR,NotCap 1      20/0           14.0.0.18
```

The **JoinCnt(sg/*g)** field shows that the 100 joins are equally distributed among the five interfaces.

NOTE: This output should resemble the output in Step 1.

Meaning The PIM automatic MBB join load-balancing feature works as configured.

See Also

- Examples: Configuring MLD on page 50
- join-load-balance on page 1116

PART 6

Configuring General Multicast Options

CHAPTER 23

Preventing Routing Loops with Reverse Path Forwarding

Examples: Configuring Reverse Path Forwarding

Understanding Multicast Reverse Path Forwarding

Unicast forwarding decisions are typically based on the destination address of the packet arriving at a router. The unicast routing table is organized by destination subnet and mainly set up to forward the packet toward the destination.

In multicast, the router forwards the packet away from the source to make progress along the distribution tree and prevent routing loops. The router's multicast forwarding state runs more logically by organizing tables based on the reverse path, from the receiver back to the root of the distribution tree. This process is known as *reverse-path forwarding* (*RPF*).

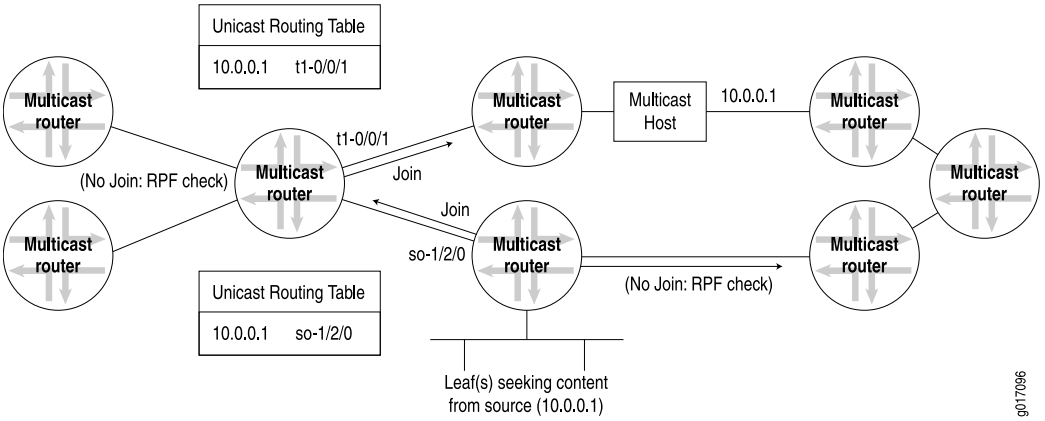
The router adds a branch to a distribution tree depending on whether the request for traffic from a multicast group passes the reverse-path-forwarding check (RPF check). Every multicast packet received must pass an RPF check before it is eligible to be replicated or forwarded on any interface.

The RPF check is essential for every router's multicast implementation. When a multicast packet is received on an interface, the router interprets the source address in the multicast IP packet as the destination address for a unicast IP packet. The source multicast address is found in the unicast routing table, and the outgoing interface is determined. If the outgoing interface found in the unicast routing table is the same as the interface that the multicast packet was received on, the packet passes the RPF check. Multicast packets

that fail the RPF check are dropped because the incoming interface is not on the *shortest path* back to the source.

Figure 116 on page 804 shows how multicast routers can use the unicast routing table to perform an RPF check and how the results obtained at each router determine where join messages are sent.

Figure 116: Multicast Routers and the RPF Check



Routers can build and maintain separate tables for RPF purposes. The router must have some way to determine its RPF interface for the group, which is the interface topologically closest to the root. For greatest efficiency, the distribution tree follows the shortest-path tree topology. The RPF check helps to construct this tree.

RPF Table

The RPF table plays the key role in the multicast router. The RPF table is consulted for every RPF check, which is performed at intervals on multicast packets entering the multicast router. Distribution trees of all types rely on the RPF table to form properly, and the multicast forwarding state also depends on the RPF table.

RPF checks are performed only on unicast addresses to find the upstream interface for the multicast source or RP.

The routing table used for RPF checks can be the same routing table used to forward unicast IP packets, or it can be a separate routing table used only for multicast RPF checks. In either case, the RPF table contains only unicast routes, because the RPF check is performed on the source address of the multicast packet, not the multicast group destination address, and a multicast address is forbidden from appearing in the source address field of an IP packet header. The unicast address can be used for RPF checks because there is only one source host for a particular stream of IP multicast content for a multicast group address, although the same content could be available from multiple sources.

If the same routing table used to forward unicast packets is also used for the RPF checks, the routing table is populated and maintained by the traditional unicast routing protocols such as BGP, IS-IS, OSPF, and the Routing Information Protocol (RIP). If a dedicated multicast RPF table is used, this table must be populated by some other method. Some multicast routing protocols (such as the Distance Vector Multicast Routing Protocol

[DVMRP]) essentially duplicate the operation of a unicast routing protocol and populate a dedicated RPF table. Others, such as PIM, do not duplicate routing protocol functions and must rely on some other routing protocol to set up this table, which is why PIM is protocol independent. .

Some traditional routing protocols such as BGP and IS-IS now have extensions to differentiate between different sets of routing information sent between routers for unicast and multicast. For example, there is multiprotocol BGP (MBGP) and multitopology routing in IS-IS (M-IS-IS). IS-IS routes can be added to the RPF table even when special features such as traffic engineering and "shortcuts" are turned on. Multicast Open Shortest Path First (MOSPF) also extends OSPF for multicast use, but goes further than MBGP or M-IS-IS and makes MOSPF into a complete multicast routing protocol on its own. When these routing protocols are used, routes can be tagged as multicast RPF routers and used by the receiving router differently than the unicast routing information.

Using the main unicast routing table for RPF checks provides simplicity. A dedicated routing table for RPF checks allows a network administrator to set up separate paths and routing policies for unicast and multicast traffic, allowing the multicast network to function more independently of the unicast network.

Multicast RPF Configuration Guidelines

You use multicast RPF checks to prevent multicast routing loops. Routing loops are particularly debilitating in multicast applications because packets are replicated with each pass around the routing loop.

In general, a router is to forward a multicast packet only if it arrives on the interface closest (as defined by a unicast routing protocol) to the origin of the packet, whether source host or rendezvous point (RP). In other words, if a unicast packet would be sent to the "destination" (the reverse path) on the interface that the multicast packet arrived on, the packet passes the RPF check and is processed. Multicast (or unicast) packets that fail the RPF check are not forwarded (this is the default behavior). For an overview of how a Juniper Networks router implements RPF checks with tables, see "Understanding Multicast Reverse Path Forwarding" on page 803.

However, there are network router configurations where multicast packets that fail the RPF check need to be forwarded. For example, when point-to-multipoint label-switched paths (LSPs) are used for distributing multicast traffic to PIM "islands" downstream from the egress router, the interface on which the multicast traffic arrives is not always the RPF interface. This is because LSPs do not follow the normal next-hop rules of independent packet routing.

In cases such as these, you can configure policies on the PE router to decide which multicast groups and sources are exempt from the default RPF check.

See Also

- *Junos OS MPLS Applications Library for Routing Devices*
- *Routing Policies, Firewall Filters, and Traffic Policers Feature Guide*

Example: Configuring a Dedicated PIM RPF Routing Table

This example explains how to configure a dedicated Protocol Independent Multicast (PIM) reverse path forwarding (RPF) routing table.

Requirements

Before you begin:

- Configure the router interfaces. See the *Interfaces Feature Guide for Security Devices*.
- Enable PIM. See "PIM Overview" on page 185.

This example uses the following software components:

- Junos OS Release 7.4 or later

Overview

By default, PIM uses the **inet.0** routing table as its RPF routing table. PIM uses an RPF routing table to resolve its RPF neighbor for a particular multicast source address and to resolve the RPF neighbor for the rendezvous point (RP) address. PIM can optionally use **inet.2** as its RPF routing table. The **inet.2** routing table is dedicated to this purpose.

PIM uses a single routing table for its RPF check, this ensures that the route with the longest matching prefix is chosen as the RPF route.

If multicast routes are exchanged by Multiprotocol Border Gateway Protocol MP-BGP or multitopology IS-IS, they are placed in **inet.2** by default.

Using **inet.2** as the RPF routing table enables you to have a control plane for multicast, which is independent of the normal unicast routing table. You might want to use **inet.2** as the RPF routing table for any of the following reasons:

- If you use traffic engineering or have an interior gateway protocol (IGP) configured for shortcuts, the router has label-switched paths (LSPs) installed as the next hops in **inet.2**. By applying policy, you can have the router install the routes with non-MPLS next-hops in the **inet.2** routing table.
- If you have an MPLS network that does not support multicast traffic over LSP tunnels, you need to configure the router to use a routing table other than **inet.0**. You can have the **inet.2** routing table populated with native IGP, BGP, and interface routes that can be used for RPF.

To populate the PIM RPF table, you use rib groups. A rib group is defined with the **rib-groups** statement at the **[edit routing-options]** hierarchy level. The rib group is applied to the PIM protocol by including the **rib-group** statement at the **[edit pim]** hierarchy level. A rib group is most frequently used to place routes in multiple routing tables.

When you configure rib groups for PIM, keep the following in mind:

- The **import-rib** statement copies routes from the protocol to the routing table.
- The **export-rib** statement has no effect on PIM.
- Only the first rib routing table specified in the **import-rib** statement is used by PIM for RPF checks.

You can also configure IS-IS or OSPF to populate **inet.2** with routes that have regular IP next hops. This allows RPF to work properly even when MPLS is configured for traffic engineering, or when IS-IS or OSPF are configured to use “shortcuts” for local traffic.

You can also configure the PIM protocol to use a rib group for RPF checks under a virtual private network (VPN) routing instance. In this case the rib group is still defined at the **[edit routing-options]** hierarchy level.

Configuration

Configuring a PIM RPF Routing Table Group Using Interface Routes

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set routing-options rib-groups mcast-rpf-rib import-rib inet.2
set protocols pim rib-group mcast-rpf-rib
set routing-options interface-routes rib-group inet if-rib
set routing-options rib-groups if-rib import-rib [ inet.0 inet.2 ]
```

Step-by-Step Procedure

In this example, the network administrator has decided to use the **inet.2** routing table for RPF checks. In this process, local routes are copied into this table by using an interface rib group.

To define an interface routing table group and use it to populate **inet.2** for RPF checks:

1. Use the **show multicast rpf** command to verify that the multicast RPF table is not populated with routes.

   ```
   user@host> show multicast rpf
   instance is not running
   ```

2. Create a multicast routing table group named **mcast-rpf-rib**.

   Each routing table group must contain one or more routing tables that Junos OS uses when importing routes (specified in the **import-rib** statement).

   Include the **import-rib** statement and specify the **inet.2** routing table at the **[edit routing-options rib-groups]** hierarchy level.

   ```
   [edit routing-options rib-groups]
   user@host# set mcast-rpf-rib import-rib inet.2
   ```

3. Configure PIM to use the **mcast-rpf-rib** rib group.

   The rib group for PIM can be applied globally or in a routing instance. In this example, the global configuration is shown.

   Include the **rib-group** statement and specify the **mcast-rpf-rib** rib group at the **[edit protocols pim]** hierarchy level.

   ```
   [edit protocols pim]
   user@host# set rib-group mcast-rpf-rib
   ```

4. Create an interface rib group named **if-rib**.

   Include the **rib-group** statement and specify the **inet** address family at the **[edit routing-options interface-routes]** hierarchy level.

   ```
   [edit routing-options interface-routes]
   user@host# set rib-group inet if-rib
   ```

5. Configure the **if-rib** rib group to import routes from the **inet.0** and **inet.2** routing tables.

   Include the **import-rib** statement and specify the **inet.0** and **inet.2** routing tables at the **[edit routing-options rib-groups]** hierarchy level.

   ```
   [edit routing-options rib-groups]
   user@host# set if-rib import-rib [ inet.0 inet.2 ]
   ```

6. Commit the configuration.

   ```
   user@host# commit
   ```

Verifying Multicast RPF Table

Purpose Verify that the multicast RPF table is now populated with routes.

Action Use the **show multicast rpf** command.

```
user@host> show multicast rpf
Multicast RPF table: inet.2 , 10 entries

10.0.24.12/30
    Protocol: Direct
    Interface: fe-0/1/2.0

10.0.24.13/32
    Protocol: Local

10.0.27.12/30
    Protocol: Direct
    Interface: fe-0/1/3.0

10.0.27.13/32
    Protocol: Local
```

```
10.0.224.8/30
    Protocol: Direct
    Interface: ge-1/3/3.0

10.0.224.9/32
    Protocol: Local

127.0.0.1/32
    Inactive

192.168.2.1/32
    Protocol: Direct
    Interface: lo0.0

192.168.187.0/25
    Protocol: Direct
    Interface: fxp0.0

192.168.187.12/32
    Protocol: Local
```

Meaning The first line of the sample output shows that the **inet.2** table is being used and that there are 10 routes in the table. The remainder of the sample output lists the routes that populate the **inet.2** routing table.

See Also

- Understanding Multicast Reverse Path Forwarding on page 803
- *Example: Enabling OSPF Traffic Engineering Support*
- *traffic-engineering*
- **show multicast rpf on page 1589**

Example: Configuring a PIM RPF Routing Table

This example shows how to configure and apply a PIM RPF routing table.

Requirements

Before you begin:

1. Determine whether the router is directly attached to any multicast sources. Receivers must be able to locate these sources.
2. Determine whether the router is directly attached to any multicast group receivers. If receivers are present, IGMP is needed.

3. Determine whether to configure multicast to use sparse, dense, or sparse-dense mode. Each mode has different configuration considerations.
4. Determine the address of the RP if sparse or sparse-dense mode is used.
5. Determine whether to locate the RP with the static configuration, BSR, or auto-RP method.
6. Determine whether to configure multicast to use its RPF routing table when configuring PIM in sparse, dense, or sparse-dense mode.
7. Configure the SAP and SDP protocols to listen for multicast session announcements. See "Configuring the Session Announcement Protocol" on page 415.
8. Configure IGMP. See "Configuring IGMP" on page 23.
9. Configure the PIM static RP. See "Configuring Static RP" on page 237.
10. Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 276 and "Example: Stopping Outgoing PIM Register Messages on a Designated Router" on page 272.

Overview

In this example, you name the new RPF routing table group **multicast-rfp-rib** and use **inet.2** for its export as well as its import routing table. Then you create a routing table group for the interface routes and name the RPF **if-rib**. Finally, you use **inet.2** and **inet.0** for its import routing tables, and add the new interface routing table group to the interface routes.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options rib-groups multicast-rpf-rib export-rib inet.2
set routing-options rib-groups multicast-rpf-rib import-rib inet.2
set protocols pim rib-group multicast-rpf-rib
set routing-options rib-groups if-rib import-rib inet.2
set routing-options rib-groups if-rib import-rib inet.0
set routing-options interface-routes rib-group inet if-rib
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For instructions on how to do that, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the PIM RPF routing table:

1. Configure a routing option and a group.

```
[edit]
user@host# edit routing-options rib-groups
```

2. Configure a name.

```
[edit routing-options rib-groups]
user@host# set multicast-rpf-rib export-rib inet.2
```

3. Create a new group for the RPF routing table.

```
[edit routing-options rib-groups]
user@host# set multicast-rpf-rib import-rib inet.2
```

4. Apply the new RPF routing table.

```
[edit protocols pim]
user@host# set rib-group multicast-rpf-rib
```

5. Create a routing table group for the interface routes.

```
[edit]
user@host# edit routing-options rib-groups
```

6. Configure a name for import routing table.

```
[edit routing-options rib-groups]
user@host# set if-rib import-rib inet.2
user@host# set if-rib import-rib inet.0
```

7. Set group to interface routes.

```
[edit routing-options interface-routes]
user@host# set rib-group inet if-rib
```

Results From configuration mode, confirm your configuration by entering the **show protocols** and **show routing-options** commands. If the output does not display the intended configuration, repeat the configuration instructions in this example to correct it.

```
[edit]
  user@host# show protocols
  pim {
  rib-group inet multicast-rpf-rib;
}
[edit]
  user@host# show routing-options
  interface-routes {
  rib-group inet if-rib;
  }
  static {
  route 0.0.0.0/0 next-hop 10.100.37.1;
  }
  rib-groups {
  multicast-rpf-rib {
  export-rib inet.2;
  import-rib inet.2;
  }
```

```
    if-rib {
    import-rib [ inet.2 inet.0 ];
    }
  }
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To confirm that the configuration is working properly, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 812
- Verifying the IGMP Version on page 812
- Verifying the PIM Mode and Interface Configuration on page 812
- Verifying the PIM RP Configuration on page 813
- Verifying the RPF Routing Table Configuration on page 813

Verifying SAP and SDP Addresses and Ports

Purpose Verify that SAP and SDP are configured to listen on the correct group addresses and ports.

Action From operational mode, enter the **show sap listen** command.

Verifying the IGMP Version

Purpose Verify that IGMP version 2 is configured on all applicable interfaces.

Action From operational mode, enter the **show igmp interface** command.

```
user@host> show igmp interface
Interface: ge-0/0/0.0
    Querier: 192.168.4.36
    State:         Up Timeout:     197 Version:  2 Groups:      0

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

Verifying the PIM Mode and Interface Configuration

Purpose Verify that PIM sparse mode is configured on all applicable interfaces.

Action From operational mode, enter the **show pim interfaces** command.

Verifying the PIM RP Configuration

Purpose Verify that the PIM RP is statically configured with the correct IP address.

Action From operational mode, enter the **show pim rps** command.

Verifying the RPF Routing Table Configuration

Purpose Verify that the PIM RPF routing table is configured correctly.

Action From operational mode, enter the **show multicast rpf** command.

See Also
- Configuring PIM Filtering on page 267
- Example: Configuring a Dedicated PIM RPF Routing Table on page 806
- Multicast Configuration Overview on page 16
- *Verifying a Multicast Configuration*

Example: Configuring RPF Policies

A multicast RPF policy disables RPF checks for a particular multicast (S,G) pair. You usually disable RPF checks on egress routing devices of a point-to-multipoint label-switched path (LSP), because the interface receiving the multicast traffic on a point-to-multipoint LSP egress router might not always be the RPF interface.

This example shows how to configure an RPF check policy named **disable-RPF-on-PE**. The **disable-RPF-on-PE** policy disables RPF checks on packets arriving for group 228.0.0.0/8 or from source address 196.168.25.6.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.

Overview

An RPF policy behaves like an import policy. If no policy term matches the input packet, the default action is to accept (that is, to perform the RPF check). The **route-filter**

statement filters group addresses, and the **source-address-filter** statement filters source addresses.

This example shows how to configure each condition as a separate policy and references both policies in the **rpf-check-policy** statement. This allows you to associate groups in one policy and sources in the other.

NOTE: Be careful when disabling RPF checks on multicast traffic. If you disable RPF checks in some configurations, multicast loops can result.

Changes to an RPF check policy take effect immediately:

- If no policy was previously configured, the policy takes effect immediately.
- If the policy name is changed, the new policy takes effect immediately and any packets no longer filtered are subjected to the RPF check.
- If the policy is deleted, all packets formerly filtered are subjected to the RPF check.
- If the underlying policy is changed, but retains the same name, the new conditions take effect immediately and any packets no longer filtered are subjected to the RPF check.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set policy-options policy-statement disable-RPF-from-group term first from route-filter
  228.0.0.0/8 orlonger
set policy-options policy-statement disable-RPF-from-group term first then reject
set policy-options policy-statement disable-RPF-from-source term first from
  source-address-filter 192.168.25.6/32 exact
set policy-options policy-statement disable-RPF-from-source term first then reject
set routing-options multicast rpf-check-policy [ disable-RPF-from-group
  disable-RPF-from-source ]
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure an RPF policy:

1. Configure a policy for group addresses.

```
[edit policy-options]
user@host# set policy-statement disable-RPF-for-group term first from route-filter
  228.0.0.0/8 orlonger
user@host# set policy-statement disable-RPF-for-group term first then reject
```

2. Configure a policy for a source address.

```
[edit policy-options]
user@host# set policy-statement disable-RPF-for-source term first from
  source-address-filter 192.168.25.6/32 exact
user@host# set policy-statement disable-RPF-for-source term first then reject
```

3. Apply the policies.

```
[edit routing-options]
user@host# set multicast rpf-check-policy [ disable-RPF-for-group
  disable-RPF-for-source ]
```

4. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show policy-options** and **show routing-options** commands.

```
user@host# show policy-options
policy-statement disable-RPF-from-group {
  term first {
    from {
      route-filter 228.0.0.0/8 orlonger;
    }
    then reject;
  }
}
policy-statement disable-RPF-from-source {
  term first {
    from {
      source-address-filter 192.168.25.6/32 exact;
    }
    then reject;
  }
}
```

```
user@host# show routing-options
multicast {
  rpf-check-policy [ disable-RPF-from-group disable-RPF-from-source ];
}
```

Verification

To verify the configuration, run the **show multicast rpf** command.

See Also

- Example: Configuring Ingress PE Redundancy on page 937
- Understanding Multicast Reverse Path Forwarding on page 803

Example: Configuring PIM RPF Selection

This example shows how to configure and verify the multicast PIM RPF next-hop neighbor selection for a group or (S,G) pair.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.
- Make sure that the RPF next-hop neighbor you want to specify is operating.

Overview

Multicast PIM RPF neighbor selection allows you to specify the RPF neighbor (next hop) and source address for a single group or multiple groups using a prefix list. RPF neighbor selection can only be configured for VPN routing and forwarding (VRF) instances.

If you have multiple service VRFs through which a receiver VRF can learn the same source or rendevous point (RP) address, PIM RPF checks typically choose the best path determined by the unicast protocol for all multicast flows. However, if RPF neighbor selection is configured, RPF checks are based on your configuration instead of the unicast routing protocols.

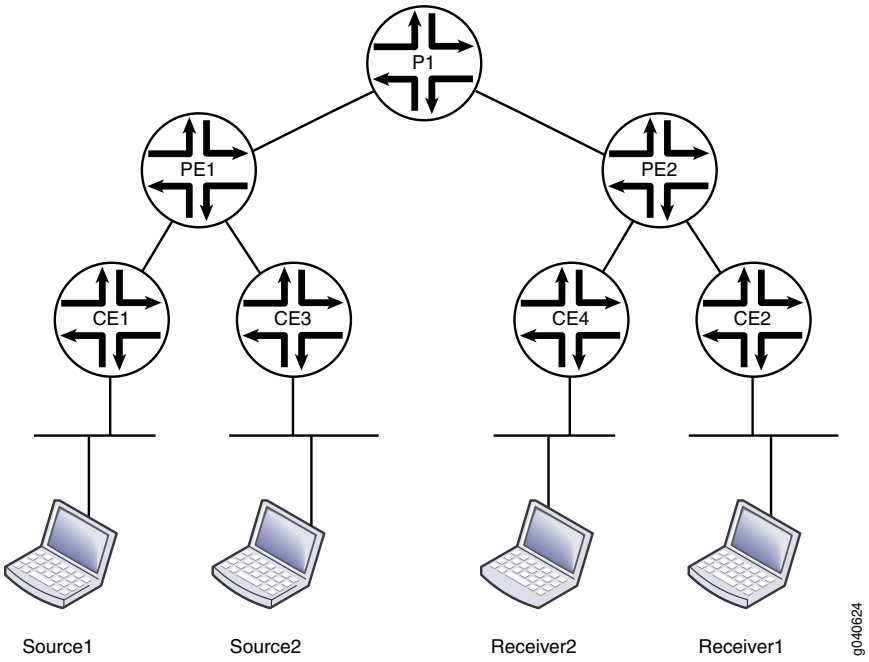
You can use this static RPF selection as a building block for particular applications. For example, an extranet. Suppose you want to split the multicast flows among parallel PIM links or assign one multicast flow to a specific PIM link. With static RPF selection configured, the router sends join and prune messages based on the configuration.

You can use wildcards to designate the source address. Whether or not you use wildcards affects how the PIM joins work:

- If you configure only a source prefix for a group, all (*,G) joins are sent to the next-hop neighbor selected by the unicast protocol, while (S,G) joins are sent to the next-hop neighbor specified for the source.
- If you configure only a wildcard source for a group, all (*,G) and (S,G) joins are sent to the upstream interface pointing to the wildcard source next-hop neighbor.
- If you configure both a source prefix and a wildcard source for a group, all (S,G) joins are sent to the next-hop neighbor defined for the source prefix, while (*,G) joins are sent to the next-hop neighbor specified for the wildcard source.

Figure 117 on page 817 shows the topology used in this example.

Figure 117: PIM RPF Selection



In this example, the RPF selection is configured on the receiver provider edge router (PE2).

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set routing-instance vpn-a protocols pim rpf-selection group 225.5.0.0/16 wildcard-source
    next-hop 10.12.5.2
set routing-instance vpn-a protocols pim rpf-selection prefix-list group12 wildcard-source
    next-hop 10.12.31.2
set routing-instance vpn-a protocols pim rpf-selection prefix-list group34 source
    22.1.12.0/24 next-hop 10.12.32.2
set policy-options prefix-list group12 225.1.1.0/24
set policy-options prefix-list group12 225.2.0.0/16
set policy-options prefix-list group34 225.3.3.3/32
set policy-options prefix-list group34 225.4.4.0/24
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure PIM RPF selection:

1. On PE2, configure RFP selection in a routing instance.

```
[edit routing-instance vpn-a protocols pim]
user@host# set rpf-selection group 225.5.0.0/16 wildcard-source next-hop 10.12.5.2
user@host# set rpf-selection prefix-list group12 wildcard-source next-hop 10.12.31.2
user@host# set rpf-selection prefix-list group34 source 22.1.12.0/24 next-hop
  10.12.32.2
user@host# exit
```

2. On PE2, configure the policy.

```
[edit policy-options]
set prefix-list group12 225.1.1.0/24
set prefix-list group12 225.2.0.0/16
set prefix-list group34 225.3.3.3/32
set prefix-list group34 225.4.4.0/24
```

3. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results From configuration mode, confirm your configuration by entering the **show policy-options** and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show policy-options
prefix-list group12 {
  225.1.1.0/24;
  225.2.0.0/16;
}
prefix-list group34 {
  225.3.3.3/32;
  225.4.4.0/24;
}
```

```
user@host# show routing-instances
vpn-a{
  protocols {
    pim {
      rpf-selection {
        group 225.5.0.0/16 {
          wildcard-source {
            next-hop 10.12.5.2;
          }
        }
        prefix-list group12 {
          wildcard-source {
            next-hop 10.12.31.2;
          }
        }
        prefix-list group34 {
          source 22.1.12.0/24 {
            next-hop 10.12.32.2;
          }
        }
      }
```

```
    }
  }
}
```

Verification

To verify the configuration, run the following commands, checking the upstream interface and the upstream neighbor:

- show pim join **extensive**
- show multicast route

See Also

- Example: Configuring RPF Policies on page 813
- RPF Table on page 804

Related Documentation

- Example: Configuring Ingress PE Redundancy on page 936

CHAPTER 24

Minimizing Packet Loss During Link Failure with Multicast-Only Fast Reroute

Understanding Multicast-Only Fast Reroute

Starting in Junos OS Release 14.1, Multicast-only fast reroute (MoFRR) functionality is available, in which packet loss is minimalized in PIM and multipoint LDP domains. MoFRR minimizes packet loss in a network when there is a link failure. It works by enhancing multicast routing protocols like Protocol Independent Multicast (PIM) and multipoint Label Distribution Protocol (multipoint LDP). MoFRR is supported on MX Series routers with MPC line cards. As a prerequisite, the router must be set to **network-services enhanced-ip** mode, and all the line-cards in the router must be MPCs.

With MoFRR enabled, join messages are sent on primary and backup upstream paths. Data packets are received from both the primary path and the backup paths. The redundant packets are discarded based on priority (weights that are assigned to the primary and backup paths). When a failure is detected on the primary path, the repair is made by changing the interface on which packets are accepted to the secondary interface. Because the repair is local , it is fast—greatly improving convergence times in the event of a link failure on the primary path.

Currently, the most likely real-world application for MoFRR is streaming IPTV. IPTV streams are multicast as UDP streams. Therefore, any lost packets are not retransmitted, and this can result in a less-than-satisfactory user experience. MoFRR can be used to improve this situation.

When fast reroute is applied to unicast streams, an upstream router preestablishes MPLS label-switched paths (LSPs) or precomputes an IP loop-free alternate (LFA) fast reroute backup path to handle failure of a segment in the downstream path.

In multicast routing, the traffic distribution graphs are usually originated by the receiver. This is unlike unicast routing, which usually establishes the path from the source to the receiver. Protocols that are capable of establishing multicast distribution graphs are PIM (for IP), multipoint LDP (for MPLS), and RSVP-TE (for MPLS). Of these, PIM and multipoint LDP receivers initiate the distribution graph setup, and therefore these are the two multicast protocols for which MoFRR is supported.

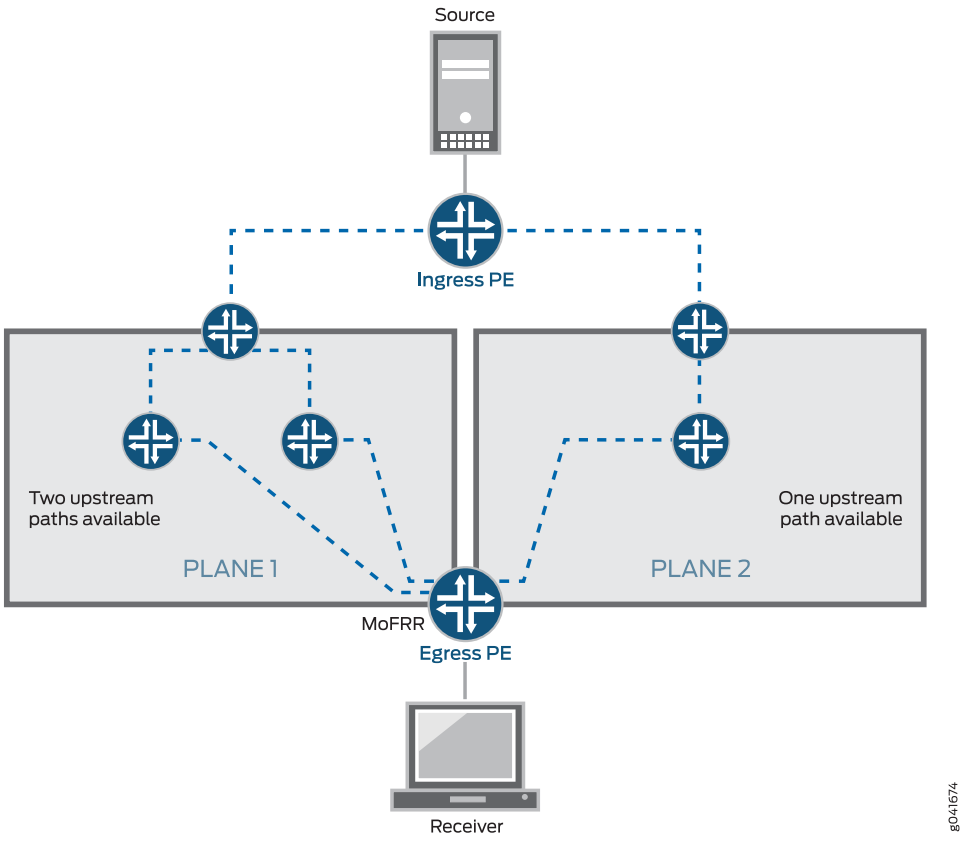
In a multicast tree, performing a reactive repair upon detection of a network-component failure can lead to significant traffic loss due to delay in setting up the alternative path. MoFRR reduces traffic loss in a multicast distribution tree when a network component fails. With MoFRR, one of the downstream routers that supports this feature sets up an alternative path toward the source to receive a backup live stream of the same multicast traffic. When a failure is detected on the primary stream, the MoFRR router switches to the backup stream.

With MoFRR enabled, for each (S,G) entry, two of the available upstream interfaces are used to send a join message and to receive multicast traffic. The protocol attempts to select two disjoint paths if two such paths are available. If disjoint paths are not available, the protocol selects two non-disjoint paths. If two non-disjoint paths are not available, only a primary path is selected with no backup. MoFRR is supported for both IPv4 and IPv6 protocol families.

In the context of load balancing, MoFRR prioritizes the disjoint backup in favor of load balancing the available paths.

Figure 118 on page 823 shows two paths from the egress provider edge (PE) router to the ingress PE router.

Figure 118: MoFRR Sample Topology



When enabled with MoFRR functionality, the egress router sets up two multicast trees, a primary path and a backup path, toward the multicast source for each (S,G). In other words, the egress router propagates the same (S,G) join messages toward two different upstream neighbors, thus creating two multicast trees.

One of the multicast trees goes through plane 1 and the other through plane 2, as shown in Figure 118 on page 823. For each (S,G), the egress PE router forwards traffic received on the primary path and drops traffic received on the backup path.

MoFRR is supported on both equal-cost multipath (ECMP) paths and non-ECMP paths. Unicast loop-free alternate (LFA) routes need to be enabled to support MoFRR on non-ECMP paths. LFA routes are enabled with the **link-protection** statement in the interior gateway protocol (IGP) configuration. When you enable link protection on an OSPF or IS-IS interface, Junos OS creates a backup LFA path to the primary next hop for all destination routes that traverse the protected interface.

Junos OS implements MoFRR in the IP network for IP MoFRR and at the MPLS label-edge router (LER) for multipoint LDP MoFRR.

Multipoint LDP MoFRR is used at the egress node of an MPLS network, where the packets are forwarded to an IP network. In the case of multipoint LDP MoFRR, the two paths toward the upstream PE router are established for receiving two streams of MPLS packets at the LER. One of the streams (the primary) is accepted, and the other one (the backup) is dropped at the LER. The backup stream is accepted if the primary path fails. A prerequisite for this feature is inband signaling support, as described in *Understanding Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs*.

PIM Functionality

Junos OS supports MoFRR for shortest-path tree (SPT) joins in PIM source-specific multicast (SSM) and any-source multicast (ASM). MoFRR is supported for both SSM and ASM ranges. To enable MoFRR for (*,G) joins, the mofrr-asm-starg configuration statement needs to be included. For each group G, either (S,G) or (*,G) (not both) will undergo MoFRR. (S,G) always takes precedence over (*,G).

With MoFRR enabled, a PIM router propagates join messages on two upstream RPF interfaces to receive multicast traffic on both links for the same join request. Preference is given to two paths that do not converge to the same immediate upstream router. PIM installs appropriate multicast routes with upstream RPF next hops with two (primary and backup) interfaces.

When the primary path fails, the backup path is upgraded to primary, and traffic is forwarded accordingly. If there are alternate paths available, a new backup path is calculated and the appropriate multicast route is updated or installed.

MoFRR can be enabled along with PIM join load balancing (with the join-load-balance **automatic** statement). However, in such cases the distribution of join messages among the links might not be even. When a new ECMP link is added, join messages on the primary path are redistributed and load-balanced. The join messages on the backup path might still follow the same path and might not be evenly redistributed.

MoFRR is enabled with a **[edit routing-options multicast stream-protection]** configuration and is managed by a set of filter policies. When an egress PIM router receives a join message or an IGMP report, the router checks for the MoFRR configuration.

If the MoFRR configuration is not present, PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 118 on page 823).

If the MoFRR configuration is present, Junos OS checks for a policy configuration.

If a policy is not present, Junos OS checks for primary and backup paths (upstream interfaces), and takes the following actions:

- If primary and backup paths are not available—PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 118 on page 823).
- If primary and backup paths are available—PIM sends the join message upstream toward two of the available upstream neighbors. Junos OS sets up primary and

secondary multicast paths to receive multicast traffic (for example, plane 1 in Figure 118 on page 823).

If a policy is present, Junos OS checks whether the policy allows MoFRR for this (S,G), and takes the following actions:

- If the policy check fails—PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 118 on page 823).
- If the policy check passes—Junos OS checks for primary and backup paths (upstream interfaces).
  - If the primary and backup paths are not available, PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 118 on page 823).
  - If the primary and backup paths are available, PIM sends the join message upstream toward two of the available upstream neighbors. Junos OS sets up primary and secondary multicast paths to receive multicast traffic (for example, plane 1 in Figure 118 on page 823).

Multipoint LDP Functionality

To avoid MPLS traffic duplication, the usual implementation of multipoint LDP selects only one upstream path. (See section 2.4.1.1. Determining One's 'upstream LSR' in RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*.)

For multipoint LDP MoFRR, the multipoint LDP node selects two separate upstream peers and sends two separate labels, one to each upstream peer. The same algorithm described in RFC 6388 is used to select the primary upstream path. The backup upstream path selection again uses the same algorithm but excludes the primary upstream LSR as a candidate. Two streams of MPLS traffic are sent to the egress node from the two different upstream peers. The MPLS traffic from only one of the upstream neighbors is selected as the primary path to accept the traffic, and the other path becomes the backup path. The traffic on the backup path is dropped. When the primary upstream path fails, the traffic from the backup path is then accepted. The multipoint LDP node selects the two upstream paths based on the interior gateway protocol (IGP) root node next hop.

A forwarding equivalency class (FEC) is a group of IP packets that are forwarded in the same manner, over the same path, and with the same forwarding treatment. Normally, the label that is put on a particular packet represents the FEC to which that packet is assigned. In MoFRR, two routes are placed into the mpls.0 table for each FEC—one route for the primary label and the other route for the backup label.

If there are parallel links toward the same immediate upstream node, both parallel links are considered to be the primary. At any point in time, the upstream node sends traffic on only one of the multiple parallel links.

A bud node is an LSR that is an egress LSR, but also has one or more directly connected downstream LSRs. In the case of a bud node, the traffic from the primary upstream path is forwarded to a downstream LSR. If the primary upstream path fails, the MPLS traffic from the backup upstream path is forwarded to the downstream LSR. This means that

the downstream LSR next hop is added to both MPLS routes along with the egress next hop.

MoFRR for multipoint LDP is enabled with a **[edit routing-options multicast stream-protection]** configuration and is managed by a set of filter policies.

If the multipoint LDP point-to-multipoint FEC is enabled for MoFRR, the following additional considerations are factored into upstream path selection:

- The targeted LDP sessions are skipped if there is a nontargeted LDP session. If there is a single targeted LDP session, the targeted LDP session is selected, but the corresponding point-to-multipoint FEC loses the MoFRR capability because there is no interface associated with the targeted LDP session.
- All interfaces that belong to the same upstream LSR are considered to be the primary path.
- For any root-node route updates, the upstream path is changed based on the latest next hops from the IGP. If a better path is available, multipoint LDP attempts to switch to the better path.

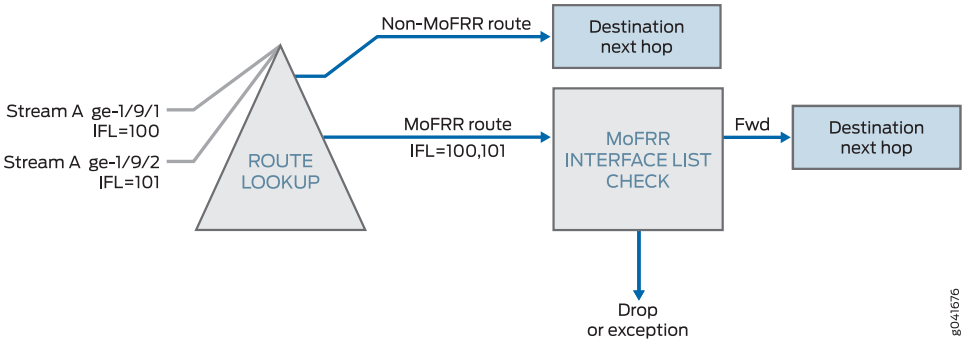
Packet Forwarding

For both PIM and multipoint LDP, multicast source stream selection is performed at the incoming interface. This prevents duplicate streams from being sent across the fabric and prevents multiple route lookups that result in drops, thus preserving fabric bandwidth and maximizing forwarding performance.

For PIM, each IP multicast stream contains the same destination address. Regardless of the interface on which the packets arrive, the packets have the same route. An interface list is attached to the route. Junos OS checks the interface upon which each packet arrives and forwards only those that are from the primary interface. If the interface matches a secondary interface, the packets are dropped. If no match is found, the packets are handled as exceptions in the control plane.

This process in shown in Figure 119 on page 826.

Figure 119: MoFRR IP Route Lookup in the Packet Forwarding Engine

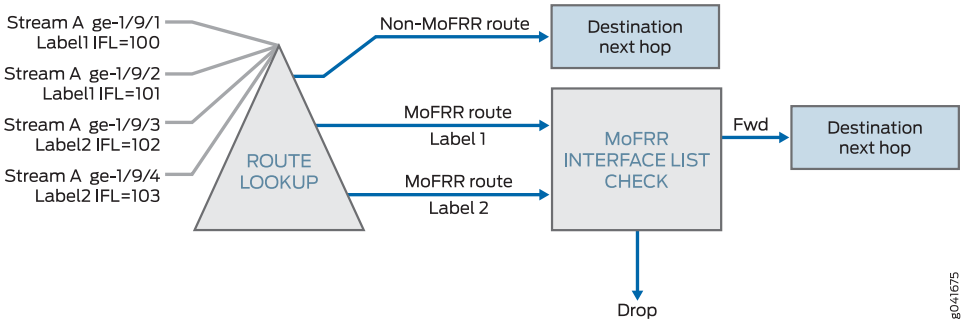


For multipoint LDP, multiple MPLS labels are used to control MoFRR stream selection. Each label represents a separate route, but each references the same interface list check.

Only the primary label is forwarded while all others are dropped. Multiple interfaces can receive packets using the same label.

This process in shown in Figure 120 on page 827.

Figure 120: MoFRR MPLS Route Lookup in the Packet Forwarding Engine



Limitations and Caveats

MoFRR has the following limitations and caveats:

- MoFRR failure detection is supported for immediate link protection of the router on which MoFRR is enabled and not on all the links (end-to-end) in the multicast traffic path.
- MoFRR supports FRR on two selected disjoint paths toward the source. Two of the selected upstream neighbors cannot be on the same interface—in other words, two upstream neighbors on a LAN segment. The same is true if the upstream interface happens to be a multicast tunnel interface.
- Detection of the maximum end-to-end disjoint upstream paths is not supported. The egress router only makes sure that there is a disjoint upstream node (the immediate previous hop). PIM and multipoint LDP do not support the equivalent of explicit route objects (EROs). Hence, disjoint upstream path detection is limited to control over the immediately previous hop node. Because of this limitation, the path to the upstream node of the previous hop selected as primary and backup might be shared.
- MoFRR does not apply to multipoint LDP traffic received on an RSVP tunnel because the RSVP tunnel is not associated with any interface.
- Some traffic loss is seen in the following scenarios:
  - A better upstream path becomes available on an egress node.
  - MoFRR is enabled or disabled on the egress node while there is an active traffic stream flowing.
- PIM join load balancing for join messages for backup paths are not supported.
- For a multicast group G, MoFRR is not allowed for both (S,G) and (*,G) join messages. (S,G) join messages have precedence over (*,G).
- MoFRR is not supported for multicast traffic streams that use two different multicast groups. Each (S,G) combination is treated as a unique multicast traffic stream.

- The bidirectional PIM range is not supported for MoFRR.
- PIM dense-mode is not supported for MoFRR
- Mixed upstream MoFRR is not supported. This refers to PIM multipoint LDP in-band signaling, wherein one upstream path is through multipoint LDP and the second upstream path is through PIM.
- Multicast statistics for the backup traffic stream are not maintained by PIM and therefore are not available in the operational output of **show** commands.
- Multipoint LDP labels as inner labels are not supported.
- If the source is reachable through multiple ingress provider edge (PE) routers, multipoint LDP MoFRR is not supported.
- Targeted upstream sessions are not selected as the upstream node for MoFRR.
- Rate monitoring is not supported.
- Multipoint LDP link protection on the backup path is not supported because there is no support for MoFRR inner labels.

Release History Table

Release	Description
14.1	Starting in Junos OS Release 14.1, Multicast-only fast reroute (MoFRR) functionality is available, in which packet loss is minimalized in PIM and multipoint LDP domains.

Related Documentation

- Configuring Multicast-Only Fast Reroute on page 834
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

Understanding Multicast-Only Fast Reroute on Switches

Starting in Junos OS Release 17.4R1, QFX Series switches support multicast-only fast reroute (MoFRR) functionality, which minimizes packet loss for traffic in a multicast distribution tree when link failures occur. MoFRR enhances multicast routing protocols like Protocol Independent Multicast (PIM). With MoFRR enabled, join messages are sent on primary and backup upstream paths towards a multicast source. Data packets are received from both the primary path and the backup paths. The redundant packets are discarded based on priority (weights that are assigned to the primary and backup paths). When a failure is detected on the primary path, the repair is made locally by changing the interface on which packets are accepted to the secondary interface for the backup path, so the repair is fast—greatly improving convergence times in the event of a link failure on the primary path.

One application for MoFRR is streaming IPTV. IPTV streams are multicast as UDP streams, so any lost packets are not retransmitted, and this can result in a less-than-satisfactory user experience. MoFRR can be used to improve this situation.

Overview of MoFRR on Switches

When fast reroute is applied to unicast streams, an upstream routing device precomputes an IP loop-free alternate (LFA) fast reroute backup path to handle failure of a segment in the downstream path.

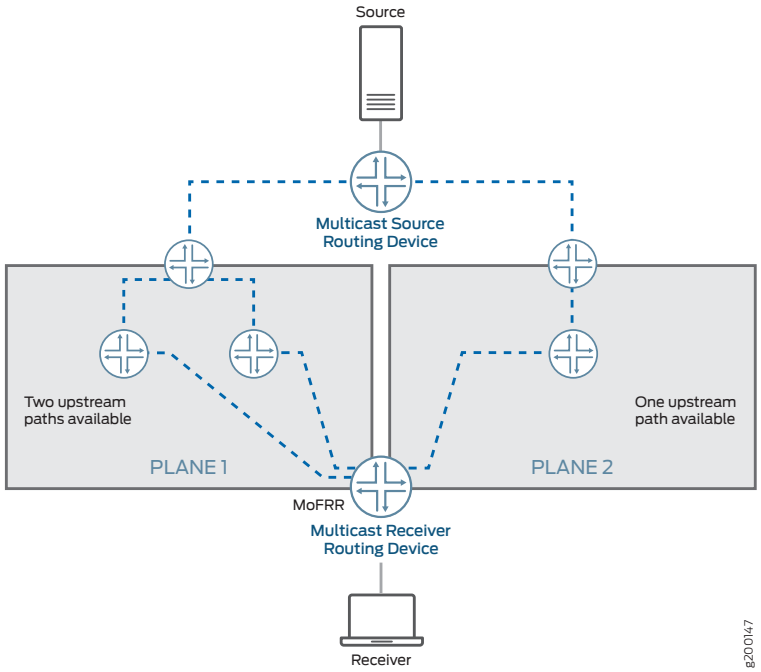
In multicast routing, the traffic distribution graphs are usually originated by the receiver. This is unlike unicast routing, which usually establishes the path from the source to the receiver. Protocol-independent Multicast (PIM) is a protocol that is capable of establishing multicast distribution graphs, and PIM receivers initiate the distribution graph setup, so MoFRR is supported in PIM domains.

In a multicast tree, performing a reactive repair upon detection of a network component failure can lead to significant traffic loss due to delay in setting up the alternative path. With MoFRR, one of the downstream devices that supports this feature sets up an alternative path toward the source to receive a backup live stream of the same multicast traffic. When a failure is detected on the primary stream, the MoFRR device switches to the backup stream.

With MoFRR enabled, for each (S,G) entry, two of the available upstream interfaces are used to send a join message and to receive multicast traffic. The protocol attempts to select two disjoint paths if two such paths are available. If disjoint paths are not available, the protocol selects two non-disjoint paths. If two non-disjoint paths are not available, only a primary path is selected with no backup. MoFRR is supported for both IPv4 and IPv6 protocol families.

Figure 121 on page 830 shows two paths from the multicast receiver routing device to the multicast source routing device.

Figure 121: MoFRR Sample Topology



When enabled with MoFRR functionality, the multicast receiver routing device sets up two multicast trees, a primary path and a backup path, toward the multicast source for each (S,G). In other words, the multicast receiver routing device propagates the same (S,G) join messages toward two different upstream neighbors, thus creating two multicast trees.

One of the multicast trees goes through plane 1 and the other through plane 2, as shown in Figure 121 on page 830. For each (S,G), the multicast receiver routing device forwards traffic received on the primary path and drops traffic received on the backup path.

MoFRR is supported on both equal-cost multipath (ECMP) paths and non-ECMP paths. Unicast loop-free alternate (LFA) routes need to be enabled to support MoFRR on non-ECMP paths. LFA routes are enabled with the **link-protection** statement in the interior gateway protocol (IGP) configuration. When you enable link protection on an OSPF or IS-IS interface, Junos OS creates a backup LFA path to the primary next hop for all destination routes that traverse the protected interface.

PIM Functionality

Junos OS supports MoFRR for shortest-path tree (SPT) joins in PIM source-specific multicast (SSM) and any-source multicast (ASM). MoFRR is supported for both SSM and ASM ranges. To enable MoFRR for (*,G) joins, include the mofrr-asm-starg configuration statement. For each group G, either (S,G) or (*,G) (not both) will undergo MoFRR. (S,G) always takes precedence over (*,G).

With MoFRR enabled, a PIM routing device propagates join messages on two upstream RPF interfaces to receive multicast traffic on both links for the same join request. Preference is given to two paths that do not converge to the same immediate upstream routing device. PIM installs appropriate multicast routes with upstream RPF next hops with two (primary and backup) interfaces.

When the primary path fails, the backup path is upgraded to primary, and traffic is forwarded accordingly. If there are alternate paths available, a new backup path is calculated and the appropriate multicast route is updated or installed.

MoFRR can be enabled along with PIM join load balancing (with the **join-load-balance automatic** statement). However, in such cases the distribution of join messages among the links might not be even. When a new ECMP link is added, join messages on the primary path are redistributed and load-balanced. The join messages on the backup path might still follow the same path and might not be evenly redistributed.

MoFRR is enabled using the **[edit routing-options multicast]** **stream-protection** configuration statement and is managed by a set of filter policies. When a PIM routing device receives a join message or an IGMP report, the device checks for the MoFRR configuration.

If the MoFRR configuration is not present, PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 121 on page 830).

If the MoFRR configuration is present, Junos OS checks for a policy configuration.

If a policy is not present, Junos OS checks for primary and backup paths (upstream interfaces), and takes the following actions:

- If primary and backup paths are not available—PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 121 on page 830).
- If primary and backup paths are available—PIM sends the join message upstream toward two of the available upstream neighbors. Junos OS sets up primary and secondary multicast paths to receive multicast traffic (for example, plane 1 in Figure 121 on page 830).

If a policy is present, Junos OS checks whether the policy allows MoFRR for this (S,G), and takes the following actions:

- If the policy check fails—PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 121 on page 830).
- If the policy check passes—Junos OS checks for primary and backup paths (upstream interfaces).
  - If the primary and backup paths are not available, PIM sends a join message upstream toward one upstream neighbor (for example, plane 2 in Figure 121 on page 830).
  - If the primary and backup paths are available, PIM sends the join message upstream toward two of the available upstream neighbors. Junos OS sets up primary and secondary multicast paths to receive multicast traffic (for example, plane 1 in Figure 121 on page 830).

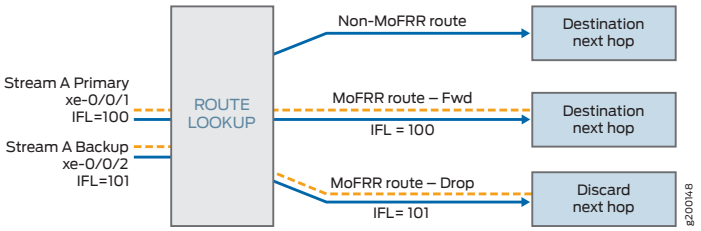
Packet Forwarding

Multicast source stream selection is performed at the incoming interface. This prevents duplicate streams from being sent across the fabric and prevents multiple route lookups that result in drops, thus preserving fabric bandwidth and maximizing forwarding performance.

For PIM, each IP multicast stream contains the same destination address. Regardless of the interface on which the packets arrive, the packets have the same route. Junos OS checks the interface upon which each packet arrives and forwards only those that are from the primary interface. If the interface matches a backup stream interface, the packets are dropped. If no match is found, the packets are handled as exceptions in the control plane.

This process in shown in Figure 122 on page 832.

Figure 122: MoFRR IP Route Handling in the Packet Forwarding Engine



Limitations and Caveats

MoFRR has the following limitations and caveats on switches:

- MoFRR failure detection is supported for immediate link protection of the multicast routing device on which MoFRR is enabled and not on all the links (end-to-end) in the multicast traffic path.
- MoFRR supports FRR on two selected disjoint paths toward the source. Two of the selected upstream neighbors cannot be on the same interface—in other words, two upstream neighbors on a LAN segment.
- Detection of the maximum end-to-end disjoint upstream paths is not supported. The multicast receiver routing device only makes sure that there is a disjoint upstream node (the immediate previous hop). PIM does not support the equivalent of explicit route objects (EROs). Hence, disjoint upstream path detection is limited to control over the immediately previous hop node. Because of this limitation, the path to the upstream node of the previous hop selected as primary and backup might be shared.
- Some traffic loss is seen in the following scenarios:
  - A better upstream path becomes available on an egress node.
  - MoFRR is enabled or disabled on the egress node while there is an active traffic stream flowing.

- PIM join load balancing for join messages for backup paths is not supported.
- For a multicast group G, MoFRR is not allowed for both (S,G) and (*,G) join messages. (S,G) join messages have precedence over (*,G).
- MoFRR is not supported for multicast traffic streams that use two different multicast groups. Each (S,G) combination is treated as a unique multicast traffic stream.
- The bidirectional PIM range is not supported for MoFRR.
- PIM dense-mode is not supported for MoFRR
- MoFRR is not supported when the upstream interface is an integrated routing and bridging (IRB) interface, which impacts other multicast features such as internet Group Management Protocol version 3 (IGMPv3) snooping.
- Multicast statistics for the backup traffic stream are not maintained by PIM and therefore are not available in the operational output of **show** commands.
- Rate monitoring is not supported.
- Packet replication and multicast lookups while forwarding multicast traffic can cause packets to recirculate through PFEs multiple times. As a result, displayed values for multicast packet counts from the **show pfe statistics traffic** command might show higher numbers than expected in output fields such as **Input packets** and **Output packets**. In an MoFRR scenario with increased traffic flow due to both primary and backup streams, increased packet counts due to this behavior might be more noticeable.

Release History Table

Release	Description
17.4R1	Starting in Junos OS Release 17.4R1, QFX Series switches support multicast-only fast reroute (MoFRR) functionality, which minimizes packet loss for traffic in a multicast distribution tree when link failures occur.

Related Documentation

- Configuring Multicast-Only Fast Reroute on page 834
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844

Configuring Multicast-Only Fast Reroute

You can configure multicast-only fast reroute (MoFRR) to minimize packet loss in a network when there is a link failure.

When fast reroute is applied to unicast streams, an upstream router preestablishes MPLS label-switched paths (LSPs) or precomputes an IP loop-free alternate (LFA) fast reroute backup path to handle failure of a segment in the downstream path.

In multicast routing, the traffic distribution graphs are usually originated by the receiver. This is unlike unicast routing, which usually establishes the path from the source to the receiver. Protocols that are capable of establishing multicast distribution graphs are PIM (for IP), multipoint LDP (for MPLS) and RSVP-TE (for MPLS). Of these, PIM and multipoint LDP receivers initiate the distribution graph setup, and therefore:

- On the QFX series, MoFRR is supported in PIM domains.
- On the MX Series and SRX Series, MoFRR is supported in PIM and multipoint LDP domains.

The configuration steps are the same for enabling MoFRR for PIM on all devices that support this feature, unless otherwise indicated. Configuration steps that are not applicable to multipoint LDP MoFRR are also indicated.

(For MX Series routers only) MoFRR is supported on MX Series routers with MPC line cards. As a prerequisite,all the line cards in the router must be MPCs.

To configure MoFRR on routers or switches:

1. (For MX Series and SRX Series routers only) Set the router to enhanced IP mode.

   ```
   [edit chassis]
   user@host# set network-services enhanced-ip
   ```

2. Enable MoFRR.

   ```
   [edit routing-options multicast]
   user@host# set stream-protection
   ```

3. (Optional) Configure a routing policy that filters for a restricted set of multicast streams to be affected by your MoFRR configuration.

   You can apply filters that are based on source or group addresses.

   For example:

   ```
   [edit policy-options]
   policy-statement mofrr-select {
     term A {
       from {
         source-address-filter 225.1.1.1/32 exact;
       }
       then {
         accept;
   ```

```
    }
  }
  term B {
    from {
      source-address-filter 226.0.0.0/8 orlonger;
    }
    then {
      accept;
    }
  }
  term C {
    from {
      source-address-filter 227.1.1.0/24 orlonger;
      source-address-filter 227.4.1.0/24 orlonger;
      source-address-filter 227.16.1.0/24 orlonger;
    }
    then {
      accept;
    }
  }
  term D {
    from {
      source-address-filter 227.1.1.1/32 exact
    }
    then {
      reject; #MoFRR disabled
    }
  }
  ...
}
```

4. (Optional) If you configured a routing policy to filter the set of multicast groups to be affected by your MoFRR configuration, apply the policy for MoFRR stream protection.

   ```
   [edit routing-options multicast stream-protection]
   user@host# set policy policy-name
   ```

   For example:

   ```
   routing-options {
     multicast {
       stream-protection {
         policy mofrr-select
       }
     }
   }
   ```

5. (Optional) In a PIM domain with MoFRR, allow MoFRR to be applied to any-source multicast (ASM) (*,G) joins.

   This is not supported for multipoint LDP MoFRR.

   ```
   [edit routing-options multicast stream-protection]
   user@host# set mofrr-asm-starg
   ```

6. (Optional) In a PIM domain with MoFRR, allow only a disjoint RPF (an RPF on a separate plane) to be selected as the backup RPF path.

   This is not supported for multipoint LDP MoFRR. In a multipoint LDP MoFRR domain, the same label is shared between parallel links to the same upstream neighbor. This is not the case in a PIM domain, where each link forms a neighbor. The **mofrr-disjoint-upstream-only** statement does not allow a backup RPF path to be selected if the path goes to the same upstream neighbor as that of the primary RPF path. This ensures that MoFRR is triggered only on a topology that has multiple RPF upstream neighbors.

   ```
   [edit routing-options multicast stream-protection]
   user@host# set mofrr-disjoint-upstream-only
   ```

7. (Optional) In a PIM domain with MoFRR, prevent sending join messages on the backup path, but retain all other MoFRR functionality.

   This is not supported for multipoint LDP MoFRR.

   ```
   [edit routing-options multicast stream-protection]
   user@host# set mofrr-no-backup-join
   ```

8. (Optional) In a PIM domain with MoFRR, allow new primary path selection to be based on the unicast gateway selection for the unicast route to the source and to change when there is a change in the unicast selection, rather than having the backup path be promoted as primary. This ensures that the primary RPF hop is always on the best path.

   When you include the **mofrr-primary-selection-by-routing** statement, the backup path is not guaranteed to get promoted to be the new primary path when the primary path goes down.

   This is not supported for multipoint LDP MoFRR.

   ```
   [edit routing-options multicast stream-protection]
   user@host# set mofrr-primary-path-selection-by-routing
   ```

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

Example: Configuring Multicast-Only Fast Reroute in a PIM Domain

This example shows how to configure multicast-only fast reroute (MoFRR) to minimize packet loss in a network when there is a link failure. It works by enhancing the multicast routing protocol, Protocol Independent Multicast (PIM).

MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path. Data packets are received from both the primary path and the backup paths. The redundant packets are discarded at topology merge points , based on priority (weights assigned to primary and backup paths). When a failure is detected on the primary path, the repair is made by changing the interface on which packets are accepted to the secondary interface. Because the repair is local, it is fast—greatly improving convergence times in the event of a link failure on the primary path.

Requirements

No special configuration beyond device initialization is required before configuring this example.

In this example, only the egress provider edge (PE) router has MoFRR enabled,MoFRR in a PIM domain can be enabled on any of the routers.

MoFRR is supported on MX Series platforms with MPC line cards. As a prerequisite, the router must be set to **network-services enhanced-ip** mode, and all the line-cards in the platform must be MPCs.

This example requires Junos OS Release 14.1 or later on the egress PE router.

Overview

In this example, Device R3 is the egress edge router. MoFRR is enabled on this device only.

OSPF or IS-IS is used for connectivity, though any interior gateway protocol (IGP) or static routes can be used.

PIM sparse mode version 2 is enabled on all devices in the PIM domain. Device R1 serves as the rendezvous point (RP).

Device R3, in addition to MoFRR, also has PIM join load balancing enabled.

For testing purposes, routers are used to simulate the source and the receiver. Device R3 is configured to statically join the desired group by using the **set protocols igmp interface**

fe-1/2/15.0 static group 225.1.1.1 command. It is just joining, not listening. The fe-1/2/15.0 interface is the Device R3 interface facing the receiver. In the case when a real multicast receiver host is not available, as in this example, this static IGMP configuration is useful. On the receiver, to make it listen to the multicast group address, this example uses **set protocols sap listen 225.1.1.1**. To make the source send multicast traffic, a multicast ping is issued from the source router. The ping command is **ping 225.1.1.1 bypass-routing interface fe-1/2/10.0 ttl 10 count 1000000000**. The fe-1/2/10.0 interface is the source interface facing Device R1.

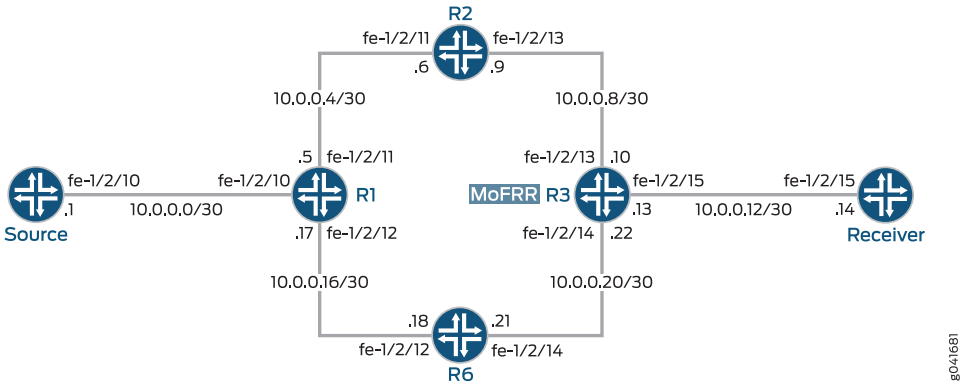
MoFRR configuration includes multiple options that are not shown in this example, but are explained separately. The options are as follows:

```
stream-protection {
    mofrr-asm-starg;
    mofrr-disjoint-upstream-only;
    mofrr-no-backup-join;
    mofrr-primary-path-selection-by-routing;
    policy policy-name;
}
```

Topology

Figure 123 on page 838 shows the sample network.

Figure 123: MoFRR in a PIM Domain



"CLI Quick Configuration" on page 838 shows the configuration for all of the devices in Figure 123 on page 838.

The section "Step-by-Step Configuration" on page 840 describes the steps on Device R3.

CLI Quick Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces fe-1/2/10 unit 0 family inet address 10.0.0.2/30
set interfaces fe-1/2/11 unit 0 family inet address 10.0.0.5/30
set interfaces fe-1/2/12 unit 0 family inet address 10.0.0.17/30
```

```
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols ospf area 0.0.0.0 interface fe-1/2/10.0
set protocols ospf area 0.0.0.0 interface fe-1/2/11.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/12.0
set protocols pim rp local family inet address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

Device R2

```
set interfaces fe-1/2/11 unit 0 family inet address 10.0.0.6/30
set interfaces fe-1/2/13 unit 0 family inet address 10.0.0.9/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols ospf area 0.0.0.0 interface fe-1/2/11.0
set protocols ospf area 0.0.0.0 interface fe-1/2/13.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

Device R3

```
set chassis network-services enhanced-ip
set interfaces fe-1/2/13 unit 0 family inet address 10.0.0.10/30
set interfaces fe-1/2/15 unit 0 family inet address 10.0.0.13/30
set interfaces fe-1/2/14 unit 0 family inet address 10.0.0.22/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols igmp interface fe-1/2/15.0 static group 225.1.1.1
set protocols ospf area 0.0.0.0 interface fe-1/2/13.0
set protocols ospf area 0.0.0.0 interface fe-1/2/15.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface fe-1/2/14.0
set protocols pim rp static address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim join-load-balance automatic
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-options forwarding-table export load-balancing-policy
set routing-options multicast stream-protection
```

Device R6

```
set interfaces fe-1/2/12 unit 0 family inet address 10.0.0.18/30
set interfaces fe-1/2/14 unit 0 family inet address 10.0.0.21/30
set interfaces lo0 unit 0 family inet address 192.168.0.6/32
set protocols ospf area 0.0.0.0 interface fe-1/2/12.0
set protocols ospf area 0.0.0.0 interface fe-1/2/14.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

Device Source

```
set interfaces fe-1/2/10 unit 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.4/32
set protocols ospf area 0.0.0.0 interface fe-1/2/10.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device Receiver

```
set interfaces fe-1/2/15 unit 0 family inet address 10.0.0.14/30
set interfaces lo0 unit 0 family inet address 192.168.0.5/32
set protocols sap listen 225.1.1.1
set protocols ospf area 0.0.0.0 interface fe-1/2/15.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Step-by-Step Configuration

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Device R3:

1. Enable enhanced IP mode.

   ```
   [edit chassis]
   user@R3# set network-services enhanced-ip
   ```

2. Configure the device interfaces.

   ```
   [edit interfaces]
   user@R3# set fe-1/2/13 unit 0 family inet address 10.0.0.10/30
   user@R3# set fe-1/2/15 unit 0 family inet address 10.0.0.13/30
   user@R3# set fe-1/2/14 unit 0 family inet address 10.0.0.22/30
   user@R3# set lo0 unit 0 family inet address 192.168.0.3/32
   ```

3. For testing purposes only, on the interface facing Device Receiver, simulate IGMP joins.

   If your test environment has receiver hosts, this step is not necessary.

   ```
   [edit protocols igmp interface fe-1/2/15.0]
   user@R3# set static group 225.1.1.1
   ```

4. Configure an IGP or static routes.

   ```
   [edit protocols ospf area 0.0.0.0]
   user@R3# set interface fe-1/2/13.0
   user@R3# set interface fe-1/2/15.0
   user@R3# set interface lo0.0 passive
   user@R3# set interface fe-1/2/14.0
   ```

5. Configure PIM.

   ```
   [edit protocols pim]
   user@R3# set rp static address 192.168.0.1
   user@R3# set interface all mode sparse
   user@R3# set interface all version 2
   ```

6. (Optional) Configure PIM join load balancing.

   ```
   [edit protocols pim]
   user@R3# set join-load-balance automatic
   ```

7. (Optional) Configure per-packet load balancing.

```
[edit policy-options policy-statement load-balancing-policy]
user@R3# set then load-balance per-packet

[edit routing-options forwarding-table]
user@R3# set export load-balancing-policy
```

8. Enable MoFRR.

```
[edit routing-options multicast]
user@R3# set stream-protection
```

Results From configuration mode, confirm your configuration by entering the **show chassis**, **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show chassis
network-services enhanced-ip;

user@R3# show interfaces
fe-1/2/13 {
    unit 0 {
        family inet {
            address 10.0.0.10/30;
        }
    }
}
fe-1/2/14 {
    unit 0 {
        family inet {
            address 10.0.0.22/30;
        }
    }
}
fe-1/2/15 {
    unit 0 {
        family inet {
            address 10.0.0.13/30;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.0.3/32;
        }
    }
}

user@R3# show protocols
igmp {
    interface fe-1/2/15.0 {
```

```
        static {
            group 225.1.1.1;
        }
    }
}
ospf {
    area 0.0.0.0 {
        interface fe-1/2/13.0;
        interface fe-1/2/15.0;
        interface lo0.0 {
            passive;
        }
        interface fe-1/2/14.0;
    }
}
pim {
    rp {
        static {
            address 192.168.0.1;
        }
    }
    interface all {
        mode sparse;
        version 2;
    }
    join-load-balance {
        automatic;
    }
}
```

```
user@R3# show policy-options
policy-statement load-balancing-policy {
    then {
        load-balance per-packet;
    }
}
```

```
user@R3# show routing-options
forwarding-table {
    export load-balancing-policy;
}
multicast {
    stream-protection;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

- Sending Multicast Traffic Into the PIM Domain on page 843
- Verifying the Upstream Interfaces on page 843
- Checking the Multicast Routes on page 843

Sending Multicast Traffic Into the PIM Domain

Purpose Use a multicast ping command to simulate multicast traffic.

Action

```
user@Source> ping 225.1.1.1 bypass-routing interface fe-1/2/10.0 ttl 10 count 1000000000

PING 225.1.1.1 (225.1.1.1): 56 data bytes
64 bytes from 10.0.0.14: icmp_seq=1 ttl=61 time=0.845 ms
64 bytes from 10.0.0.14: icmp_seq=2 ttl=61 time=0.661 ms
64 bytes from 10.0.0.14: icmp_seq=3 ttl=61 time=0.615 ms
64 bytes from 10.0.0.14: icmp_seq=4 ttl=61 time=0.640 ms
```

Meaning The interface on Device Source, facing Device R1, is fe-1/2/10.0. Keep in mind that multicast pings have a TTL of 1 by default, so you must use the **ttl** option.

Verifying the Upstream Interfaces

Purpose Make sure that the egress device has two upstream interfaces for the multicast group join.

Action

```
user@R3> show pim join 225.1.1.1 extensive sg
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 225.1.1.1
    Source: 10.0.0.1
    Flags: sparse,spt
    Active upstream interface: fe-1/2/13.0
    Active upstream neighbor: 10.0.0.9
    MoFRR Backup upstream interface: fe-1/2/14.0
    MoFRR Backup upstream neighbor: 10.0.0.21
    Upstream state: Join to Source, No Prune to RP
    Keepalive timeout: 354
    Uptime: 00:00:06
    Downstream neighbors:
        Interface: fe-1/2/15.0
            10.0.0.13 State: Join Flags: S   Timeout: Infinity
            Uptime: 00:00:06 Time since last Join: 00:00:06
    Number of downstream interfaces: 1
```

Meaning The output shows an active upstream interface and neighbor, and also an MoFRR backup upstream interface and neighbor.

Checking the Multicast Routes

Purpose Examine the IP multicast forwarding table to make sure that there is an upstream RPF interface list, with a primary and a backup interface.

Action

```
user@R3> show multicast route extensive

Instance: master Family: INET

Group: 225.1.1.1
    Source: 10.0.0.1/32
    Upstream rpf interface list:
        fe-1/2/13.0 (P) fe-1/2/14.0 (B)
    Downstream interface list:
        fe-1/2/15.0
    Session description: Unknown
    Forwarding statistics are not available
    RPF Next-hop ID: 836
    Next-hop ID: 1048585
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 171 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:03:09
```

Meaning The output shows an upstream RPF interface list, with a primary and a backup interface.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Configuring Multicast-Only Fast Reroute on page 834
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches

This example shows how to configure multicast-only fast reroute (MoFRR) to minimize packet loss in a network when there is a link failure. It works by enhancing the multicast routing protocol, Protocol Independent Multicast (PIM).

MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path. Data packets are received from both the primary path and the backup paths. The redundant packets are discarded at topology merge points, based on priority (weights assigned to primary and backup paths). When a failure is detected on the primary path, the repair is made by changing the interface on which packets are accepted to the secondary interface. Because the repair is local, it is fast—greatly improving convergence times in the event of a link failure on the primary path.

Requirements

No special configuration beyond device initialization is required before configuring this example.

This example uses QFX Series switches, and only the egress provider edge (PE) device has MoFRR enabled. This topology might alternatively include MX Series routers for the other devices where MoFRR is not enabled; in that case, substitute the corresponding interfaces for MX Series device ports used for the primary or backup multicast traffic streams.

This example requires Junos OS Release 17.4R1 or later on the device running MoFRR.

Overview

In this example, Device R3 is the egress edge device. MoFRR is enabled on this device only.

OSPF or IS-IS is used for connectivity, though any interior gateway protocol (IGP) or static routes can be used.

PIM sparse mode version 2 is enabled on all devices in the PIM domain. Device R1 serves as the rendezvous point (RP).

Device R3, in addition to MoFRR, also has PIM join load balancing enabled.

For testing purposes, routing or switching devices are used to simulate the multicast source and the receiver. Device R3 is configured to statically join the desired group by using the **set protocols igmp interface xe-0/0/15.0 static group 225.1.1.1** command. It is just joining, not listening. The xe-0/0/15.0 interface is the Device R3 interface facing the receiver. In the case when a real multicast receiver host is not available, as in this example, this static IGMP configuration is useful. On the receiver, to listen to the multicast group address, this example uses **set protocols sap listen 225.1.1.1**. For the source to send multicast traffic, a multicast ping is issued from the source device. The ping command is **ping 225.1.1.1 bypass-routing interface xe-0/0/10.0 ttl 10 count 1000000000**. The xe-0/0/10.0 interface is the source interface facing Device R1.

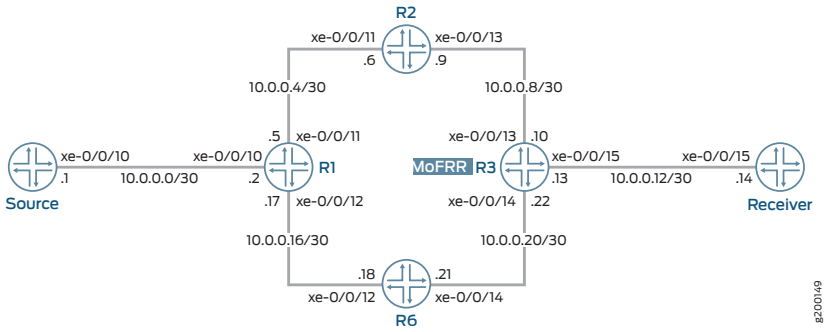
MoFRR configuration includes multiple options that are not shown in this example, but are explained separately. The options are as follows:

```
stream-protection {
    mofrr-asm-starg;
    mofrr-disjoint-upstream-only;
    mofrr-no-backup-join;
    mofrr-primary-path-selection-by-routing;
    policy policy-name;
}
```

Topology

Figure 124 on page 846 shows the sample network.

Figure 124: MoFRR in a PIM Domain



"CLI Quick Configuration" on page 838 shows the configuration for all of the devices in Figure 124 on page 846.

The section "Step-by-Step Configuration" on page 840 describes the steps on Device R3.

CLI Quick Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device R1

```
set interfaces xe-0/0/10 unit 0 family inet address 10.0.0.2/30
set interfaces xe-0/0/11 unit 0 family inet address 10.0.0.5/30
set interfaces xe-0/0/12 unit 0 family inet address 10.0.0.17/30
set interfaces lo0 unit 0 family inet address 192.168.0.1/32
set protocols ospf area 0.0.0.0 interface xe-0/0/10.0
set protocols ospf area 0.0.0.0 interface xe-0/0/11.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface xe-0/0/12.0
set protocols pim rp local family inet address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

Device R2

```
set interfaces xe-0/0/11 unit 0 family inet address 10.0.0.6/30
set interfaces xe-0/0/13 unit 0 family inet address 10.0.0.9/30
set interfaces lo0 unit 0 family inet address 192.168.0.2/32
set protocols ospf area 0.0.0.0 interface xe-0/0/11.0
set protocols ospf area 0.0.0.0 interface xe-0/0/13.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

Device R3

```
set interfaces xe-0/0/13 unit 0 family inet address 10.0.0.10/30
set interfaces xe-0/0/15 unit 0 family inet address 10.0.0.13/30
set interfaces xe-0/0/14 unit 0 family inet address 10.0.0.22/30
set interfaces lo0 unit 0 family inet address 192.168.0.3/32
set protocols igmp interface xe-0/0/15.0 static group 225.1.1.1
```

```
set protocols ospf area 0.0.0.0 interface xe-0/0/13.0
set protocols ospf area 0.0.0.0 interface xe-0/0/15.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface xe-0/0/14.0
set protocols pim rp static address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
set protocols pim join-load-balance automatic
set policy-options policy-statement load-balancing-policy then load-balance per-packet
set routing-options forwarding-table export load-balancing-policy
set routing-options multicast stream-protection
```

Device R6

```
set interfaces xe-0/0/12 unit 0 family inet address 10.0.0.18/30
set interfaces xe-0/0/14 unit 0 family inet address 10.0.0.21/30
set interfaces lo0 unit 0 family inet address 192.168.0.6/32
set protocols ospf area 0.0.0.0 interface xe-0/0/12.0
set protocols ospf area 0.0.0.0 interface xe-0/0/14.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 192.168.0.1
set protocols pim interface all mode sparse
set protocols pim interface all version 2
```

Device Source

```
set interfaces xe-0/0/10 unit 0 family inet address 10.0.0.1/30
set interfaces lo0 unit 0 family inet address 192.168.0.4/32
set protocols ospf area 0.0.0.0 interface xe-0/0/10.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device Receiver

```
set interfaces xe-0/0/15 unit 0 family inet address 10.0.0.14/30
set interfaces lo0 unit 0 family inet address 192.168.0.5/32
set protocols sap listen 225.1.1.1
set protocols ospf area 0.0.0.0 interface xe-0/0/15.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Step-by-Step Configuration

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Device R3:

1. Configure the device interfaces.

```
[edit interfaces]
user@R3# set xe-0/0/13 unit 0 family inet address 10.0.0.10/30
user@R3# set xe-0/0/15 unit 0 family inet address 10.0.0.13/30
user@R3# set xe-0/0/14 unit 0 family inet address 10.0.0.22/30
user@R3# set lo0 unit 0 family inet address 192.168.0.3/32
```

2. For testing purposes only, on the interface facing the device labeled Receiver, simulate IGMP joins.

   If your test environment has receiver hosts, this step is not necessary.

```
[edit protocols igmp interface xe-0/0/15.0]
user@R3# set static group 225.1.1.1
```

3. Configure IGP or static routes.

```
[edit protocols ospf area 0.0.0.0]
user@R3# set interface xe-0/0/13.0
user@R3# set interface xe-0/0/15.0
user@R3# set interface lo0.0 passive
user@R3# set interface xe-0/0/14.0
```

4. Configure PIM.

```
[edit protocols pim]
user@R3# set rp static address 192.168.0.1
user@R3# set interface all mode sparse
user@R3# set interface all version 2
```

5. (Optional) Configure PIM join load balancing.

```
[edit protocols pim]
user@R3# set join-load-balance automatic
```

6. (Optional) Configure per-packet load balancing.

```
[edit policy-options policy-statement load-balancing-policy]
user@R3# set then load-balance per-packet
```

```
[edit routing-options forwarding-table]
user@R3# set export load-balancing-policy
```

7. Enable MoFRR.

```
[edit routing-options multicast]
user@R3# set stream-protection
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show interfaces
xe-0/0/13 {
    unit 0 {
        family inet {
            address 10.0.0.10/30;
        }
    }
}
xe-0/0/14 {
    unit 0 {
        family inet {
```

```
        address 10.0.0.22/30;
      }
    }
  }
  xe-0/0/15 {
    unit 0 {
      family inet {
        address 10.0.0.13/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 192.168.0.3/32;
      }
    }
  }
```

```
user@R3# show protocols
igmp {
  interface xe-0/0/15.0 {
    static {
      group 225.1.1.1;
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface xe-0/0/13.0;
    interface xe-0/0/15.0;
    interface lo0.0 {
      passive;
    }
    interface xe-0/0/14.0;
  }
}
pim {
  rp {
    static {
      address 192.168.0.1;
    }
  }
  interface all {
    mode sparse;
    version 2;
  }
  join-load-balance {
    automatic;
  }
}
```

```
user@R3# show policy-options
policy-statement load-balancing-policy {
  then {
    load-balance per-packet;
  }
```

```
}
```

```
user@R3# show routing-options
forwarding-table {
  export load-balancing-policy;
}
multicast {
  stream-protection;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Sending Multicast Traffic Into the PIM Domain

Purpose Use a multicast ping command to simulate multicast traffic.

Action

```
user@Source> ping 225.1.1.1 bypass-routing interface xe-0/0/10.0 ttl 10 count 1000000000

PING 225.1.1.1 (225.1.1.1): 56 data bytes
64 bytes from 10.0.0.14: icmp_seq=1 ttl=61 time=0.845 ms
64 bytes from 10.0.0.14: icmp_seq=2 ttl=61 time=0.661 ms
64 bytes from 10.0.0.14: icmp_seq=3 ttl=61 time=0.615 ms
64 bytes from 10.0.0.14: icmp_seq=4 ttl=61 time=0.640 ms
```

Meaning The interface on Device Source, facing Device R1, is xe-0/0/10.0. Keep in mind that multicast pings have a TTL of 1 by default, so you must use the **ttl** option.

Verifying the Upstream Interfaces

Purpose Make sure that the egress device has two upstream interfaces for the multicast group join.

Action

```
user@R3> show pim join 225.1.1.1 extensive sg
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 225.1.1.1
    Source: 10.0.0.1
    Flags: sparse,spt
    Active upstream interface: xe-0/0/13.0
    Active upstream neighbor: 10.0.0.9
    MoFRR Backup upstream interface: xe-0/0/14.0
    MoFRR Backup upstream neighbor: 10.0.0.21
    Upstream state: Join to Source, No Prune to RP
    Keepalive timeout: 354
    Uptime: 00:00:06
    Downstream neighbors:
        Interface: xe-0/0/15.0
            10.0.0.13 State: Join Flags: S   Timeout: Infinity
            Uptime: 00:00:06 Time since last Join: 00:00:06
    Number of downstream interfaces: 1
```

Meaning The output shows an active upstream interface and neighbor, and also an MoFRR backup upstream interface and neighbor.

Checking the Multicast Routes

Purpose Examine the IP multicast forwarding table to make sure that there is an upstream RPF interface list, with a primary and a backup interface.

Action

```
user@R3> show multicast route extensive

Instance: master Family: INET

Group: 225.1.1.1
    Source: 10.0.0.1/32
    Upstream rpf interface list:
        xe-0/0/13.0 (P) xe-0/0/14.0 (B)
    Downstream interface list:
        xe-0/0/15.0
    Session description: Unknown
    Forwarding statistics are not available
    RPF Next-hop ID: 836
    Next-hop ID: 1048585
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 171 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:03:09
```

Meaning The output shows an upstream RPF interface list, with a primary and a backup interface.

Related Documentation

- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Configuring Multicast-Only Fast Reroute on page 834

Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain

This example shows how to configure multicast-only fast reroute (MoFRR) to minimize packet loss in a network when there is a link failure.

Multipoint LDP MoFRR is used at the egress node of an MPLS network, where the packets are forwarded to an IP network. In the case of multipoint LDP MoFRR, the two paths toward the upstream provider edge (PE) router are established for receiving two streams of MPLS packets at the label-edge router (LER). One of the streams (the primary) is accepted, and the other one (the backup) is dropped at the LER. The backup stream is accepted if the primary path fails.

Requirements

No special configuration beyond device initialization is required before configuring this example.

In a multipoint LDP domain, for MoFRR to work, only the egress PE router needs to have MoFRR enabled. The other routers do not need to support MoFRR.

MoFRR is supported on MX Series platforms with MPC line cards. As a prerequisite, the router must be set to **network-services enhanced-ip** mode, and all the line-cards in the platform must be MPCs.

This example requires Junos OS Release 14.1 or later on the egress PE router.

Overview

In this example, Device R3 is the egress edge router. MoFRR is enabled on this device only.

OSPF is used for connectivity, though any interior gateway protocol (IGP) or static routes can be used.

For testing purposes, routers are used to simulate the source and the receiver. Device R4 and Device R8 are configured to statically join the desired group by using the **set protocols igmp interface interface-name static group *group*** command. In the case when a real multicast receiver host is not available, as in this example, this static IGMP configuration is useful. On the receivers, to make them listen to the multicast group address, this example uses **set protocols sap listen *group***.

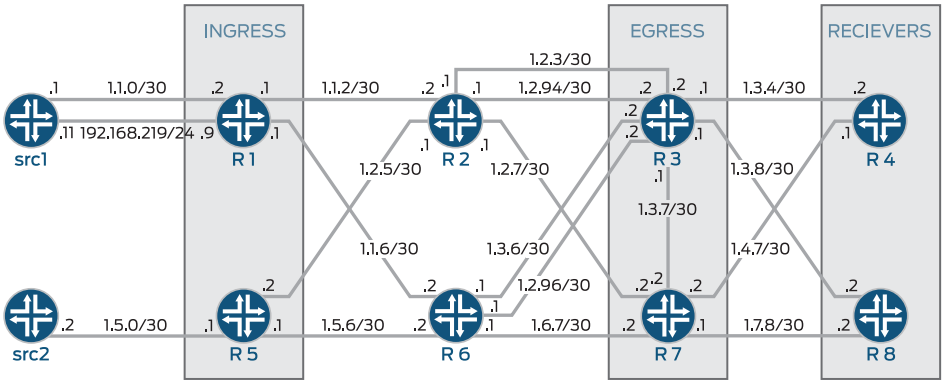
MoFRR configuration includes a policy option that is not shown in this example, but is explained separately. The option is configured as follows:

```
stream-protection {
  policy policy-name;
}
```

Topology

Figure 125 on page 853 shows the sample network.

Figure 125: MoFRR in a Multipoint LDP Domain



lo0:			
src1	1.1.1.17	R4	1.1.1.4
src2	1.1.1.18	R5	1.1.1.5
R1	1.1.1.1	R6	1.1.1.6
R2	1.1.1.2	R7	1.1.1.7
R3	1.1.1.3	R8	1.1.1.8

src1	ge-1/2/10	>	ge-1/2/10	R1	R3	ge-1/2/19	>	ge-1/2/19	R6
src1	ge-1/2/11	>	ge-1/2/11	R1	R3	ge-1/2/20	>	ge-1/2/20	R6
R1	ge-1/2/12	>	ge-1/2/12	R2	R3	ge-1/2/21	>	ge-1/2/21	R7
R1	ge-1/2/13	>	ge-1/2/13	R6	R3	ge-1/2/22	>	ge-1/2/22	R8
R2	ge-1/2/14	>	ge-1/2/14	R3	R4	ge-1/2/23	>	ge-1/2/23	R7
R2	ge-1/2/15	>	ge-1/2/15	R3	src2	ge-1/2/24	>	ge-1/2/24	R5
R2	ge-1/2/16	>	ge-1/2/16	R5	R5	ge-1/2/25	>	ge-1/2/25	R6
R2	ge-1/2/17	>	ge-1/2/17	R7	R6	ge-1/2/26	>	ge-1/2/26	R7
R3	ge-1/2/18	>	ge-1/2/18	R4	R7	ge-1/2/27	>	ge-1/2/27	R8

"CLI Quick Configuration" on page 853 shows the configuration for all of the devices in Figure 125 on page 853.

The section "Configuration" on page 859 describes the steps on Device R3.

CLI Quick Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Device src1

```
set interfaces ge-1/2/10 unit 0 description src1-to-R1
set interfaces ge-1/2/10 unit 0 family inet address 1.1.0.1/30
set interfaces ge-1/2/11 unit 0 description src1-to-R1
set interfaces ge-1/2/11 unit 0 family inet address 192.168.219.11/24
set interfaces lo0 unit 0 family inet address 1.1.1.17/32
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device src2

```
set interfaces ge-1/2/24 unit 0 description src2-to-R5
```

```
set interfaces ge-1/2/24 unit 0 family inet address 1.5.0.2/30
set interfaces lo0 unit 0 family inet address 1.1.1.18/32
set protocols rsvp interface all
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
```

Device R1

```
set interfaces ge-1/2/12 unit 0 description R1-to-R2
set interfaces ge-1/2/12 unit 0 family inet address 1.1.2.1/30
set interfaces ge-1/2/12 unit 0 family mpls
set interfaces ge-1/2/13 unit 0 description R1-to-R6
set interfaces ge-1/2/13 unit 0 family inet address 1.1.6.1/30
set interfaces ge-1/2/13 unit 0 family mpls
set interfaces ge-1/2/10 unit 0 description R1-to-src1
set interfaces ge-1/2/10 unit 0 family inet address 1.1.0.2/30
set interfaces ge-1/2/11 unit 0 description R1-to-src1
set interfaces ge-1/2/11 unit 0 family inet address 192.168.219.9/30
set interfaces lo0 unit 0 family inet address 1.1.1.1/32
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 1.1.1.1
set protocols bgp group ibgp export static-route-tobgp
set protocols bgp group ibgp peer-as 10
set protocols bgp group ibgp neighbor 1.1.1.3
set protocols bgp group ibgp neighbor 1.1.1.7
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-1/2/12.0
set protocols ldp interface ge-1/2/13.0
set protocols ldp interface lo0.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim rp static address 1.1.1.5
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/10.0
set protocols pim interface ge-1/2/11.0
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address
  1.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter
  1.1.1.7/32 orlonger
set policy-options policy-statement mldppim-ex term A from source-address-filter
  1.1.0.0/30 orlonger
set policy-options policy-statement mldppim-ex term A then accept
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set routing-options autonomous-system 10
```

Device R2

```
set interfaces ge-1/2/12 unit 0 description R2-to-R1
```

```
set interfaces ge-1/2/12 unit 0 family inet address 1.1.2.2/30
set interfaces ge-1/2/12 unit 0 family mpls
set interfaces ge-1/2/14 unit 0 description R2-to-R3
set interfaces ge-1/2/14 unit 0 family inet address 1.2.3.1/30
set interfaces ge-1/2/14 unit 0 family mpls
set interfaces ge-1/2/16 unit 0 description R2-to-R5
set interfaces ge-1/2/16 unit 0 family inet address 1.2.5.1/30
set interfaces ge-1/2/16 unit 0 family mpls
set interfaces ge-1/2/17 unit 0 description R2-to-R7
set interfaces ge-1/2/17 unit 0 family inet address 1.2.7.1/30
set interfaces ge-1/2/17 unit 0 family mpls
set interfaces ge-1/2/15 unit 0 description R2-to-R3
set interfaces ge-1/2/15 unit 0 family inet address 1.2.94.1/30
set interfaces ge-1/2/15 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.2/32
set interfaces lo0 unit 0 family mpls
set protocols rsvp interface all
set protocols mpls interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp p2mp
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address
  1.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 10
```

Device R3

```
set chassis network-services enhanced-ip
set interfaces ge-1/2/14 unit 0 description R3-to-R2
set interfaces ge-1/2/14 unit 0 family inet address 1.2.3.2/30
set interfaces ge-1/2/14 unit 0 family mpls
set interfaces ge-1/2/18 unit 0 description R3-to-R4
set interfaces ge-1/2/18 unit 0 family inet address 1.3.4.1/30
set interfaces ge-1/2/18 unit 0 family mpls
set interfaces ge-1/2/19 unit 0 description R3-to-R6
set interfaces ge-1/2/19 unit 0 family inet address 1.3.6.2/30
set interfaces ge-1/2/19 unit 0 family mpls
set interfaces ge-1/2/21 unit 0 description R3-to-R7
set interfaces ge-1/2/21 unit 0 family inet address 1.3.7.1/30
set interfaces ge-1/2/21 unit 0 family mpls
set interfaces ge-1/2/22 unit 0 description R3-to-R8
set interfaces ge-1/2/22 unit 0 family inet address 1.3.8.1/30
set interfaces ge-1/2/22 unit 0 family mpls
set interfaces ge-1/2/15 unit 0 description R3-to-R2
set interfaces ge-1/2/15 unit 0 family inet address 1.2.94.2/30
set interfaces ge-1/2/15 unit 0 family mpls
set interfaces ge-1/2/20 unit 0 description R3-to-R6
set interfaces ge-1/2/20 unit 0 family inet address 1.2.96.2/30
set interfaces ge-1/2/20 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.3/32 primary
```

```
set routing-options autonomous-system 10
set routing-options multicast stream-protection
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 1.1.1.3
set protocols bgp group ibgp peer-as 10
set protocols bgp group ibgp neighbor 1.1.1.1
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface fxp0.0 disable
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/18.0
set protocols pim interface ge-1/2/22.0
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter
  1.1.0.1/30 orlonger
set policy-options policy-statement mldppim-ex term A then accept
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
```

Device R4

```
set interfaces ge-1/2/18 unit 0 description R4-to-R3
set interfaces ge-1/2/18 unit 0 family inet address 1.3.4.2/30
set interfaces ge-1/2/18 unit 0 family mpls
set interfaces ge-1/2/23 unit 0 description R4-to-R7
set interfaces ge-1/2/23 unit 0 family inet address 1.4.7.1/30
set interfaces lo0 unit 0 family inet address 1.1.1.4/32
set protocols igmp interface ge-1/2/18.0 version 3
set protocols igmp interface ge-1/2/18.0 static group 232.1.1.1 group-count 2
set protocols igmp interface ge-1/2/18.0 static group 232.1.1.1 source 192.168.219.11
set protocols igmp interface ge-1/2/18.0 static group 232.2.2.2 source 1.2.7.7
set protocols sap listen 232.1.1.1
set protocols sap listen 232.2.2.2
set protocols rsvp interface all
set protocols mpls interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface ge-1/2/23.0
set protocols pim interface ge-1/2/18.0
set protocols pim interface lo0.0
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
```

```
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address
  1.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 10
```

Device R5

```
set interfaces ge-1/2/24 unit 0 description R5-to-src2
set interfaces ge-1/2/24 unit 0 family inet address 1.5.0.1/30
set interfaces ge-1/2/16 unit 0 description R5-to-R2
set interfaces ge-1/2/16 unit 0 family inet address 1.2.5.2/30
set interfaces ge-1/2/16 unit 0 family mpls
set interfaces ge-1/2/25 unit 0 description R5-to-R6
set interfaces ge-1/2/25 unit 0 family inet address 1.5.6.1/30
set interfaces ge-1/2/25 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.5/32
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 1.1.1.5
set protocols bgp group ibgp export static-route-tobgp
set protocols bgp group ibgp peer-as 10
set protocols bgp group ibgp neighbor 1.1.1.7
set protocols bgp group ibgp neighbor 1.1.1.3
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-1/2/16.0
set protocols ldp interface ge-1/2/25.0
set protocols ldp p2mp
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/24.0
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set routing-options autonomous-system 10
```

Device R6

```
set interfaces ge-1/2/13 unit 0 description R6-to-R1
set interfaces ge-1/2/13 unit 0 family inet address 1.1.6.2/30
set interfaces ge-1/2/13 unit 0 family mpls
set interfaces ge-1/2/19 unit 0 description R6-to-R3
set interfaces ge-1/2/19 unit 0 family inet address 1.3.6.1/30
set interfaces ge-1/2/19 unit 0 family mpls
set interfaces ge-1/2/25 unit 0 description R6-to-R5
set interfaces ge-1/2/25 unit 0 family inet address 1.5.6.2/30
set interfaces ge-1/2/25 unit 0 family mpls
set interfaces ge-1/2/26 unit 0 description R6-to-R7
set interfaces ge-1/2/26 unit 0 family inet address 1.6.7.1/30
set interfaces ge-1/2/26 unit 0 family mpls
set interfaces ge-1/2/20 unit 0 description R6-to-R3
set interfaces ge-1/2/20 unit 0 family inet address 1.2.96.1/30
set interfaces ge-1/2/20 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.6/30
```

```
set protocols rsvp interface all
set protocols mpls interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface all
set protocols ldp p2mp
```

Device R7

```
set interfaces ge-1/2/17 unit 0 description R7-to-R2
set interfaces ge-1/2/17 unit 0 family inet address 1.2.7.2/30
set interfaces ge-1/2/17 unit 0 family mpls
set interfaces ge-1/2/21 unit 0 description R7-to-R3
set interfaces ge-1/2/21 unit 0 family inet address 1.3.7.2/30
set interfaces ge-1/2/21 unit 0 family mpls
set interfaces ge-1/2/23 unit 0 description R7-to-R4
set interfaces ge-1/2/23 unit 0 family inet address 1.4.7.2/30
set interfaces ge-1/2/23 unit 0 family mpls
set interfaces ge-1/2/26 unit 0 description R7-to-R6
set interfaces ge-1/2/26 unit 0 family inet address 1.6.7.2/30
set interfaces ge-1/2/26 unit 0 family mpls
set interfaces ge-1/2/27 unit 0 description R7-to-R8
set interfaces ge-1/2/27 unit 0 family inet address 1.7.8.1/30
set interfaces ge-1/2/27 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.7/32
set protocols rsvp interface all
set protocols mpls interface all
set protocols bgp group ibgp local-address 1.1.1.7
set protocols bgp group ibgp export static-route-tobgp
set protocols bgp group ibgp peer-as 10
set protocols bgp group ibgp neighbor 1.1.1.5
set protocols bgp group ibgp neighbor 1.1.1.1
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-1/2/17.0
set protocols ldp interface ge-1/2/21.0
set protocols ldp interface ge-1/2/26.0
set protocols ldp p2mp
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface lo0.0
set protocols pim interface ge-1/2/27.0
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then accept
set policy-options policy-statement mldppim-ex term A from source-address-filter
  1.1.0.1/30 orlonger
set policy-options policy-statement mldppim-ex term A then accept
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set routing-options autonomous-system 10
set routing-options multicast stream-protection policy mldppim-ex
```

Device R8

```
set interfaces ge-1/2/22 unit 0 description R8-to-R3
set interfaces ge-1/2/22 unit 0 family inet address 1.3.8.2/30
set interfaces ge-1/2/22 unit 0 family mpls
set interfaces ge-1/2/27 unit 0 description R8-to-R7
set interfaces ge-1/2/27 unit 0 family inet address 1.7.8.2/30
set interfaces ge-1/2/27 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 1.1.1.8/32
set protocols igmp interface ge-1/2/22.0 version 3
set protocols igmp interface ge-1/2/22.0 static group 232.1.1.1 group-count 2
set protocols igmp interface ge-1/2/22.0 static group 232.1.1.1 source 192.168.219.11
set protocols igmp interface ge-1/2/22.0 static group 232.2.2.2 source 1.2.7.7
set protocols sap listen 232.1.1.1
set protocols sap listen 232.2.2.2
set protocols rsvp interface all
set protocols ospf traffic-engineering
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim mldp-inband-signalling policy mldppim-ex
set protocols pim interface ge-1/2/27.0
set protocols pim interface ge-1/2/22.0
set protocols pim interface lo0.0
set policy-options policy-statement static-route-tobgp term static from protocol static
set policy-options policy-statement static-route-tobgp term static from protocol direct
set policy-options policy-statement static-route-tobgp term static then accept
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.0.0/24 orlonger
set policy-options policy-statement mldppim-ex term B from source-address-filter
  192.168.219.11/32 orlonger
set policy-options policy-statement mldppim-ex term B then p2mp-lsp-root address
  1.1.1.2
set policy-options policy-statement mldppim-ex term B then accept
set routing-options autonomous-system 10
```

Configuration

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure Device R3:

1. Enable enhanced IP mode.

```
[edit chassis]
user@R3# set network-services enhanced-ip
```

2. Configure the device interfaces.

```
[edit interfaces]
user@R3# set ge-1/2/14 unit 0 description R3-to-R2
user@R3# set ge-1/2/14 unit 0 family inet address 1.2.3.2/30
user@R3# set ge-1/2/14 unit 0 family mpls

user@R3# set ge-1/2/18 unit 0 description R3-to-R4
user@R3# set ge-1/2/18 unit 0 family inet address 1.3.4.1/30
```

```
user@R3# set ge-1/2/18 unit 0 family mpls

user@R3# set ge-1/2/19 unit 0 description R3-to-R6
user@R3# set ge-1/2/19 unit 0 family inet address 1.3.6.2/30
user@R3# set ge-1/2/19 unit 0 family mpls

user@R3# set ge-1/2/21 unit 0 description R3-to-R7
user@R3# set ge-1/2/21 unit 0 family inet address 1.3.7.1/30
user@R3# set ge-1/2/21 unit 0 family mpls

user@R3# set ge-1/2/22 unit 0 description R3-to-R8
user@R3# set ge-1/2/22 unit 0 family inet address 1.3.8.1/30
user@R3# set ge-1/2/22 unit 0 family mpls

user@R3# set ge-1/2/15 unit 0 description R3-to-R2
user@R3# set ge-1/2/15 unit 0 family inet address 1.2.94.2/30
user@R3# set ge-1/2/15 unit 0 family mpls

user@R3# set ge-1/2/20 unit 0 description R3-to-R6
user@R3# set ge-1/2/20 unit 0 family inet address 1.2.96.2/30
user@R3# set ge-1/2/20 unit 0 family mpls

user@R3# set lo0 unit 0 family inet address 1.1.1.3/32 primary
```

3. Configure the autonomous system (AS) number.

```
user@R3# set routing-options autonomous-system 10
```

4. Configure the routing policies.

```
[edit policy-options policy-statement mldppim-ex]
user@R3# set term B from source-address-filter 192.168.0.0/24 orlonger
user@R3# set term B from source-address-filter 192.168.219.11/32 orlonger
user@R3# set term B then accept
user@R3# set term A from source-address-filter 1.1.0.1/30 orlonger
user@R3# set term A then accept

[edit policy-options policy-statement static-route-tobgp]
user@R3# set term static from protocol static
user@R3# set term static from protocol direct
user@R3# set term static then accept
```

5. Configure PIM.

```
[edit protocols pim]
user@R3# set mldp-inband-signalling policy mldppim-ex
user@R3# set interface lo0.0
user@R3# set interface ge-1/2/18.0
user@R3# set interface ge-1/2/22.0
```

6. Configure LDP.

```
[edit protocols ldp]
user@R3# set interface all
user@R3# set p2mp
```

7. Configure an IGP or static routes.

```
[edit protocols ospf]
user@R3# set traffic-engineering
user@R3# set area 0.0.0.0 interface all
user@R3# set area 0.0.0.0 interface fxp0.0 disable
user@R3# set area 0.0.0.0 interface lo0.0 passive
```

8. Configure internal BGP.

```
[edit protocols bgp group ibgp]
user@R3# set local-address 1.1.1.3
user@R3# set peer-as 10
user@R3# set neighbor 1.1.1.1
user@R3# set neighbor 1.1.1.5
```

9. Configure MPLS and, optionally, RSVP.

```
[edit protocols mpls]
user@R3# set interface all

[edit protocols rsvp]
user@R3# set interface all
```

10. Enable MoFRR.

```
[edit routing-options multicast]
user@R3# set stream-protection
```

Results From configuration mode, confirm your configuration by entering the **show chassis**, **show interfaces**, **show protocols**, **show policy-options**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@R3# show chassis
network-services enhanced-ip;

user@R3# show interfaces
ge-1/2/14 {
   unit 0 {
      description R3-to-R2;
      family inet {
         address 1.2.3.2/30;
      }
      family mpls;
   }
}
ge-1/2/18 {
   unit 0 {
```

```
        description R3-to-R4;
        family inet {
          address 1.3.4.1/30;
        }
        family mpls;
      }
    }
    ge-1/2/19 {
      unit 0 {
        description R3-to-R6;
        family inet {
          address 1.3.6.2/30;
        }
        family mpls;
      }
    }
    ge-1/2/21 {
      unit 0 {
        description R3-to-R7;
        family inet {
          address 1.3.7.1/30;
        }
        family mpls;
      }
    }
    ge-1/2/22 {
      unit 0 {
        description R3-to-R8;
        family inet {
          address 1.3.8.1/30;
        }
        family mpls;
      }
    }
    ge-1/2/15 {
      unit 0 {
        description R3-to-R2;
        family inet {
          address 1.2.94.2/30;
        }
        family mpls;
      }
    }
    ge-1/2/20 {
      unit 0 {
        description R3-to-R6;
        family inet {
          address 1.2.96.2/30;
        }
        family mpls;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 192.168.15.1/32;
```

```
        address 1.1.1.3/32 {
          primary;
        }
      }
    }
}
```

```
user@R3# show protocols
rsvp {
  interface all;
}
mpls {
  interface all;
}
bgp {
  group ibgp {
    local-address 1.1.1.3;
    peer-as 10;
    neighbor 1.1.1.1;
    neighbor 1.1.1.5;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface lo0.0 {
      passive;
    }
  }
}
ldp {
  interface all;
  p2mp;
}
pim {
  mldp-inband-signalling {
    policy mldppim-ex;
  }
  interface lo0.0;
  interface ge-1/2/18.0;
  interface ge-1/2/22.0;
}
```

```
user@R3# show policy-options
policy-statement mldppim-ex {
  term B {
    from {
      source-address-filter 192.168.0.0/24 orlonger;
      source-address-filter 192.168.219.11/32 orlonger;
    }
    then accept;
  }
  term A {
```

```
      from {
        source-address-filter 1.1.0.1/30 orlonger;
      }
      then accept;
    }
  }
  policy-statement static-route-tobgp {
    term static {
      from protocol [ static direct ];
      then accept;
    }
  }
```

```
user@R3# show routing-options
autonomous-system 10;
multicast {
  stream-protection;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Checking the LDP Point-to-Multipoint Forwarding Equivalency Classes

Purpose Make sure the MoFRR is enabled, and determine what labels are being used.

Action

```
user@R3> show ldp p2mp fec

LDP P2MP FECs:
 P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src: 192.168.219.11
  MoFRR enabled
  Fec type: Egress (Active)
  Label: 301568
 P2MP root-addr 1.1.1.1, grp: 232.1.1.2, src: 192.168.219.11
  MoFRR enabled
  Fec type: Egress (Active)
  Label: 301600
```

Meaning The output shows that MoFRR is enabled, and it shows that the labels 301568 and 301600 are being used for the two multipoint LDP point-to-multipoint LSPs.

Examining the Label Information

Purpose Make sure that the egress device has two upstream interfaces for the multicast group join.

Action

```
user@R3> show route label 301568 detail

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301568 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x2735208
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 1397
                Address: 0x2735d2c
                Next-hop reference count: 3
                Next hop: 1.3.8.2 via ge-1/2/22.0
                Label operation: Pop
                Load balance label:  None;
                Next hop type: Router, Next hop index: 1395
                Address: 0x2736290
                Next-hop reference count: 3
                Next hop: 1.3.4.2 via ge-1/2/18.0
                Label operation: Pop
                Load balance label:  None;
                State: <Active Int AckRequest MulticastRPF>
                Local AS:    10
                Age: 54:05      Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
               FECs bound to route: P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src:
 192.168.219.11
                Primary Upstream : 1.1.1.3:0--1.1.1.2:0
                   RPF Nexthops  :
                        ge-1/2/15.0, 1.2.94.1, Label: 301568, weight: 0x1
                        ge-1/2/14.0, 1.2.3.1, Label: 301568, weight: 0x1
                Backup Upstream : 1.1.1.3:0--1.1.1.6:0
                   RPF Nexthops :
                        ge-1/2/20.0, 1.2.96.1, Label: 301584, weight: 0xfffe
                        ge-1/2/19.0, 1.3.6.1, Label: 301584, weight: 0xfffe


user@R3> show route label 301600 detail

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301600 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x27356b4
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 1520
                Address: 0x27350f4
                Next-hop reference count: 3
                Next hop: 1.3.8.2 via ge-1/2/22.0
                Label operation: Pop
                Load balance label:  None;
                Next hop type: Router, Next hop index: 1481
                Address: 0x273645c
                Next-hop reference count: 3
                Next hop: 1.3.4.2 via ge-1/2/18.0
                Label operation: Pop
                Load balance label:  None;
                State: <Active Int AckRequest MulticastRPF>
```

```
                Local AS:    10
                Age: 54:25      Metric: 1
                Validation State: unverified
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
               FECs bound to route: P2MP root-addr 1.1.1.1, grp: 232.1.1.2, src:
192.168.219.11
                Primary Upstream : 1.1.1.3:0--1.1.1.6:0
                   RPF Nexthops  :
                        ge-1/2/20.0, 1.2.96.1, Label: 301600, weight: 0x1
                        ge-1/2/19.0, 1.3.6.1, Label: 301600, weight: 0x1
                Backup Upstream : 1.1.1.3:0--1.1.1.2:0
                   RPF Nexthops :
                        ge-1/2/15.0, 1.2.94.1, Label: 301616, weight: 0xfffe
                        ge-1/2/14.0, 1.2.3.1, Label: 301616, weight: 0xfffe
```

Meaning The output shows the primary upstream paths and the backup upstream paths. It also shows the RPF next hops.

Checking the Multicast Routes

Purpose Examine the IP multicast forwarding table to make sure that there is an upstream RPF interface list, with a primary and a backup interface.

Action

```
user@R3> show ldp p2mp path
P2MP path type: Transit/Egress
  Output Session (label): 1.1.1.2:0 (301568) (Primary)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 1.2.94.1, 301568, 1
                   Interface ge-1/2/20.0, 1.2.96.1, 301584, 65534
                   Interface ge-1/2/14.0, 1.2.3.1, 301568, 1
                   Interface ge-1/2/19.0, 1.3.6.1, 301584, 65534
  Attached FECs:  P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src: 192.168.219.11
(Active)
P2MP path type: Transit/Egress
  Output Session (label): 1.1.1.6:0 (301584) (Backup)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 1.2.94.1, 301568, 1
                   Interface ge-1/2/20.0, 1.2.96.1, 301584, 65534
                   Interface ge-1/2/14.0, 1.2.3.1, 301568, 1
                   Interface ge-1/2/19.0, 1.3.6.1, 301584, 65534
  Attached FECs:  P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src: 192.168.219.11
(Active)
P2MP path type: Transit/Egress
  Output Session (label): 1.1.1.6:0 (301600) (Primary)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 1.2.94.1, 301616, 65534
                   Interface ge-1/2/20.0, 1.2.96.1, 301600, 1
                   Interface ge-1/2/14.0, 1.2.3.1, 301616, 65534
                   Interface ge-1/2/19.0, 1.3.6.1, 301600, 1
  Attached FECs:  P2MP root-addr 1.1.1.1, grp: 232.1.1.2, src: 192.168.219.11
(Active)
P2MP path type: Transit/Egress
  Output Session (label): 1.1.1.2:0 (301616) (Backup)
  Egress Nexthops: Interface ge-1/2/18.0
                   Interface ge-1/2/22.0
  RPF Nexthops:    Interface ge-1/2/15.0, 1.2.94.1, 301616, 65534
                   Interface ge-1/2/20.0, 1.2.96.1, 301600, 1
                   Interface ge-1/2/14.0, 1.2.3.1, 301616, 65534
                   Interface ge-1/2/19.0, 1.3.6.1, 301600, 1
  Attached FECs:  P2MP root-addr 1.1.1.1, grp: 232.1.1.2, src: 192.168.219.11
(Active)
```

Meaning The output shows primary and backup sessions, and RPF next hops.

Checking the LDP Point-to-Multipoint Traffic Statistics

Purpose Make sure that both primary and backup statistics are listed.

Action

```
user@R3> show ldp traffic-statistics p2mp

P2MP FEC Statistics:

FEC(root_addr:lsp_id/grp,src)     Nexthop              Packets             Bytes
 Shared
1.1.1.1:232.1.1.1,192.168.219.11, Label: 301568
                                  1.3.8.2                    0                 0
 No
                                  1.3.4.2                    0                 0
 No
1.1.1.1:232.1.1.1,192.168.219.11, Label: 301584, Backup route
                                  1.3.4.2                    0                 0
 No
                                  1.3.8.2                    0                 0
 No
1.1.1.1:232.1.1.2,192.168.219.11, Label: 301600
                                  1.3.8.2                    0                 0
 No
                                  1.3.4.2                    0                 0
 No
1.1.1.1:232.1.1.2,192.168.219.11, Label: 301616, Backup route
                                  1.3.4.2                    0                 0
 No
                                  1.3.8.2                    0                 0
 No
```

Meaning

The output shows both primary and backup routes with the labels.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Configuring Multicast-Only Fast Reroute on page 834
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837

CHAPTER 25

Enabling Multicast Between Layer 2 and Layer 3 Devices Using Snooping

Multicast Snooping on MX Series Routers

Because MX Series routers can support both Layer 3 and Layer 2 functions at the same time, you can configure the Layer 3 multicast protocols Protocol Independent Multicast (PIM) and the Internet Group Membership Protocol (IGMP) as well as Layer 2 VLANs on an MX Series router.

Normal encapsulation rules restrict Layer 2 processing to accessing information in the frame header and Layer 3 processing to accessing information in the packet header. However, in some cases, an interface running a Layer 2 protocol needs information available only at Layer 3. In multicast applications, the VLANs need the group membership information and multicast tree information available to the Layer 3 IGMP and PIM protocols. In these cases, the Layer 3 configurations can use PIM or IGMP snooping to provide the needed information at the VLAN level.

For information about configuring multicast snooping for the operational details of a Layer 3 protocol on behalf of a Layer 2 spanning-tree protocol process, see "Understanding Multicast Snooping and VPLS Root Protection" on page 872.

Snooping configuration statements and examples are not included in the *Junos OS Layer 2 Switching and Bridging Library* . For more information about configuring PIM and IGMP snooping, see the *Multicast Protocols Feature Guide* .

Related Documentation

- Understanding Multicast Snooping and VPLS Root Protection on page 872

- Configuring Multicast Snooping to Ignore Spanning Tree Topology Change Messages on page 883
- Example: Configuring Multicast Snooping for a Bridge Domain on page 881

Example: Configuring Multicast Snooping

- Understanding Multicast Snooping on page 872
- Understanding Multicast Snooping and VPLS Root Protection on page 872
- Configuring Multicast Snooping on page 873
- Example: Configuring Multicast Snooping on page 874
- Enabling Bulk Updates for Multicast Snooping on page 879
- Enabling Multicast Snooping for Multichassis Link Aggregation Group Interfaces on page 880

Understanding Multicast Snooping

Network devices such as routers operate mainly at the packet level, or Layer 3. Other network devices such as bridges or LAN switches operate mainly at the frame level, or Layer 2. Multicasting functions mainly at the packet level, Layer 3, but there is a way to map Layer 3 IP multicast group addresses to Layer 2 MAC multicast group addresses at the frame level.

Routers can handle both Layer 2 and Layer 3 addressing information because the frame and its addresses must be processed to access the encapsulated packet inside. Routers can run Layer 3 multicast protocols such as PIM or IGMP and determine where to forward multicast content or when a host on an interface joins or leaves a group. However, bridges and LAN switches, as Layer 2 devices, are not supposed to have access to the multicast information inside the packets that their frames carry.

How then are bridges and other Layer 2 devices to determine when a device on an interface joins or leaves a multicast tree, or whether a host on an attached LAN wants to receive the content of a particular multicast group?

The answer is for the Layer 2 device to implement multicast snooping. Multicast snooping is a general term and applies to the process of a Layer 2 device “snooping” at the Layer 3 packet content to determine which actions are taken to process or forward a frame. There are more specific forms of snooping, such as IGMP snooping or PIM snooping. In all cases, snooping involves a device configured to function at Layer 2 having access to normally “forbidden” Layer 3 (packet) information. Snooping makes multicasting more efficient in these devices.

See Also
- Layer 2 Frames and IPv4 Multicast Addresses on page 9

Understanding Multicast Snooping and VPLS Root Protection

Snooping occurs when a Layer 2 protocol such as a spanning-tree protocol is aware of the operational details of a Layer 3 protocol such as the Internet Group Management

Protocol (IGMP) or other multicast protocol. Snooping is necessary when Layer 2 devices such as VLAN switches must be aware of Layer 3 information such as the media access control (MAC) addresses of members of a multicast group.

VPLS root protection is a spanning-tree protocol process in which only one interface in a multihomed environment is actively forwarding spanning-tree protocol frames. This protects the root of the spanning tree against bridging loops, but also prevents both devices in the multihomed topology from snooped information, such as IGMP membership reports.

For example, consider a collection of multicast-capable hosts connected to two customer edge (CE) routers (CE1 and CE2) which are connected to each other (a CE1–CE2 link is configured) and multihomed to two provider edge (PE) routers (PE1 and PE2, respectively). The active PE only receives forwarded spanning-tree protocol information on the active PE-CE link, due to root protection operation. As long as the CE1–CE2 link is operational, this is not a problem. However, if the link between CE1 and CE2 fails, and the other PE becomes the active spanning-tree protocol link, no multicast snooping information is available on the new active PE. The new active PE will not forward multicast traffic to the CE and the hosts serviced by this CE router.

The service outage is corrected once the hosts send new group membership IGMP reports to the CE routers. However, the service outage can be avoided if multicast snooping information is available to both PEs in spite of normal spanning-tree protocol root protection operation.

You can configure multicast snooping to ignore messages about spanning tree topology changes on bridge domains on virtual switches and bridge domains default routing switches. You can use the **ignore-stp-topology-change** command to ignore messages about spanning tree topology changes

See Also

- *Understanding VPLS Multihomed Layer 2 Ring and MPLS Infrastructure* in the *Junos OS Layer 2 Switching and Bridging Library*
- Multicast Snooping on MX Series Routers on page 871 in the *Junos OS Layer 2 Switching and Bridging Library*
- Configuring Multicast Snooping to Ignore Spanning Tree Topology Change Messages on page 883 in the *Junos OS Layer 2 Switching and Bridging Library*
- Example: Configuring Multicast Snooping for a Bridge Domain on page 881 in the *Junos OS Layer 2 Switching and Bridging Library*
- *Multicast Protocols Feature Guide*
- ignore-stp-topology-change on page 1080

Configuring Multicast Snooping

To configure the general multicast snooping parameters for MX Series routers, include the **multicast-snooping-options** statement:

```
multicast-snooping-options {
  flood-groups [ ip-addresses ];
```

```
    forwarding-cache {
      threshold suppress value <reuse value>;
    }
    graceful-restart <restart-duration seconds>;
    ignore-stp-topology-change;
    multichassis-lag-replicate-state;
    nexthop-hold-time milliseconds;
    traceoptions {
      file filename <files number> <size size> <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
  }
```

You can include this statement at the following hierarchy levels:

- **[edit routing-instances *routing-instance-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**

By default, multicast snooping is disabled. You can enable multicast snooping in VPLS or virtual switch instance types in the instance hierarchy.

If there are multiple bridge domains configured under a VPLS or virtual switch instance, the multicast snooping options configured at the instance level apply to all the bridge domains.

NOTE: The **ignore-stp-topology-change** statement is supported for the **virtual-switch** routing instance type only and is not supported under the **[edit logical-systems]** hierarchy.

NOTE: The **nexthop-hold-time** statement is supported only at the **[edit routing-instances *routing-instance-name*]** hierarchy, and only for an instance type of **virtual-switch** or **vpls**.

See Also

- Configuring IGMP Snooping on page 104
- Configuring VLAN-Specific IGMP Snooping Parameters on page 105
- Configuring IGMP Snooping Trace Operations on page 113
- Example: Configuring IGMP Snooping on page 106

Example: Configuring Multicast Snooping

This example shows how to configure multicast snooping in a bridge or VPLS routing-instance scenario.

- Requirements on page 875
- Overview and Topology on page 875

Requirements

This example uses the following hardware components:

- One MX Series router
- One Layer 3 device functioning as a multicast router

Before you begin:

- Configure the interfaces.
- Configure an interior gateway protocol. See the *Junos OS Routing Protocols Library*.
- Configure a multicast protocol. This feature works with the following multicast protocols:
  - DVMRP
  - PIM-DM
  - PIM-SM
  - PIM-SSM

Overview and Topology

IGMP snooping prevents Layer 2 devices from indiscriminately flooding multicast traffic out all interfaces. The settings that you configure for multicast snooping help manage the behavior of IGMP snooping.

You can configure multicast snooping options on the default master instance and on individual bridge or VPLS instances. The default master instance configuration is global and applies to all individual bridge or VPLS instances in the logical router. The configuration for the individual instances overrides the global configuration.

This example includes the following statements:

- **flood-groups**—Enables you to list multicast group addresses for which traffic must be flooded. This setting if useful for making sure that IGMP snooping does not prevent necessary multicast flooding. The block of multicast addresses from 224.0.0.1 through 224.0.0.255 is reserved for local wire use. Groups in this range are assigned for various uses, including routing protocols and local discovery mechanisms. For example, OSPF uses 224.0.0.5 for all OSPF routers.
- **forwarding-cache**—Specifies how forwarding entries are aged out and how the number of entries is controlled.

  You can configure threshold values on the forwarding cache to suppress (suspend) snooping when the cache entries reach a certain maximum and reuse the cache when the number falls to another threshold value. By default, no threshold values are enabled on the router.

The suppress threshold suppresses new multicast forwarding cache entries. An optional reuse threshold specifies the point at which the router begins to create new multicast forwarding cache entries. The range for both thresholds is from 1 through 200,000. If configured, the reuse value must be less than the suppression value. The suppression value is mandatory. If you do not specify the optional reuse value, then the number of multicast forwarding cache entries is limited to the suppression value. A new entry is created as soon as the number of multicast forwarding cache entries falls below the suppression value.

- **graceful-restart**—Configures the time after which routes learned before a restart are replaced with routes relearned. If graceful restart for multicast snooping is disabled, snooping information is lost after a Routing Engine restart.

  By default, the graceful restart duration is 180 seconds (3 minutes). You can set this value between 0 and 300 seconds. If you set the duration to 0, graceful restart is effectively disabled. Set this value slightly larger than the IGMP query response interval.

- **ignore-stp-topology-change**—Configures the MX Series router to ignore messages about the spanning-tree topology state change.

  By default the IGMP snooping process on an MX Series router detects interface state changes made by any of the spanning tree protocols (STPs).

  In a VPLS multihoming environment where two PE routers are connected to two interconnected CE routers and STP root protection is enabled on the PE routers, one of the PE router interfaces is in forwarding state and the other is in blocking state.

  If the link interconnecting the two CE routers fails, the PE router interface in blocking state transitions to the forwarding state.

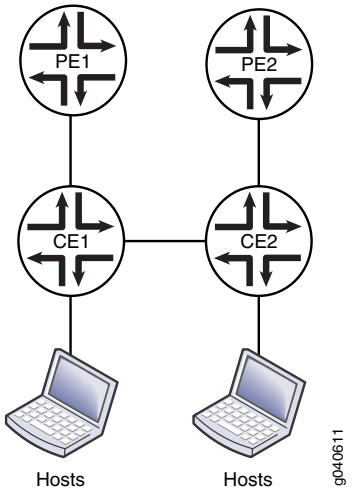
  The PE router interface does not wait to receive membership reports in response to the next general or group-specific query. Instead, the IGMP snooping process sends a general query message toward the CE router. The hosts connected to the CE router reply with reports for all groups they are interested in.

  When the link interconnecting the two CE routers is restored, the original spanning-tree state on both PE routers is restored. The forwarding PE receives a spanning-tree topology change message and sends a general query message toward the CE router to immediately reconstruct the group membership state.

NOTE: **The ignore-stp-topology-change statement is supported for the virtual-switch routing instance type only.**

Figure 126 on page 877 shows a VPLS multihoming topology in which a customer network has two CE devices with a link between them. Each CE is connected to one PE.

Figure 126: VPLS Multihoming Topology



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set bridge-domains domain1 multicast-snooping-options forwarding-cache threshold
  suppress 100
set bridge-domains domain1 multicast-snooping-options forwarding-cache threshold
  reuse 50
set bridge-domains domain1 multicast-snooping-options graceful-restart restart-duration
  120
set routing-instances ce1 instance-type virtual-switch
set routing-instances ce1 bridge-domains domain1 domain-type bridge
set routing-instances ce1 bridge-domains domain1 vlan-id 100
set routing-instances ce1 bridge-domains domain1 interface ge-0/3/9.0
set routing-instances ce1 bridge-domains domain1 interface ge-0/0/6.0
set routing-instances ce1 bridge-domains domain1 multicast-snooping-options
  flood-groups 224.0.0.5
set routing-instances ce1 bridge-domains domain1 multicast-snooping-options
  ignore-stp-topology-change
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the *CLI User Guide*.

To configure IGMP snooping:

1. Configure multicast snooping settings in the master routing instance.

```
[edit bridge-domains domain1]
user@host# set multicast-snooping-options forwarding-cache threshold suppress
  100 reuse 50
user@host# set multicast-snooping-options graceful-restart 120
```

2. Configure the routing instance.

```
[edit routing-instances ce1]
user@host# set instance-type virtual-switch
```

3. Configure the bridge domain in the routing instance.

```
[edit routing-instances ce1 bridge-domains domain1]
user@host# set domain-type bridge
user@host# set interface ge-0/0/6.0
user@host# set interface ge-0/3/9.0
user@host# set vlan-id 100
```

4. Configure flood groups.

```
[edit routing-instances ce1 bridge-domains domain1]
user@host# set multicast-snooping-options flood-groups 224.0.0.5
```

5. Configure the router to ignore messages about spanning-tree topology state changes.

```
[edit routing-instances ce1 bridge-domains domain1]
user@host# set multicast-snooping-options ignore-stp-topology-change
```

6. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results Confirm your configuration by entering the **show bridge-domains** and **show routing-instances** commands.

```
user@host# show bridge-domains
domain1 {
  multicast-snooping-options {
    forwarding-cache {
      threshold {
        suppress 100;
        reuse 50;
      }
    }
  }
}
```

```
user@host# show routing-instances
ce1 {
  instance-type virtual-switch;
  bridge-domains {
    domain1 {
      domain-type bridge;
      vlan-id 100;
      interface ge-0/3/9.0; ## 'ge-0/3/9.0' is not defined
      interface ge-0/0/6.0; ## 'ge-0/0/6.0' is not defined
      multicast-snooping-options {
```

```
            flood-groups 224.0.0.5;
            ignore-stp-topology-change;
        }
    }
  }
}
```

Verification

To verify the configuration, run the following commands:

- show igmp snooping interface
- show igmp snooping membership
- show igmp snooping statistics
- show multicast snooping route
- show route table

See Also

- Example: Configuring IGMP Snooping on page 106
- *Understanding Root Protection for Spanning-Tree Instance Interfaces in a Layer 2 Switched Network*
- Understanding Multicast Snooping and VPLS Root Protection on page 872
- query-response-interval on page 1236

Enabling Bulk Updates for Multicast Snooping

Whenever an individual interface joins or leaves a multicast group, a new next hop entry is installed in the routing table and the forwarding table. You can use the **nexthop-hold-time** statement to specify a time, from 1 through 1000 milliseconds (ms), during which outgoing interface changes are accumulated and then updated in bulk to the routing table and forwarding table. Bulk updating reduces the processing time and memory overhead required to process join and leave messages. This is useful for applications such as Internet Potocol television (IPTV), in which users changing channels can create thousands of interfaces joining or leaving a group in a short period. In IPTV scenarios, typically there is a relatively small and controlled number of streams and a high number of outgoing interfaces. Using bulk updates can reduce the join delay.

In this example, you configure a hold-time of 20 milliseconds for **instance-type virtual-switch**, using the **nexthop-hold-time** statement:

1. Enable the **nexthop-hold-time** statement by configuring it under **multicast-snooping-options**, using 20 milliseconds for the time value.

```
[edit routing-instances vs]
multicast-snooping-options {
  nexthop-hold-time 20;
}
```

2. Use the **show multicast snooping route** command to verify that the bulk updates feature is turned on.

```
user@host> show multicast snooping route instance vs
Nexthop Bulking: ON
Family: INET
Group: 224.0.0.0
```

You can include the **nexthop-hold-time** statement only for routing-instance types of **virtual-switch** or **vpls** at the following hierarchy level.

- **[edit routing-instances** *routing-instance-name* **multicast-snooping-options]**

If the **nexthop-hold-time** statement is deleted from the router configuration, bulk updates are disabled.

See Also
- multicast-snooping-options on page 1174
- nexthop-hold-time on page 1183

Enabling Multicast Snooping for Multichassis Link Aggregation Group Interfaces

Include the **multichassis-lag-replicate-state** statement at the **[edit multicast-snooping-options]** hierarchy level to enable IGMP snooping and state replication for multichassis link aggregation group (MC-LAG) interfaces.

```
[edit]
multicast-snooping-options {
  multichassis-lag-replicate-state;
}
```

Replicating join and leave messages between links of a dual-link MC-LAG interface enables faster recovery of membership information for MC-LAG interfaces that experience service interruption.

Without state replication, if a dual-link MC-LAG interface experiences a service interruption (for example, if an active link switches to standby), the membership information for the interface is recovered by generating an IGMP query to the network. This method can take from 1 through 10 seconds to complete, which might be too long for some applications.

When state replication is provided for MC-LAG interfaces, IGMP join or leave messages received on an MC-LAG device are replicated from the active MC-LAG link to the standby link through an Interchassis Communication Protocol (ICCP) connection. The standby link processes the messages as if they were received from the corresponding active MC-LAG link, except it does not add itself as a next hop and it does not flood the message to the network. After a failover, the multicast membership status of the link can be recovered within a few seconds or less by retrieving the replicated messages.

This example enables state replication for MC-LAG interfaces:

1. Enable state replication for MC-LAG interfaces on the routing device.

```
user@host# set multicast-snooping-options multicast-lag-replicate-state
```

After you commit the configuration, multicast snooping automatically identifies the active link during initialization or after failover, and replicates data between the active and standby links without administrator intervention.

2. Use the **show igmp snooping interface** command to display the state for MC-LAG interfaces.

```
user@host> show igmp snooping interface

Learning-Domain: default
Interface: ae0.1
   State: Up Groups: 1
   mc-lag state: standby
   Immediate leave: Off
Router interface: no
Interface: ge-0/1/3.100
   State: Up Groups: 1
   Immediate leave: Off
Router interface: no
Interface: ae1.2
   State: Up Groups: 1
   mc-lag state: standby
   Immediate leave: Off
Router interface: no
```

NOTE: You can use the **show igmp snooping membership** command to display group membership information for the links of MC-LAG interfaces.

If you delete the **multicast-lag-replicate-state** statement or the configuration of IGMP snooping, replication between MC-LAG links stops within the hierarchy level from which the configuration was deleted. Then, multicast membership is recovered as needed by generating standard IGMP queries over the network.

See Also

- multichassis-lag-replicate-state on page 1175
- Configuring Multicast Snooping on page 873

Example: Configuring Multicast Snooping for a Bridge Domain

This example configures the multicast snooping option for a bridge domain named **Ignore-STP** in a virtual switch routing instance named **vs_routing_instance_multihomed_CEs**:

```
[edit]
routing-instances {
   vs_routing_instance_multihomed_CEs {
      instance-type virtual-switch;
      bridge-domains {
         bd_ignore_STP {
            multicast-snooping-options {
               ignore-stp-topology-change;
```

```
        }
      }
    }
  }
}
```

NOTE: This is not a complete router configuration.

Related Documentation

- Multicast Snooping on MX Series Routers on page 871
- Understanding Multicast Snooping and VPLS Root Protection on page 872
- Configuring Multicast Snooping to Ignore Spanning Tree Topology Change Messages on page 883

Configuring Multicast Snooping to Ignore Spanning Tree Topology Change Messages

You can configure the multicast snooping process for a virtual switch to ignore VPLS root protection topology change messages.

Before you begin, complete the following tasks:

1. Configure the spanning-tree protocol. For configuration details, see one of the following topics:
   - *Configuring Rapid Spanning Tree Protocol*
   - *Configuring Multiple Spanning Tree Protocol*
   - *Configuring VLAN SpanningTree Protocol*
2. Configure VPLS root protection. For configuration details, see one of the following topics:
   - *Configuring VPLS Root Protection Topology Change Actions to Control Global Spanning-Tree Behavior*
   - *Configuring VPLS Root Protection Topology Change Actions to Control VLAN Spanning-Tree Behavior*

To configure multicast snooping to ignore spanning tree topology change messages:

1. Configure a **virtual-switch** routing instance to isolate a LAN segment with its VSTP instance.
   a. Enable configuration of a virtual switch routing instance:

      ```
      [edit]
      user@host# edit routing-instances routing-instance-name
      user@host# set instance-type virtual-switch
      ```

      You can configure multicast snooping to ignore messages about spanning tree topology changes for the **virtual-switch** routing-instance type only.

   b. Enable configuration of a bridge domain:

      ```
      [edit routing-instances routing-instance-name]
      user@host# edit bridge-domains bridge-domain-name
      user@host# set domain-type bridge
      ```

   c. Configure the logical interfaces for the bridge domain in the virtual switch:

      ```
      [edit routing-instances routing-instance-name bridge-domains bridge-domain-name]
      user@host# set interface interface-name
      ```

   d. Configure the VLAN identifiers for the bridge domain in the virtual switch. For detailed information, see *Configuring a Virtual Switch Routing Instance on MX Series Routers*.

2. Configure the multicast snooping process to ignore any spanning tree topology change messages sent to the virtual switch routing instance:

   [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]
   user@host# **set** multicast-snooping-options ignore-stp-topology-change

3. Verify the configuration of multicast snooping for the virtual-switch routing instance to ignore spanning tree topology change messages:

   [edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]
   user@host# **top**
   user@host# **show routing-instances**

```
routing-instance-name {
    instance-type virtual-switch;
    bridge-domains {
        bridge-domain-name {
            domain-type bridge {
            interface interface-name;
            ...VLAN-identifiers-configuration...
            multicast-snooping-options {
                ignore-stp-topology-change;
            }
        }
    }
}
```

Related Documentation

- Multicast Snooping on MX Series Routers on page 871
- Understanding Multicast Snooping and VPLS Root Protection on page 872
- Example: Configuring Multicast Snooping for a Bridge Domain on page 881

Configuring Graceful Restart for Multicast Snooping

When graceful restart is enabled for multicast snooping, no data traffic is lost during a process restart or a graceful Routing Engine switchover (GRES). Graceful restart can be configured for multicast snooping either at the global level or at the level of individual routing instances.

At the global level, graceful restart is enabled by default for multicast snooping. To change this default setting, you can configure the **disable** statement at the **[edit multicast-snooping-options graceful-restart]** hierarchy level:

```
multicast-snooping-options {
    graceful-restart disable;
}
```

To configure graceful restart for multicast snooping on a global level:

1. Configure the duration for graceful restart.

   [edit multicast-snooping-options graceful-restart]
   user@host# **set** restart-duration **200**

The range for **restart-duration** is from 0 through 300 seconds. The default value is 180 seconds. After this period, the Routing Engine resumes normal multicast operation.

You can also set the **graceful-restart** statement for an individual routing instance level at the **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* multicast-snooping-options]** hierarchy level.

2. Verify your configuration by using the **show multicast-snooping-options** command.

```
[edit]
user@host# show multicast-snooping-options
```

```
graceful-restart {
    restart-duration 200;
}
```

3. Commit the configuration.

```
[edit]
user@host# commit
```

To configure graceful restart for multicast snooping for an individual routing instance level:

1. Configure the duration for graceful restart.

```
[edit routing-instances ri1 multicast-snooping-options graceful-restart]
user@host# set restart-duration 200
```

The range for **restart-duration** is from 0 through 300 seconds. The default value is 180 seconds. After this period, the Routing Engine resumes normal multicast operation.

NOTE: You can also set the graceful-restart statement for an individual routing instance level at the [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* multicast-snooping-options] hierarchy level.

2. Verify your configuration by using the **show routing-instances *routing-instance-name* multicast-snooping-options** command.

```
[edit]
user@host# show routing-instances ri1 multicast-snooping-options
```

```
graceful-restart {
    restart-duration 200;
}
```

3. Commit the configuration.

```
[edit]
user@host# commit
```

Related Documentation
- Example: Configuring Multicast Snooping on page 874
- graceful-restart (Multicast Snooping) on page 1037

PIM Snooping for VPLS

- Understanding PIM Snooping for VPLS on page 886
- Example: Configuring PIM Snooping for VPLS on page 887

Understanding PIM Snooping for VPLS

There are two ways to direct PIM control packets:

- By the use of PIM snooping
- By the use of PIM proxying

PIM snooping configures a device to examine and operate only on PIM hello and join/prune packets. A PIM snooping device snoops PIM hello and join/prune packets on each interface to find interested multicast receivers and populates the multicast forwarding tree with this information. PIM snooping differs from PIM proxying in that both PIM hello and join/prune packets are transparently flooded in the VPLS as opposed to the flooding of only hello packets in the case of PIM proxying. PIM snooping is configured on PE routers connected through pseudowires. PIM snooping ensures that no new PIM packets are generated in the VPLS, with the exception of PIM messages sent through LDP on pseudowires.

NOTE: In the VPLS documentation, the word *router* in terms such as *PE router* is used to refer to any device that provides routing functions.

A device that supports PIM snooping snoops hello packets received on attachment circuits. It does not introduce latency in the VPLS core when it forwards PIM join/prune packets.

To configure PIM snooping on a PE router, use the **pim-snooping** statement at the **[edit routing-instances *instance-name* protocols]** hierarchy level:

```
routing-instances {
   customer {
      instance-type vpls;
      ...
      protocols {
         pim-snooping{
            traceoptions {
               file pim.log size 10m;
               flag all;
               flag timer disable;
            }
         }
      }
   }
```

```
}
```

"Example: Configuring PIM Snooping for VPLS" on page 887 explains the PIM snooping method. The use of the PIM proxying method is not discussed here and is outside the scope of this document. For more information about PIM proxying, see PIM Snooping over VPLS.

See Also

- Example: Configuring PIM Snooping for VPLS on page 887

Example: Configuring PIM Snooping for VPLS

This example shows how to configure PIM snooping in a virtual private LAN service (VPLS) to restrict multicast traffic to interested devices.

Requirements

This example uses the following hardware and software components:

- M Series Multiservice Edge Routers (M7i and M10i with Enhanced CFEB, M120, and M320 with E3 FPCs) or MX Series 3D Universal Edge Routers (MX80, MX240, MX480, and MX960)
- Junos OS Release 13.2 or later

Overview

The following example shows how to configure PIM snooping to restrict multicast traffic to interested devices in a VPLS.

NOTE: This example demonstrates PIM snooping by the use of a PIM snooping device to restrict multicast traffic. The use of the PIM proxying method to achieve PIM snooping is out of the scope of this document and is yet to be implemented in Junos OS.

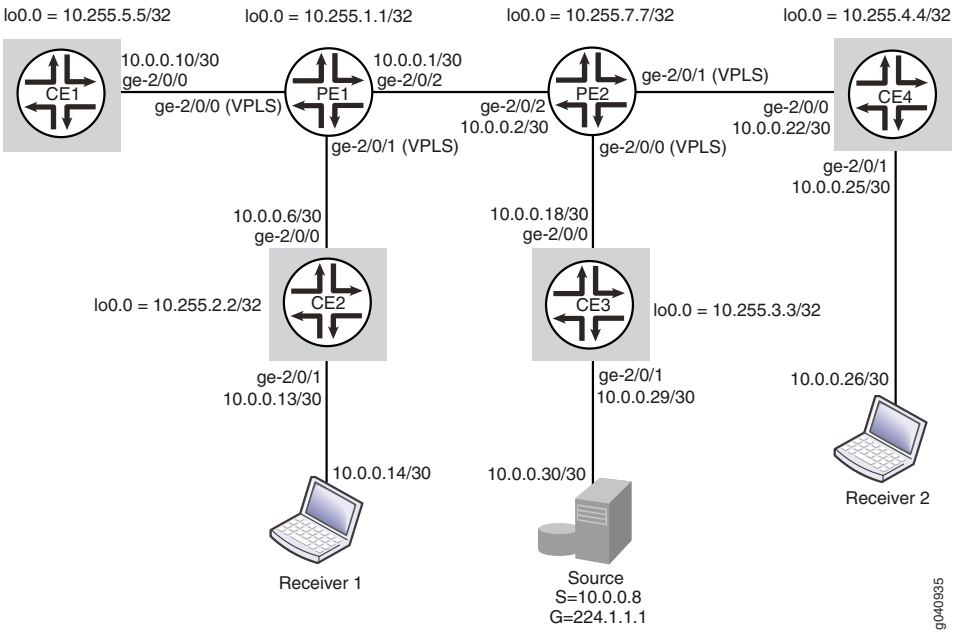
Topology

In this example, two PE routers are connected to each other through a pseudowire connection. Router PE1 is connected to Routers CE1 and CE2. A multicast receiver is attached to Router CE2. Router PE2 is connected to Routers CE3 and CE4. A multicast source is connected to Router CE3, and a second multicast receiver is attached to Router CE4.

PIM snooping is configured on Routers PE1 and PE2. Hence, data sent from the multicast source is received only by members of the multicast group.

Figure 127 on page 888 shows the topology used in this example.

Figure 127: PIM Snooping for VPLS



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router PE1

```
set multicast-snooping-options traceoptions file snoop.log size 10m
set interfaces ge-2/0/0 encapsulation ethernet-vpls
set interfaces ge-2/0/0 unit 0 description toCE1
set interfaces ge-2/0/1 encapsulation ethernet-vpls
set interfaces ge-2/0/1 unit 0 description toCE2
set interfaces ge-2/0/2 unit 0 description toPE2
set interfaces ge-2/0/2 unit 0 family inet address 10.0.0.1/30
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.1.1/32
set routing-options router-id 10.255.1.1
set protocols mpls interface ge-2/0/1.0
set protocols bgp group toPE2 type internal
set protocols bgp group toPE2 local-address 10.255.1.1
set protocols bgp group toPE2 family l2vpn signaling
set protocols bgp group toPE2 neighbor 10.255.7.7
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ldp interface ge-2/0/2.0
set protocols ldp interface lo0.0
set routing-instances titanium instance-type vpls
set routing-instances titanium vlan-id none
```

```
set routing-instances titanium interface ge-2/0/0.0
set routing-instances titanium interface ge-2/0/1.0
set routing-instances titanium route-distinguisher 101:101
set routing-instances titanium vrf-target target:201:201
set routing-instances titanium protocols vpls vpls-id 15
set routing-instances titanium protocols vpls site pe1 site-identifier 1
set routing-instances titanium protocols pim-snooping
```

Router CE1

```
set interfaces ge-2/0/0 unit 0 description toPE1
set interfaces ge-2/0/0 unit 0 family inet address 10.0.0.10/30
set interfaces lo0 unit 0 family inet address 10.255.2.2./32
set routing-options router-id 10.255.2.2
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 10.255.3.3
set protocols pim interface all
```

Router CE2

```
set interfaces ge-2/0/0 unit 0 description toPE1
set interfaces ge-2/0/0 unit 0 family inet address 10.0.0.6/30
set interfaces ge-2/0/1 unit 0 description toReceiver1
set interfaces ge-2/0/1 unit 0 family inet address 10.0.0.13/30
set interfaces lo0 unit 0 family inet address 10.255.2.2
set routing-options router-id 10.255.2.2
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 10.255.3.3
set protocols pim interface all
```

Router PE2

```
set multicast-snooping-options traceoptions file snoop.log size 10m
set interfaces ge-2/0/0 encapsulation ethernet-vpls
set interfaces ge-2/0/0 unit 0 description toCE3
set interfaces ge-2/0/1 encapsulation ethernet-vpls
set interfaces ge-2/0/1 unit 0 description toCE4
set interfaces ge-2/0/2 unit 0 description toPE1
set interfaces ge-2/0/2 unit 0 family inet address 10.0.0.2/30
set interfaces ge-2/0/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 10.255.7.7/32
set routing-options router-id 10.255.7.7
set protocols mpls interface ge-2/0/2.0
set protocols bgp group toPE1 type internal
set protocols bgp group toPE1 local-address 10.255.7.7
set protocols bgp group toPE1 family l2vpn signaling
set protocols bgp group toPE1 neighbor 10.255.1.1
set protocols ospf area 0.0.0.0 interface ge-2/0/2.0
set protocols ospf area 0.0.0.0 interface lo0.0
set protocols ldp interface ge-2/0/2.0
set protocols ldp interface lo0.0
set routing-instances titanium instance-type vpls
set routing-instances titanium vlan-id none
set routing-instances titanium interface ge-2/0/0.0
set routing-instances titanium interface ge-2/0/1.0
set routing-instances titanium route-distinguisher 101:101
set routing-instances titanium vrf-target target:201:201
set routing-instances titanium protocols vpls vpls-id 15
```

```
set routing-instances titanium protocols vpls site pe2 site-identifier 2
set routing-instances titanium protocols pim-snooping
```

Router CE3 (RP)

```
set interfaces ge-2/0/0 unit 0 description toPE2
set interfaces ge-2/0/0 unit 0 family inet address 10.0.0.18/30
set interfaces ge-2/0/1 unit 0 description toSource
set interfaces ge-2/0/1 unit 0 family inet address 10.0.0.29/30
set interfaces lo0 unit 0 family inet address 10.255.3.3/32
set routing-options router-id 10.255.3.3
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp local address 10.255.3.3
set protocols pim interface all
```

Router CE4

```
set interfaces ge-2/0/0 unit 0 description toPE2
set interfaces ge-2/0/0 unit 0 family inet address 10.0.0.22/30
set interfaces ge-2/0/1 unit 0 description toReceiver2
set interfaces ge-2/0/1 unit 0 family inet address 10.0.0.25/30
set interfaces lo0 unit 0 family inet address 10.255.4.4/32
set routing-options router-id 10.255.4.4
set protocols ospf area 0.0.0.0 interface all
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols pim rp static address 10.255.3.3
set protocols pim interface all
```

Configuring PIM Snooping for VPLS

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

NOTE: This section includes a step-by-step configuration procedure for one or more routers in the topology. For comprehensive configurations for all routers, see "CLI Quick Configuration" on page 888.

To configure PIM snooping for VPLS:

1. Configure the router interfaces forming the links between the routers.

   Router PE2

   ```
   [edit interfaces]
   user@PE2# set ge-2/0/0 encapsulation ethernet-vpls
   user@PE2# set ge-2/0/0 unit 0 description toCE3
   user@PE2# set ge-2/0/1 encapsulation ethernet-vpls
   user@PE2# set ge-2/0/1 unit 0 description toCE4
   user@PE2# set ge-2/0/2 unit 0 description toPE1
   user@PE2# set ge-2/0/2 unit 0 family mpls
   user@PE2# set ge-2/0/2 unit 0 family inet address 10.0.0.2/30
   user@PE2# set lo0 unit 0 family inet address 10.255.7.7/32
   ```

NOTE: ge-2/0/0.0 and ge-2/0/1.0 are configured as VPLS interfaces and connect to Routers CE3 and CE4. See *Virtual Private LAN Service Feature Guide* for more details.

```
Router CE3
[edit interfaces]
user@CE3# set ge-2/0/0 unit 0 description toPE2
user@CE3# set ge-2/0/0 unit 0 family inet address 10.0.0.18/30
user@CE3# set ge-2/0/1 unit 0 description toSource
user@CE3# set ge-2/0/1 unit 0 family inet address 10.0.0.29/30
user@CE3# set lo0 unit 0 family inet address 10.255.3.3/32
```

NOTE: The ge-2/0/1.0 interface on Router CE3 connects to the multicast source.

```
Router CE4
[edit interfaces]
user@CE4# set ge-2/0/0 unit 0 description toPE2
user@CE4# set ge-2/0/0 unit 0 family inet address 10.0.0.22/30
user@CE4# set ge-2/0/1 unit 0 description toReceiver2
user@CE4# set ge-2/0/1 unit 0 family inet address 10.0.0.25/30
user@CE4# set lo0 unit 0 family inet address 10.255.4.4/32
```

NOTE: The ge-2/0/1.0 interface on Router CE4 connects to a multicast receiver.

Similarly, configure Routers PE1, CE1, and CE2.

2. Configure the router IDs of all routers.

```
Router PE2
[edit routing-options]
user@PE2# set router-id 10.255.7.7
```

Similarly, configure other routers.

3. Configure an IGP on interfaces of all routers.

```
Router PE2
[edit protocols ospf area 0.0.0.0]
user@PE2# set interface ge-2/0/2.0
user@PE2# set interface lo0.0
```

Similarly, configure other routers.

4. Configure the LDP, MPLS, and BGP protocols on the PE routers.

```
Router PE2
[edit protocols]
user@PE2# set ldp interface lo0.0
user@PE2# set mpls interface ge-2/0/2.0
user@PE2# set bgp group toPE1 type internal
user@PE2# set bgp group toPE1 local-address 10.255.7.7
user@PE2# set bgp group toPE1 family l2vpn signaling
user@PE2# set bgp group toPE1 neighbor 10.255.1.1
user@PE2# set ldp interface ge-2/0/2.0
```

The BGP group is required for interfacing with the other PE router. Similarly, configure Router PE1.

5. Configure PIM on all CE routers.

   Ensure that Router CE3 is configured as the rendezvous point (RP) and that the RP address is configured on other CE routers.

   ```
   Router CE3
   [edit protocols pim]
   user@CE3# set rp local address 10.255.3.3
   user@CE3# set interface all
   ```

   ```
   Router CE4
   [edit protocols pim]
   user@CE4# set rp static address 10.255.3.3
   user@CE4# set interface all
   ```

   Similarly, configure Routers CE1 and CE2.

6. Configure multicast snooping options on the PE routers.

   ```
   Router PE2
   [edit multicast-snooping-options traceoptions]
   user@PE2# set file snoop.log size 10m
   ```

   Similarly, configure Router PE1.

7. Create a routing instance (**titanium**), and configure the VPLS on the PE routers.

   ```
   Router PE2
   [edit routing-instances titanium]
   user@PE2# set instance-type vpls
   user@PE2# set vlan-id none
   user@PE2# set interface ge-2/0/0.0
   user@PE2# set interface ge-2/0/1.0
   user@PE2# set route-distinguisher 101:101
   user@PE2# set vrf-target target:201:201
   user@PE2# set protocols vpls vpls-id 15
   user@PE2# set protocols vpls site pe2 site-identifier 2
   ```

   Similarly, configure Router PE1.

8. Configure PIM snooping on the PE routers.

   ```
   Router PE2
   ```

```
[edit routing-instances titanium]
user@PE2# set protocols pim-snooping
```

Similarly, configure Router PE1.

Results

From configuration mode, confirm your configuration by entering the **show interfaces**, **show routing-options**, **show protocols**, **show multicast-snooping-options**, and **show routing-instances** commands.

If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE2# show interfaces
ge-2/0/2 {
    unit 0 {
        description toPE1
        family inet {
            address 10.0.0.2/30;
        }
        family mpls;
    }
}
ge-2/0/0 {
    encapsulation ethernet-vpls;
    unit 0 {
         description toCE3;
           }
}
ge-2/0/1 {
    encapsulation ethernet-vpls;
    unit 0 {
         description toCE4;
           }
}
lo0 {
    unit 0 {
        family inet {
            address 10.255.7.7/32;
        }
    }
}


user@PE2# show routing-options
router-id 10.255.7.7;


user@PE2# show protocols
mpls {
    interface ge-2/0/2.0;
}
ospf {
    area 0.0.0.0 {
        interface ge-2/0/2.0;
        interface lo0.0;
    }
}
```

```
ldp {
    interface ge-2/0/2.0;
    interface lo0.0;
}
bgp {
    group toPE1 {
        type internal;
        local-address 10.255.7.7;
        family l2vpn {
            signaling;
        }
        neighbor 10.255.1.1;
    }
```

```
user@PE2# show multicast-snooping-options
traceoptions {
    file snoop.log size 10m;
}
```

```
user@PE2# show routing-instances
titanium {
    instance-type vpls;
    vlan-id none;
    interface ge-2/0/0.0;
    interface ge-2/0/1.0;
    route-distinguisher 101:101;
    vrf-target target:201:201;
    protocols {
        vpls {
            site pe2 {
                site-identifier 2;
            }
            vpls-id 15;
        }
        pim-snooping;
  }
}
```

Similarly, confirm the configuration on all other routers. If you are done configuring the routers, enter **commit** from configuration mode.

NOTE: Use the **show protocols** command on the CE routers to verify the configuration for the PIM RP .

Verification

Confirm that the configuration is working properly.

- Verifying PIM Snooping for VPLS on page 894

Verifying PIM Snooping for VPLS

Purpose Verify that PIM Snooping is operational in the network.

Action To verify that PIM snooping is working as desired, use the following commands:

- show pim snooping interfaces
- show pim snooping neighbors **detail**
- show pim snooping statistics
- show pim snooping join
- show pim snooping join **extensive**
- show multicast snooping route **extensive instance** *<instance-name>* **group** *<group-name>*

1. From operational mode on Router PE2, run the **show pim snooping interfaces** command.

```
user@PE2> show pim snooping interfaces
Instance: titanium

Learning-Domain: default

Name                 State IP NbrCnt
ge-2/0/0.0           Up    4       1
ge-2/0/1.0           Up    4       1

DR address: 10.0.0.22
DR flooding is ON
```

   The output verifies that PIM snooping is configured on the two interfaces connecting Router PE2 to Routers CE3 and CE4.

   Similarly, check the PIM snooping interfaces on Router PE1.

2. From operational mode on Router PE2, run the **show pim snooping neighbors detail** command.

```
user@PE2> show pim snooping neighbors detail
Instance: titanium
Learning-Domain: default

Interface: ge-2/0/0.0

    Address: 10.0.0.18
        Uptime: 00:17:06
        Hello Option Holdtime: 105 seconds 99 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 552495559
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                      Tracking is supported

Interface: ge-2/0/1.0

    Address: 10.0.0.22
        Uptime: 00:15:16
        Hello Option Holdtime: 105 seconds 103 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1131703485
```

```
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                      Tracking is supported
```

The output verifies that Router PE2 can detect the IP addresses of its PIM snooping neighbors (10.0.0.18 on CE3 and 10.0.0.22 on CE4).

Similarly, check the PIM snooping neighbors on Router PE1.

3. From operational mode on Router PE2, run the **show pim snooping statistics** command.

```
user@PE2> show pim snooping statistics
Instance: titanium

Learning-Domain: default

Tx J/P messages                             0
RX J/P messages                           246
Rx J/P messages -- seen                     0
Rx J/P messages -- received               246
Rx Hello messages                        1036
Rx Version Unknown                          0
Rx Neighbor Unknown                         0
Rx Upstream Neighbor Unknown                0
Rx J/P Busy Drop                            0
Rx J/P Group Aggregate                      0
Rx Malformed Packet                         0

Rx No PIM Interface                         0
Rx Bad Length                               0
Rx Unknown Hello Option                     0
Rx Unknown Packet Type                      0
Rx Bad TTL                                  0
Rx Bad Destination Address                  0
Rx Bad Checksum                             0
Rx Unknown Version                          0
```

The output shows the number of hello and join/prune messages received by Router PE2. This verifies that PIM sparse mode is operational in the network.

4. Send multicast traffic from the source terminal attached to Router CE3, for the multicast group 203.0.113.1.

5. From operational mode on Router PE2, run the **show pim snooping join**, **show pim snooping join extensive**, and **show multicast snooping route extensive instance *<instance-name>* group *<group-name>*** commands to verify PIM snooping.

```
user@PE2> show pim snooping join
Instance: titanium
Learning-Domain: default

Group: 203.0.113.1
    Source: *
    Flags: sparse,rptree,wildcard
    Upstream neighbor: 10.0.0.18, Port: ge-2/0/0.0
```

```
Group: 203.0.113.1
    Source: 10.0.0.30
    Flags: sparse
    Upstream neighbor: 10.0.0.18, Port: ge-2/0/0.0

user@PE2> show pim snooping join extensive
Instance: titanium
Learning-Domain: default

Group: 203.0.113.1
    Source: *
    Flags: sparse,rptree,wildcard
    Upstream neighbor: 10.0.0.18, Port: ge-2/0/0.0
        Downstream port: ge-2/0/1.0
            Downstream neighbors:
                10.0.0.22 State: Join Flags: SRW Timeout: 180

Group: 203.0.113.1
    Source: 10.0.0.30
    Flags: sparse
    Upstream neighbor: 10.0.0.18, Port: ge-2/0/0.0
        Downstream port: ge-2/0/1.0
            Downstream neighbors:
                10.0.0.22 State: Join Flags: S Timeout: 180
```

The outputs show that multicast traffic sent for the group 203.0.113.1 is sent to Receiver 2 through Router CE4 and also display the upstream and downstream neighbor details.

```
user@PE2> show multicast snooping route extensive instance titanium group 203.0.113.1
Nexthop Bulking: OFF

             Family: INET

Group: 203.0.113.1/24
    Bridge-domain: titanium
    Mesh-group: __all_ces__
    Downstream interface list:
        ge-2/0/1.0 -(1072)
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 1048577
    Route state: Active
    Forwarding state: Forwarding

Group: 203.0.113.1/24
    Source: 10.0.0.8
    Bridge-domain: titanium
    Mesh-group: __all_ces__
    Downstream interface list:
        ge-2/0/1.0 -(1072)
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 1048577
    Route state: Active
    Forwarding state: Forwarding
```

Meaning PIM snooping is operational in the network.

See Also
- Understanding PIM Snooping for VPLS on page 886

CHAPTER 26

Configuring Multicast Routing Options

Examples: Configuring Administrative Scoping

Understanding Multicast Administrative Scoping

You use multicast scoping to limit multicast traffic by configuring it to an administratively defined topological region. Multicast scoping controls the propagation of multicast messages—both multicast group join messages that are sent upstream toward a source and data forwarding downstream. Scoping can relieve stress on scarce resources, such as bandwidth, and improve privacy or scaling properties.

IP multicast implementations can achieve some level of scoping by using the time-to-live (TTL) field in the IP header. However, TTL scoping has proven difficult to implement reliably, and the resulting schemes often are complex and difficult to understand.

Administratively scoped IP multicast provides clearer and simpler semantics for multicast scoping. Packets addressed to administratively scoped multicast addresses do not cross configured administrative boundaries. Administratively scoped multicast addresses are locally assigned, and hence are not required to be unique across administrative boundaries.

The administratively scoped IP version 4 (IPv4) multicast address space is the range from 239.0.0.0 through 239.255.255.255.

The structure of the IPv4 administratively scoped multicast space is based loosely on the IP version 6 (IPv6) addressing architecture described in RFC 1884, *IP Version 6 Addressing Architecture*.

There are two well-known scopes:

- IPv4 local scope—This scope comprises addresses in the range 239.255.0.0/16. The local scope is the minimal enclosing scope and is not further divisible. Although the exact extent of a local scope is site-dependent, locally scoped regions must not span any other scope boundary and must be contained completely within or be equal to any larger scope. If scope regions overlap in an area, the area of overlap must be within the local scope.
- IPv4 organization local scope—This scope comprises 239.192.0.0/14. It is the space from which an organization allocates subranges when defining scopes for private use.

The ranges 239.0.0.0/10, 239.64.0.0/10, and 239.128.0.0/10 are unassigned and available for expansion of this space.

Two other scope classes already exist in IPv4 multicast space: the statically assigned link-local scope, which is 224.0.0.0/24, and the static global scope allocations, which contain various addresses.

All scoping is inherently bidirectional in the sense that join messages and data forwarding are controlled in both directions on the scoped interface.

You can configure multicast scoping either by creating a named scope associated with a set of routing device interfaces and an address range, or by referencing a scope policy that specifies the interfaces and configures the address range as a series of filters. You cannot combine the two methods (the commit operation fails for a configuration that includes both). The methods differ somewhat in their requirements and result in different output from the **show multicast scope** command. For details and configuration instructions, see and .

Routing loops must be avoided in IP multicast networks. Because multicast routers must replicate packets for each downstream branch, not only do looping packets not arrive at a destination, but each pass around the loop multiplies the number of looping packets, eventually overwhelming the network.

Scoping limits the routers and interfaces that can be used to forward a multicast packet. Scoping can use the TTL field in the IP packet header, but TTL scoping depends on the administrator having a thorough knowledge of the network topology. This topology can change as links fail and are restored, making TTL scoping a poor solution for multicast.

Multicast scoping is administrative in the sense that a range of multicast addresses is reserved for scoping purposes, as described in RFC 2365. Routers at the boundary must be able to filter multicast packets and make sure that the packets do not stray beyond the established limit.

Administrative scoping is much better than TTL scoping, but in many cases the dropping of administratively scoped packets is still determined by the network administrator. For example, the multicast address range 239/8 is defined in RFC 2365 as administratively scoped, and packets using this range are not to be forwarded beyond a network "boundary," usually a routing domain. But only the network administrator knows where the border routers are and can implement the scoping correctly.

Multicast groups used by unicast routing protocols, such as 224.0.0.5 for all OSPF routers, are administratively scoped for that LAN only. This scoping allows the same multicast address to be used without conflict on every LAN running OSPF.

See Also

- Example: Creating a Named Scope for Multicast Scoping on page 901
- Example: Using a Scope Policy for Multicast Scoping on page 903
- Supported IP Multicast Protocol Standards on page 19 in *Standards Reference*

Example: Creating a Named Scope for Multicast Scoping

This example shows how to configure multicast scoping with four scopes: **local**, **organization**, **engineering**, and **marketing**.

Requirements

Before you begin:

- Configure a tunnel interface. See the *Junos OS Network Interfaces Library for Routing Devices*.
- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.

Overview

The **local** scope is configured on a GRE tunnel interface. The **organization** scope is configured on a GRE tunnel interface and a SONET/SDH interface. The **engineering** scope is configured on an IP-IP tunnel interface and two SONET/SDH interfaces. The **marketing** scope is configured on a GRE tunnel interface and two SONET/SDH interfaces. The Junos OS can scope any user-configurable IPv6 or IPv4 group.

To configure multicast scoping by defining a named scope, you must specify a name for the scope, the set of routing device interfaces on which you are configuring scoping, and the scope's address range.

NOTE: The prefix specified with the **prefix** statement must be unique for each **scope** statement. If multiple scopes contain the same prefix, only the last scope applies to the interfaces. If you need to scope the same prefix on multiple interfaces, list all of them in the **interface** statement for a single **scope** statement.

When you configure multicast scoping with a named scope, all scope boundaries must include the **local** scope. If this scope is not configured, it is added automatically at all scoped interfaces. The **local** scope limits the use of the multicast group **239.255.0.0/16** to an attached LAN.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options multicast scope local prefix fe00::239.255.0.0/128
set routing-options multicast scope local interface gr-2/1/0.0
set routing-options multicast scope organization prefix 239.192.0.0/14
set routing-options multicast scope organization interface gr-2/1/0.0
set routing-options multicast scope organization interface so-0/0/0.0
set routing-options multicast scope engineering prefix 239.255.255.0/24
set routing-options multicast scope engineering interface ip-2/1/0.0
set routing-options multicast scope engineering interface so-0/0/1.0
set routing-options multicast scope engineering interface so-0/0/2.0
set routing-options multicast scope marketing prefix 239.255.254.0/24
set routing-options multicast scope marketing interface gr-2/1/0.0
set routing-options multicast scope marketing interface so-0/0/2.0
set routing-options multicast scope marketing interface so-1/0/0.0
```

Step-by-Step Procedure

1. The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

   Configure the local scope.

   ```
   [edit routing-options multicast]
   user@host# set scope local interface gr-2/1/0
   user@host# set scope localprefix fe00::239.255.0.0/128
   ```

2. Configure the organization scope.

   ```
   [edit routing-options multicast]
   user@host# set scope organization interface [ gr-2/1/0 so-0/0/0 ]
   user@host# set scope organization prefix 239.192.0.0/14
   ```

3. Configure the engineering scope.

   ```
   [edit routing-options multicast]
   user@host# set scope engineering interface [ ip-2/1/0 so-0/0/1 so-0/0/2 ]
   user@host# set scope engineering prefix 239.255.255.0/24
   ```

4. Configure the marketing scope.

   ```
   [edit routing-options multicast]
   user@host# set scope marketing interface [ gr-2/1/0 so-0/0/2 so-1/0/0 ]
   user@host# set scope marketing prefix 239.255.254.0/24
   ```

5. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show routing-options** command.

```
user@host# show routing-options
multicast {
  scope local {
    interface gr-2/1/0;
    prefix fe00::239.255.0.0/128;
  }
  scope organization {
    interface [ gr-2/1/0 so-0/0/0 ];
    prefix 239.192.0.0/14;
  }
  scope engineering {
    interface [ ip-2/1/0 so-0/0/1 so-0/0/2 ];
    prefix 239.255.255.0/24;
  }
  scope marketing {
    interface [ gr-2/1/0 so-0/0/2 so-1/0/0 ];
    prefix 239.255.254.0/24;
  }
```

Verification

To verify that group scoping is in effect, issue the show multicast scope command:

```
user@host> show multicast scope
Resolve
Scope name      Group prefix          Interface      Rejects
local           fe00::239.255.0.0/128 gr-2/1/00
organization    239.192.0.0/14        gr-2/1/0       so-0/0/00
engineering     239.255.255.0/24      ip-2/1/0       so-0/0/1 so-0/0/20
marketing       239.255.254.0/24      gr-2/1/0       so-0/0/2 so-1/0/00
```

When you configure scoping with a named scope, the **show multicast scope** operational mode command displays the names of the defined scopes, prefixes, and interfaces.

See Also

- Example: Using a Scope Policy for Multicast Scoping on page 903
- Understanding Multicast Administrative Scoping on page 899

Example: Using a Scope Policy for Multicast Scoping

This example shows how to configure a multicast scope policy named **allow-auto-rp-on-backbone**, allowing packets for auto-RP groups 224.0.1.39/32 and

224.0.1.40/32 on backbone-facing interfaces, and rejecting all other addresses in the 224.0.1.0/24 and 239.0.0.0/8 address ranges.

Requirements

Before you begin:

- Configure an interior gateway protocol or static routing. See the *Junos OS Routing Protocols Library*.

Overview

Each referenced policy must be correctly configured at the **[edit policy-options]** hierarchy level, specifying the set of routing device interfaces on which to configure scoping, and defining the scope's address range as a series of route filters. Only the **interface**, **route-filter**, and **prefix-list** match conditions are supported for multicast scope policies. All other configured match conditions are ignored. The only actions supported are **accept**, **reject**, and the policy flow actions **next-term** and **next-policy**. The **reject** action means that joins and multicast forwarding are suppressed in both directions on the configured interfaces. The **accept** action allows joins and multicast forwarding in both directions on the interface. By default, scope policies apply to all interfaces. The default action is **accept**.

NOTE: Multicast scoping configured with a scope policy differs in some ways from scoping configured with a named scope (which uses the scope statement):

- You cannot apply a scope policy to a specific routing instance, because all scope policies apply to all routing instances. In contrast, a named scope does apply individually to a specific routing instance.
- In contrast to scoping with a named scope, scoping with a scope policy does not automatically add the local scope at scope boundaries. You must explicitly configure the local scope boundaries. The local scope limits the use of the multicast group 239.255.0.0/16 to an attached LAN.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set policy-options policy-statement allow-auto-rp-on-backbone term allow-auto-rp from
    interface so-0/0/0.0
```

```
set policy-options policy-statement allow-auto-rp-on-backbone term allow-auto-rp from
  interface so-0/0/1.0
set policy-options policy-statement allow-auto-rp-on-backbone term allow-auto-rp from
  route-filter 224.0.1.39/32 exact
set policy-options policy-statement allow-auto-rp-on-backbone term allow-auto-rp from
  route-filter 224.0.1.40/32 exact
set policy-options policy-statement allow-auto-rp-on-backbone term allow-auto-rp then
  accept
set policy-options policy-statement allow-auto-rp-on-backbone term reject-these from
  route-filter 224.0.1.0/24 orlonger
set policy-options policy-statement allow-auto-rp-on-backbone term reject-these from
  route-filter 239.0.0.0/8 orlonger
set policy-options policy-statement allow-auto-rp-on-backbone term reject-these then
  reject
set routing-options multicast scope-policy allow-auto-rp-on-backbone
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

1. Define which packets are allowed.

```
[edit policy-options policy-statement allow-auto-rp-on-backbone]
user@host# set term allow-auto-rp from interface so-0/0/0.0
user@host# set term allow-auto-rp from interface so-0/0/1.0
user@host# set term allow-auto-rp from route-filter 224.0.1.39/32 exact
user@host# set term allow-auto-rp from route-filter 224.0.1.40/32 exact
user@host# set term allow-auto-rp then accept
```

2. Define which packets are not allowed.

```
[edit policy-options policy-statement allow-auto-rp-on-backbone]
user@host# set term reject-these from route-filter 224.0.1.0/24 orlonger
user@host# set term reject-these from route-filter 239.0.0.0/8 orlonger
user@host# set term reject-these then reject
```

3. Apply the policy.

```
[edit routing-options multicast]
user@host# set scope-policy allow-auto-rp-on-backbone
```

4. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show policy-options** and **show routing-options** commands.

```
user@host# show policy-options
policy-statement allow-auto-rp-on-backbone {
```

```
    term allow-auto-rp {
      from {
        /* backbone-facing interfaces */
        interface [ so-0/0/0.0 so-0/0/1.0 ];
        route-filter 224.0.1.39/32 exact;
        route-filter 224.0.1.40/32 exact;
      }
      then {
        accept;
      }
    }
    term reject-these {
      from {
        route-filter 224.0.1.0/24 orlonger;
        route-filter 239.0.0.0/8 orlonger;
      }
      then reject;
    }
  }
```

```
user@host# show routing-options
multicast {
  scope-policy allow-auto-rp-on-backbone;
}
```

Verification

To verify that the scope policy is in effect, issue the **show multicast scope** configuration mode command:

```
user@host> show multicast scope
Scope policy: [ allow-auto-rp-on-backbone ]
```

When you configure multicast scoping with a scope policy, the **show multicast scope** operational mode command displays only the name of the scope policy.

See Also

- Example: Creating a Named Scope for Multicast Scoping on page 901
- Understanding Multicast Administrative Scoping on page 899

Example: Configuring Externally Facing PIM Border Routers

In this example, you add the **scope** statement at the **[edit routing-options multicast]** hierarchy level to prevent auto-RP traffic from “leaking” into or out of your PIM domain. Two scopes defined below, **auto-rp-39** and **auto-rp-40**, are for specific addresses. The **scoped-range** statement defines a group range, thus preventing group traffic from leaking.

```
routing-options {
  multicast {
    scope auto-rp-39 {
      prefix 224.0.1.39/32;
      interface t1-0/0/0.0;
    }
    scope auto-rp-40 {
```

```
            prefix 224.0.1.40/32;
            interface t1-0/0/0.0;
        }
        scope scoped-range {
            prefix 239.0.0.0/8;
            interface t1-0/0/0.0;
        }
    }
}
```

Related Documentation

- Examples: Configuring Bandwidth Management on page 907
- Examples: Configuring the Multicast Forwarding Cache on page 928

Examples: Configuring Bandwidth Management

Understanding Bandwidth Management for Multicast

Bandwidth management enables you to control the multicast flows that leave a multicast interface. This control enables you to better manage your multicast traffic and reduce or eliminate the chances of interface oversubscription or congestion.

Bandwidth management ensures that multicast traffic oversubscription does not occur on an interface. When managing multicast bandwidth, you define the maximum amount of multicast bandwidth that an individual interface can use as well as the bandwidth individual multicast flows use.

For example, the routing software cannot add a flow to an interface if doing so exceeds the allowed bandwidth for that interface. Under these circumstances, the interface is rejected. This rejection, however, does not prevent a multicast protocol (for example, PIM) from sending a join message upstream. Traffic continues to arrive on the router, even though the router is not sending the flow from the expected outgoing interfaces.

You can configure the flow bandwidth statically by specifying a bandwidth value for the flow in bits per second, or you can enable the flow bandwidth to be measured and adaptively changed. When using the adaptive bandwidth option, the routing software queries the statistics for the flows to be measured at 5-second intervals and calculates the bandwidth based on the queries. The routing software uses the maximum value measured within the last minute (that is, the last 12 measuring points) as the flow bandwidth.

For more information, see the following sections:

- Bandwidth Management and PIM Graceful Restart on page 908
- Bandwidth Management and Source Redundancy on page 908
- Logical Systems and Bandwidth Oversubscription on page 908

Bandwidth Management and PIM Graceful Restart

When using PIM graceful restart, after the routing process restarts on the Routing Engine, previously admitted interfaces are always readmitted and the available bandwidth is adjusted on the interfaces. When using the adaptive bandwidth option, the bandwidth measurement is initially based on the configured or default starting bandwidth, which might be inaccurate during the first minute. This means that new flows might be incorrectly rejected or admitted temporarily. You can correct this problem by issuing the **clear multicast bandwidth-admission** operational command.

If PIM graceful restart is not configured, after the routing process restarts, previously admitted or rejected interfaces might be rejected or admitted in an unpredictable manner.

See Also

- **clear multicast bandwidth-admission** in the CLI Explorer

Bandwidth Management and Source Redundancy

When using source redundancy, multiple sources (for example, s1 and s2) might exist for the same destination group (g). However, only one of the sources can actively transmit at any time. In this case, multiple forwarding entries—(s1,g) and (s2,g)—are created after each goes through the admission process.

With redundant sources, unlike unrelated entries, an OIF that is already admitted for one entry—for example, (s1,g)—is automatically admitted for other redundancy entries—for example, (s2,g). The remaining bandwidth on the interface is deducted each time an outbound interface is added, even though only one sender actively transmits. By measuring bandwidth, the bandwidth deducted for the inactive entries is credited back when the router detects no traffic is being transmitted.

For more information about defining redundant sources, see "Example: Configuring a Multicast Flow Map" on page 931.

Logical Systems and Bandwidth Oversubscription

You can manage bandwidth at both the physical and logical interface level. However, if more than one logical system shares the same physical interface, the interface might become oversubscribed. Oversubscription occurs if the total bandwidth of all separately configured maximum bandwidth values for the interfaces on each logical system exceeds the bandwidth of the physical interface.

When displaying interface bandwidth information, a negative available bandwidth value indicates oversubscription on the interface.

Interface bandwidth can become oversubscribed when the configured maximum bandwidth decreases or when some flow bandwidths increase because of a configuration change or an actual increase in the traffic rate.

Interface bandwidth can become available again if one of the following occurs:

- The configured maximum bandwidth increases.
- Some flows are no longer transmitted from interfaces, and bandwidth reserves for them are now available to other flows.
- Some flow bandwidths decrease because of a configuration change or an actual decrease in the traffic rate.

Interfaces that are rejected for a flow because of insufficient bandwidth are not automatically readmitted, even when bandwidth becomes available again. Rejected interfaces have an opportunity to be readmitted when one of the following occurs:

- The multicast routing protocol updates the forwarding entry for the flow after receiving a join, leave, or prune message or after a topology change occurs.
- The multicast routing protocol updates the forwarding entry for the flow due to configuration changes.
- You manually reapply bandwidth management to a specific flow or to all flows using the **clear multicast bandwidth-admission** operational command.

In addition, even if previously available bandwidth is no longer available, already admitted interfaces are not removed until one of the following occurs:

- The multicast routing protocol explicitly removes the interfaces after receiving a leave or prune message or after a topology change occurs.
- You manually reapply bandwidth management to a specific flow or to all flows using the **clear multicast bandwidth-admission** operational command.

See Also

- **clear multicast bandwidth-admission** in the CLI Explorer

Example: Defining Interface Bandwidth Maximums

This example shows you how to configure the maximum bandwidth for a physical or logical interface.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol. See the *Junos OS Routing Protocols Library*.
- Configure a multicast protocol. This feature works with the following multicast protocols:
  - DVMRP
  - PIM-DM
  - PIM-SM
  - PIM-SSM

Overview

The maximum bandwidth setting applies admission control either against the configured interface bandwidth or against the native speed of the underlying interface (when there is no configured bandwidth for the interface).

If you configure several logical interfaces (for example, to support VLANs or PVCs) on the same underlying physical interface, and no bandwidth is configured for the logical interfaces, it is assumed that the logical interfaces all have the same bandwidth as the underlying interface. This can cause oversubscription. To prevent oversubscription, configure bandwidth for the logical interfaces, or configure admission control at the physical interface level.

You only need to define the maximum bandwidth for an interface on which you want to apply bandwidth management. An interface that does not have a defined maximum bandwidth transmits all multicast flows as determined by the multicast protocol that is running on the interface (for example, PIM).

If you specify **maximum-bandwidth** without including a bits-per-second value, admission control is enabled based on the bandwidth configured for the interface. In the following example, admission control is enabled for logical interface unit **200**, and the maximum bandwidth is 20 Mbps. If the bandwidth is not configured on the interface, the maximum bandwidth is the link speed.

```
routing-options {
   multicast {
      interface fe-0/2/0.200 {
         maximum-bandwidth;
      }
interfaces {
   fe-0/2/0 {
      unit 200 {
         bandwidth 20m;
      }
   }
}
```

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces fe-0/2/0 unit 200 bandwidth 20m
set routing-options multicast interface fe-0/2/0.200 maximum-bandwidth
set routing-options multicast interface fe-0/2/1 maximum-bandwidth 60m
set routing-options multicast interface fe-0/2/1.200 maximum-bandwidth 10m
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a bandwidth maximum:

1. Configure the a logical interface bandwidth.

   ```
   [edit interfaces]
   user@host# set fe-0/2/0 unit 200 bandwidth 20m
   ```

2. Enable admission control on the logical interface.

   ```
   [edit routing-options]
   user@host# set multicast interface fe-0/2/0.200 maximum-bandwidth
   ```

3. On a physical interface, enable admission control and set the maximum bandwidth to 60 Mbps.

   ```
   [edit routing-options]
   user@host# set multicast interface fe-0/2/1 maximum-bandwidth 60m
   ```

4. For a logical interface on the same physical interface shown in Step 3, set a smaller maximum bandwidth.

   ```
   [edit routing-options]
   user@host# set multicast interface fe-0/2/1.200 maximum-bandwidth 10m
   ```

Results

Confirm your configuration by entering the **show interfaces** and **show routing-options** commands.

```
user@host# show interfaces
fe-0/2/0 {
    unit 200 {
        bandwidth 20m;
    }
}
```

```
user@host# show routing-options
multicast {
  interface fe-0/2/0.200 {
    maximum-bandwidth;
  }
  interface fe-0/2/1 {
    maximum-bandwidth 60m;
  }
  interface fe-0/2/1.200 {
    maximum-bandwidth 10m;
  }
}
```

Verification

To verify the configuration, run the **show multicast interface** command.

See Also

- Example: Configuring a Multicast Flow Map on page 931
- Understanding Bandwidth Management for Multicast on page 907

Example: Configuring Multicast with Subscriber VLANs

This example shows how to configure an MX Series router to function as a broadband service router (BSR).

Requirements

This example uses the following hardware components:

- One MX Series router or EX Series switch with a PIC that supports traffic control profile queuing
- One DSLAM

Before you begin:

- Configure an interior gateway protocol. See the *Junos OS Routing Protocols Library*.
- Configure PIM and IGMP or MLD on the interfaces.

Overview and Topology

When multiple BSR interfaces receive IGMP and MLD join and leave requests for the same multicast stream, the BSR sends a copy of the multicast stream on each interface. Both the multicast control packets (IGMP and MLD) and the multicast data packets flow on the same BSR interface, along with the unicast data. Because all per-customer traffic has its own interface on the BSR, per-customer accounting, call admission control (CAC),

and quality-of-service (QoS) adjustment are supported. The QoS bandwidth used by multicast reduces the unicast bandwidth.

Multiple interfaces on the BSR might connect to a shared device (for example, a DSLAM). The BSR sends the same multicast stream multiple times to the shared device, thus wasting bandwidth. It is more efficient to send the multicast stream one time to the DSLAM and replicate the multicast streams in the DSLAM. There are two approaches that you can use.

The first approach is to continue to send unicast data on the per-customer interfaces, but have the DSLAM route all the per-customer IGMP and MLD join and leave requests to the BSR on a single dedicated interface (a multicast VLAN). The DSLAM receives the multicast streams from the BSR on the dedicated interface with no unnecessary replication and performs the necessary replication to the customers. Because all multicast control and data packets use only one interface, only one copy of a stream is sent even if there are multiple requests. This approach is called reverse outgoing interface (OIF) mapping. Reverse OIF mapping enables the BSR to propagate the multicast state of the shared interface to the customer interfaces, which enables per-customer accounting and QoS adjustment to work. When a customer changes the TV channel, the router gateway (RG) sends an IGMP or MLD join and leave messages to the DSLAM. The DSLAM transparently passes the request to the BSR through the multicast VLAN. The BSR maps the IGMP or MLD request to one of the subscriber VLANs based on the IP source address or the source MAC address. When the subscriber VLAN is found, QoS adjustment and accounting are perfomed on that VLAN or interface.

The second approach is for the DSLAM to continue to send unicast data and all the per-customer IGMP and MLD join and leave requests to the BSR on the individual customer interfaces, but to have the multicast streams arrive on a single dedicated interface. If multiple customers request the same multicast stream, the BSR sends one copy of the data on the dedicated interface. The DSLAM receives the multicast streams from the BSR on the dedicated interface and performs the necessary replication to the customers. Because the multicast control packets use many customer interfaces, configuration on the BSR must specify how to map each customer's multicast data packets to the single dedicated output interface. QoS adjustment is supported on the customer interfaces. CAC is supported on the shared interface. This second approach is called multicast OIF mapping.

OIF mapping and reverse OIF mapping are not supported on the same customer interface or shared interface. This example shows how to configure the two different approaches. Both approaches support QoS adjustment, and both approaches support MLD/IPv6. The reverse OIF mapping example focuses on IGMP/IPv4 and enables QoS adjustment. The OIF mapping example focuses on MLD/IPv6 and disables QoS adjustment.

The first approach (reverse OIF mapping) includes the following statements:

- **flow-map**—Defines a flow map that controls the bandwidth for each flow.
- **maximum-bandwidth**—Enables CAC.
- **reverse-oif-mapping**—Enables the routing device to identify a subscriber VLAN or interface based on an IGMP or MLD join or leave request that it receives over the multicast VLAN.

After the subscriber VLAN is identified, the routing device immediately adjusts the QoS (in this case, the bandwidth) on that VLAN based on the addition or removal of a subscriber.

The routing device uses IGMP and MLD join or leave reports to obtain the subscriber VLAN information. This means that the connecting equipment (for example, the DSLAM) must forward all IGMP and MLD reports to the routing device for this feature to function properly. Using report suppression or an IGMP proxy can result in reverse OIF mapping not working properly.

- **subscriber-leave-timer**—Introduces a delay to the QoS update. After receiving an IGMP or MLD leave request, this statement defines a time delay (between 1 and 30 seconds) that the routing device waits before updating the QoS for the remaining subscriber interfaces. You might use this delay to decrease how often the routing device adjusts the overall QoS bandwidth on the VLAN when a subscriber sends rapid leave and join messages (for example, when changing channels in an IPTV network).
- **traffic-control-profile**—Configures a shaping rate on the logical interface. The configured shaping rate must be configured as an absolute value, not as a percentage.

The second approach (OIF mapping) includes the following statements:

- **map-to-interface**—In a policy statement, enables you to build the OIF map.

  The OIF map is a routing policy statement that can contain multiple terms. When creating OIF maps, keep the following in mind:

  - If you specify a physical interface (for example, **ge-0/0/0**), a ".0" is appended to the interface to create a logical interface (for example, **ge-0/0/0.0**).
  - Configure a routing policy for each logical system. You cannot configure routing policies dynamically.
  - The interface must also have IGMP, MLD, or PIM configured.
  - You cannot map to a mapped interface.
  - We recommend that you configure policy statements for IGMP and MLD separately.
  - Specify either a logical interface or the keyword **self**. The **self** keyword specifies that multicast data packets be sent on the same interface as the control packets and that no mapping occur. If no term matches, then no multicast data packets are sent.

- **no-qos-adjust**—Disables QoS adjustment.

  QoS adjustment decreases the available bandwidth on the client interface by the amount of bandwidth consumed by the multicast streams that are mapped from the client interface to the shared interface. This action always occurs unless it is explicitly disabled.

  If you disable QoS adjustment, available bandwidth is not reduced on the customer interface when multicast streams are added to the shared interface.

NOTE: You can dynamically disable QoS adjustment for IGMP and MLD interfaces using dynamic profiles.

- **oif-map**—Associate a map with an IGMP or MLD interface. The OIF map is then applied to all IGMP or MLD requests received on the configured interface. In this example, subscriber VLANs 1 and 2 have MLD configured, and each VLAN points to an OIF map that directs some traffic to **ge-2/3/9.4000**, some traffic to **ge-2/3/9.4001**, and some traffic to **self**.

  NOTE: You can dynamically associate OIF maps with IGMP interfaces using dynamic profiles.

- **passive**—Defines either IGMP or MLD to use passive mode.

  The OIF map interface should not typically pass IGMP or MLD control traffic and should be configured as passive. However, the OIF map implementation does support running IGMP or MLD on an interface (control and data) in addition to mapping data streams to the same interface. In this case, you should configure IGMP or MLD normally (that is, not in passive mode) on the mapped interface. In this example, the OIF map interfaces (**ge-2/3/9.4000** and **ge-2/3/9.4001**) are configured as MLD passive.

  By default, specifying the **passive** statement means that no general queries, group-specific queries, or group-source-specific queries are sent over the interface and that all received control traffic is ignored by the interface. However, you can selectively activate up to two out of the three available options for the **passive** statement while keeping the other functions passive (inactive).

  These options include the following:

  - **send-general-query**—When specified, the interface sends general queries.
  - **send-group-query**—When specified, the interface sends group-specific and group-source-specific queries.
  - **allow-receive**—When specified, the interface receives control traffic.

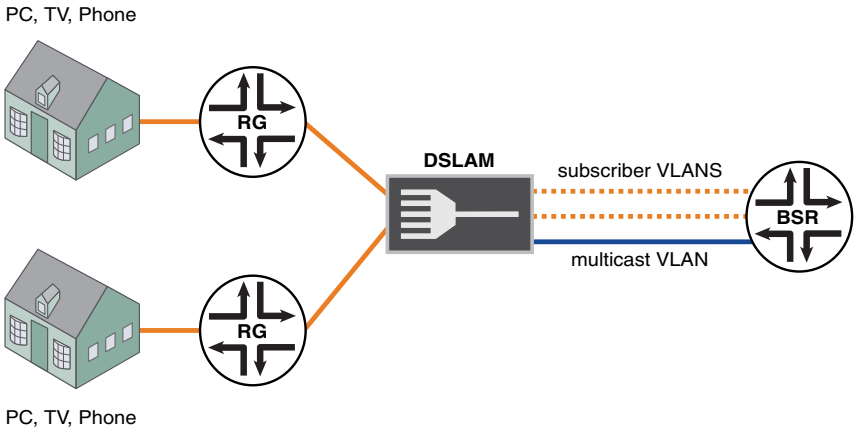
Figure 128 on page 916 shows the scenario.

In both approaches, if multiple customers request the same multicast stream, the BSR sends one copy of the stream on the shared multicast VLAN interface. The DSLAM receives the multicast stream from the BSR on the shared interface and performs the necessary replication to the customers.

In the first approach (reverse OIF mapping), the DSLAM uses the per-customer subscriber VLANs for unicast data only. IGMP and MLD join and leave requests are sent on the multicast VLAN.

In the second approach (OIF mapping), the DSLAM uses the per-customer subscriber VLANs for unicast data and for IGMP and MLD join and leave requests. The multicast VLAN is used only for multicast streams, not for join and leave requests.

Figure 128: Multicast with Subscriber VLANs



RG: Router Gateway
BSR: Broadband Service Router

Configuration

Configuring a Reverse OIF Map

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode. .

```
set class-of-service traffic-control-profiles tcp-ifl shaping-rate 20m
set class-of-service interfaces ge-2/2/0 shaping-rate 240m
set class-of-service interfaces ge-2/2/0 unit 50 output-traffic-control-profile tcp-ifl
set class-of-service interfaces ge-2/2/0 unit 51 output-traffic-control-profile tcp-ifl
set interfaces ge-2/0/0 unit 0 family inet address 30.0.0.2/24
set interfaces ge-2/2/0 hierarchical-scheduler
set interfaces ge-2/2/0 vlan-tagging
set interfaces ge-2/2/0 unit 10 vlan-id 10
set interfaces ge-2/2/0 unit 10 family inet address 40.0.0.2/24
set interfaces ge-2/2/0 unit 50 vlan-id 50
set interfaces ge-2/2/0 unit 50 family inet address 50.0.0.2/24
set interfaces ge-2/2/0 unit 51 vlan-id 51
set interfaces ge-2/2/0 unit 51 family inet address 50.0.1.2/24
set policy-options policy-statement all-mcast-groups from source-address-filter
  30.0.0.0/8 orlonger
set policy-options policy-statement all-mcast-groups then accept
set protocols igmp interface all
set protocols igmp interface fxp0.0 disable
set protocols pim rp local address 20.0.0.2
set protocols pim interface all
set protocols pim interface fxp0.0 disable
set protocols pim interface ge-2/2/0.10 disable
set routing-options multicast flow-map map1 policy all-mcast-groups
set routing-options multicast flow-map map1 bandwidth 10m
set routing-options multicast flow-map map1 bandwidth adaptive
set routing-options multicast interface ge-2/2/0.10 maximum-bandwidth 500m
```

```
set routing-options multicast interface ge-2/2/0.10 reverse-oif-mapping
set routing-options multicast interface ge-2/2/0.10 subscriber-leave-timer 20
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure reverse OIF mapping:

1. Configure a logical interface for unicast data traffic.

```
[edit interfaces ge-2/0/0]
user@host# set unit 0 family inet address 30.0.0.2/24
```

2. Configure a logical interface for subscriber control traffic.

```
[edit interfaces ge-2/2/0]
user@host# set hierarchical-scheduler
user@host# set vlan-tagging
user@host# set unit 10 vlan-id 10
user@host# set unit 10 family inet address 40.0.0.2/24
```

3. Configure two logical interfaces on which QoS adjustments are made.

```
[edit interfaces ge-2/2/0]
user@host# set unit 50 vlan-id 50
user@host# set unit 50 family inet address 50.0.0.2/24
user@host# set unit 51 vlan-id 51
user@host# set unit 51 family inet address 50.0.1.2/24
```

4. Configure a policy.

```
[edit policy-options policy-statement all-mcast-groups]
user@host# set from source-address-filter 30.0.0.0/8 orlonger
user@host# set then accept
```

5. Enable a flow map that references the policy.

```
[edit routing-options multicast]
user@host# set flow-map map1 policy all-mcast-groups
user@host# set flow-map map1 bandwidth 10m adaptive
```

6. Enable OIF mapping on the logical interface that receives subscriber control traffic.

```
[edit routing-options multicast]
user@host# set interface ge-2/2/0.10 maximum-bandwidth 500m
user@host# set interface ge-2/2/0.10 reverse-oif-mapping
user@host# set interface ge-2/2/0.10 subscriber-leave-timer 20
```

7. Configure PIM and IGMP.

```
[edit protocols]
user@host# set igmp interface all
user@host# set igmp interface fxp0.0 disable
```

```
user@host# set pim rp local address 20.0.0.2
user@host# set pim interface all
user@host# set pim interface fxp0.0 disable
user@host# set pim interface ge-2/2/0.10 disable
```

8. Configure the hierarchical scheduler by configuring a shaping rate for the physical interface and a slower shaping rate for the logical interfaces on which QoS adjustments are made.

```
[edit class-of-service interfaces ge-2/2/0]
user@host# set shaping-rate 240m
user@host# set unit 50 output-traffic-control-profile tcp-ifl
user@host# set unit 51 output-traffic-control-profile tcp-ifl

[edit class-of-service traffic-control-profiles tcp-30m-no-smap]
user@host# set shaping-rate 20m
```

Results From configuration mode, confirm your configuration by entering the **show class-of-service**, **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show class-of-service
traffic-control-profiles {
    tcp-ifl {
        shaping-rate 20m;
    }
}
interfaces {
    ge-2/2/0 {
        shaping-rate 240m;
        unit 50 {
            output-traffic-control-profile tcp-ifl;
        }
        unit 51 {
            output-traffic-control-profile tcp-ifl;
        }
    }
}
```

```
user@host# show interfaces
ge-2/0/0 {
    unit 0 {
        family inet {
            address 30.0.0.2/24;
        }
    }
}
ge-2/2/0 {
    hierarchical-scheduler;
    vlan-tagging;
    unit 10 {
        vlan-id 10;
```

```
      family inet {
        address 40.0.0.2/24;
      }
    }
    unit 50 {
      vlan-id 50;
      family inet {
        address 50.0.0.2/24;
      }
    }
    unit 51 {
      vlan-id 51;
      family inet {
        address 50.0.1.2/24;
      }
    }
}
```

```
user@host# show policy-options
policy-statement all-mcast-groups {
  from {
    source-address-filter 30.0.0.0/8 orlonger;
  }
  then accept;
}
```

```
user@host# show protocols
igmp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
pim {
  rp {
    local {
      address 20.0.0.2;
    }
  }
  interface all;
  interface fxp0.0 {
    disable;
  }
  interface ge-2/2/0.10 {
    disable;
  }
}
```

```
user@host# show routing-options
multicast {
  flow-map map1 {
    policy all-mcast-groups;
    bandwidth 10m adaptive;
  }
  interface ge-2/2/0.10 {
    maximum-bandwidth 500m;
    reverse-oif-mapping;
```

```
        subscriber-leave-timer 20;
    }
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Configuring an OIF Map

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set interfaces ge-2/3/8 unit 0 family inet6 address C300:0101::/24
set interfaces ge-2/3/9 vlan-tagging
set interfaces ge-2/3/9 unit 1 vlan-id 1
set interfaces ge-2/3/9 unit 1 family inet6 address C400:0101::/24
set interfaces ge-2/3/9 unit 2 vlan-id 2
set interfaces ge-2/3/9 unit 2 family inet6 address C400:0201::/24
set interfaces ge-2/3/9 unit 4000 vlan-id 4000
set interfaces ge-2/3/9 unit 4000 family inet6 address C40F:A001::/24
set interfaces ge-2/3/9 unit 4001 vlan-id 4001
set interfaces ge-2/3/9 unit 4001 family inet6 address C40F:A101::/24
set policy-options policy-statement g539-v6 term g539-4000 from route-filter
  FF05:0101:0000::/39 orlonger
set policy-options policy-statement g539-v6 term g539-4000 then map-to-interface
  ge-2/3/9.4000
set policy-options policy-statement g539-v6 term g539-4000 then accept
set policy-options policy-statement g539-v6 term g539-4001 from route-filter
  FF05:0101:0200::/39 orlonger
set policy-options policy-statement g539-v6 term g539-4001 then map-to-interface
  ge-2/3/9.4001
set policy-options policy-statement g539-v6 term g539-4001 then accept
set policy-options policy-statement g539-v6 term self from route-filter
  FF05:0101:0700::/40 orlonger
set policy-options policy-statement g539-v6 term self then map-to-interface self
set policy-options policy-statement g539-v6 term self then accept
set policy-options policy-statement g539-v6-all term g539 from route-filter 0::/0 orlonger
set policy-options policy-statement g539-v6-all term g539 then map-to-interface
  ge-2/3/9.4000
set policy-options policy-statement g539-v6-all term g539 then accept
set protocols mld interface fxp0.0 disable
set protocols mld interface ge-2/3/9.4000 passive
set protocols mld interface ge-2/3/9.4001 passive
set protocols mld interface ge-2/3/9.1 version 1
set protocols mld interface ge-2/3/9.1 oif-map g539-v6
set protocols mld interface ge-2/3/9.2 version 2
set protocols mld interface ge-2/3/9.2 oif-map g539-v6
set protocols pim rp local address 20.0.0.4
set protocols pim rp local family inet6 address C000::1
set protocols pim interface ge-2/3/8.0 mode sparse
set protocols pim interface ge-2/3/8.0 version 2
set routing-options multicast interface ge-2/3/9.1 no-qos-adjust
set routing-options multicast interface ge-2/3/9.2 no-qos-adjust
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see the *CLI User Guide*.

To configure reverse OIF mapping:

1. Configure a logical interface for unicast data traffic.

```
[edit interfaces ge-2/3/8 ]
user@host# set unit 0 family inet6 address C300:0101::/24
```

2. Configure logical interfaces for subscriber VLANs.

```
[edit interfaces ge-2/3/9]
user@host# set vlan-tagging
user@host# set unit 1 vlan-id 1
user@host# set unit 1 family inet6 address C400:0101::/24
user@host# set unit 2 vlan-id 2
user@host# set unit 2 family inet6 address C400:0201::/24 lo0 unit 0 family inet6
  address C000::1/128
user@host# set unit 2 family inet6 address C400:0201::/24
```

3. Configure two map-to logical interfaces.

```
[edit interfaces ge-2/2/0]
user@host# set unit 4000 vlan-id 4000
user@host# set unit 4000 family inet6 address C40F:A001::/24
user@host# set unit 4001 vlan-id 4001
user@host# set unit 4001 family inet6 address C40F:A101::/24
```

4. Configure the OIF map.

```
[edit policy-options policy-statement g539-v6]
user@host# set term g539-4000 from route-filter FF05:0101:0000::/39 orlonger
user@host# set then map-to-interface ge-2/3/9.4000
user@host# set then accept
user@host# set term g539-4001 from route-filter FF05:0101:0200::/39 orlonger
user@host# set then map-to-interface ge-2/3/9.4001
user@host# set then accept
user@host# set term self from route-filter FF05:0101:0700::/40 orlonger
user@host# set then map-to-interface self
user@host# set then accept

[edit policy-options policy-statement g539-v6-all]
user@host# set term g539 from route-filter 0::/0 orlonger
user@host# set then map-to-interface ge-2/3/9.4000
user@host# set then accept
```

5. Disable QoS adjustment on the subscriber VLANs.

```
[edit routing-options multicast]
user@host# set interface ge-2/3/9.1 no-qos-adjust
user@host# set interface ge-2/3/9.2 no-qos-adjust
```

6. Configure PIM and MLD. Point the MLD subscriber VLANs to the OIF map.

```
[edit protocols]
user@host# set pim rp local address 20.0.0.4
user@host# set pim rp local family inet6 address C000::1 #C000::1 is the address
  of lo0
user@host# set pim interface ge-2/3/8.0 mode sparse
user@host# set pim interface ge-2/3/8.0 version 2
user@host# set mld interface fxp0.0 disable
user@host# set interface ge-2/3/9.4000 passive
user@host# set interface ge-2/3/9.4001 passive
user@host# set interface ge-2/3/9.1 version 1
user@host# set interface ge-2/3/9.1 oif-map g539-v6
user@host# set interface ge-2/3/9.2 version 2
user@host# set interface ge-2/3/9.2 oif-map g539-v6
```

Results From configuration mode, confirm your configuration by entering the **show interfaces**, **show policy-options**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@host# show interfaces
ge-2/3/8 {
  unit 0 {
    family inet6 {
      address C300:0101::/24;
    }
  }
}
ge-2/3/9 {
  vlan-tagging;
  unit 1 {
    vlan-id 1;
    family inet6 {
      address C400:0101::/24;
    }
  }
  unit 2 {
    vlan-id 2;
    family inet6 {
      address C400:0201::/24;
    }
  }
  unit 4000 {
    vlan-id 4000;
    family inet6 {
      address C40F:A001::/24;
    }
  }
  unit 4001 {
    vlan-id 4001;
    family inet6 {
      address C40F:A101::/24;
    }
  }
}
```

```
user@host# show policy-options
policy-statement g539-v6 {
  term g539-4000 {
    from {
      route-filter FF05:0101:0000::/39 orlonger;
    }
    then {
      map-to-interface ge-2/3/9.4000;
      accept;
    }
  }
  term g539-4001 {
    from {
      route-filter FF05:0101:0200::/39 orlonger;
    }
    then {
      map-to-interface ge-2/3/9.4001;
      accept;
    }
  }
  term self {
    from {
      route-filter FF05:0101:0700::/40 orlonger;
    }
    then {
      map-to-interface self;
      accept;
    }
  }
}
policy-statement g539-v6-all {
  term g539 {
    from {
      route-filter 0::/0 orlonger;
    }
    then {
      map-to-interface ge-2/3/9.4000;
      accept;
    }
  }
}

user@host# show protocols
mld {
  interface fxp0.0 {
    disable;
  }
  interface ge-2/3/9.4000 {
    passive;
  }
  interface ge-2/3/9.4001 {
    passive;
  }
  interface ge-2/3/9.1 {
    version 1;
    oif-map g539-v6;
```

```
    }
    interface ge-2/3/9.2 {
      version 2;
      oif-map g539-v6;
    }
  }
  pim {
    rp {
      local {
        address 20.0.0.4;
        family inet6 {
          address C000::1;
        }
      }
    }
    interface ge-2/3/8.0 {
      mode sparse;
      version 2;
    }
  }
```

```
user@host# show routing-options
multicast {
  interface ge-2/3/9.1 no-qos-adjust;
  interface ge-2/3/9.2 no-qos-adjust;
}
```

If you are done configuring the device, enter **commit** from configuration mode.

Verification

To verify the configuration, run the following commands:

- **show igmp statistics**
- **show class-of-service interface**
- **show interfaces statistics**
- **show mld statistics**
- **show multicast interface**
- **show policy**

See Also

- Example: Configuring a Multicast Flow Map on page 931
- Configuring Multicast Routing over IP Demux Interfaces on page 925

Configuring Multicast Routing over IP Demux Interfaces

In a subscriber management network, fields in packets sent from IP demux interfaces are intended to correspond to a specific client that resides on the other side of an aggregation device (for example, a Multiservice Access Node [MSAN]). However, packets sent from a Broadband Services Router (BSR) to an MSAN do not identify the demux interface. Once it obtains a packet, it is up to the MSAN device to determine which client receives the packet.

Depending on the intelligence of the MSAN device, determining which client receives the packet can occur in an inefficient manner. For example, when it receives IGMP control traffic, an MSAN might forward the control traffic to all clients instead of the one intended client. In addition, once a data stream destination is established, though an MSAN can use IGMP snooping to determine which hosts reside in a particular group and limit data streams to only that group, the MSAN still must send multiple copies of the data stream to each group member, even if that data stream is intended for only one client in the group.

Various multicast features, when combined, enable you to avoid the inefficiencies mentioned above. These features include the following:

- The ability to configure the IP demux interface **family** statement to use **inet** for either the numbered or unnumbered primary interface.
- The ability to configure IGMP on the primary interface to send general queries for all clients. The demux configuration prevents the primary IGMP interface from receiving any client IGMP control packets. Instead, all IGMP control packets go to the demux interfaces. However, to guarantee that no joins occur on the primary interface:
  - For static IGMP interfaces—Include the **passive send-general-query** statement in the IGMP configuration at the **[edit protocols igmp interface *interface-name*]** hierarchy level.
  - For dynamic IGMP demux interfaces—Include the **passive send-general-query** statement at the **[edit dynamic-profiles *profile-name* protocols igmp interface *interface-name*]** hierarchy level.
- The ability to map all multicast groups to the primary interface as follows:
  - For static IGMP interfaces—Include the **oif-map** statement at the **[edit protocols igmp interface *interface-name*]** hierarchy level.
  - For dynamic IGMP demux interfaces—Include the **oif-map** statement at the **[edit dynamic-profiles *profile-name* protocols igmp interface *interface-name*]** hierarchy level.

Using the **oif-map** statement, you can map the same IGMP group to the same output interface and send only one copy of the multicast stream from the interface.

- The ability to configure IGMP on each demux interface. To prevent duplicate general queries:
  - For static IGMP interfaces—Include the **passive allow-receive send-group-query** statement at the **[edit protocols igmp interface *interface-name*]** hierarchy level.
  - For dynamic demux interfaces—Include the **passive allow-receive send-group-query** statement at the **[edit dynamic-profiles *profile-name* protocols igmp interface *interface-name*]** hierarchy level.

NOTE: **To send only one copy of each group, regardless of how many customers join, use the oif-map statement as previously mentioned.**

See Also
- Example: Configuring Multicast with Subscriber VLANs on page 912
- *Junos OS Broadband Subscriber Management and Services Library*

Classifying Packets by Egress Interface

For Juniper Networks M320 Multiservice Edge Routers and T Series Core Routers with the Intelligent Queuing (IQ), IQ2, Enhanced IQ (IQE), Multiservices link services intelligent queuing (LSQ) interfaces, or ATM2 PICs, you can classify unicast and multicast packets based on the egress interface. For unicast traffic, you can also use a multifield filter, but only egress interface classification applies to multicast traffic as well as unicast traffic. If you configure egress classification of an interface, you cannot perform Differentiated Services code point (DSCP) rewrites on the interface. By default, the system does not perform any classification based on the egress interface.

On an MX Series router that contains MPCs and MS-DPCs, multicast packets are dropped on the router and not processed properly if the router contains MLPPP LSQ logical interfaces that function as multicast receivers and if the network services mode is configured as enhanced IP mode on the router. This behavior is expected with LSQ interfaces in conjunction with enhanced IP mode. In such a scenario, if enhanced IP mode is not configured, multicasting works correctly. However, if the router contains redundant LSQ interfaces and enhanced IP network services mode configured with FIB localization, multicast works properly.

To enable packet classification by the egress interface, you first configure a forwarding class map and one or more queue numbers for the egress interface at the **[edit class-of-service forwarding-class-map *forwarding-class-map-name*]** hierarchy level:

```
[edit class-of-service]
forwarding-classes-interface-specific  forwarding-class-map-name {
   class class-name queue-num queue-number [ restricted-queue queue-number ];
}
```

For T Series routers that are restricted to only four queues, you can control the queue assignment with the **restricted-queue** option, or you can allow the system to automatically determine the queue in a modular fashion. For example, a map assigning packets to queue 6 would map to queue 2 on a four-queue system.

NOTE: If you configure an output forwarding class map associating a forwarding class with a queue number, this map is not supported on multiservices link services intelligent queuing (lsq-) interfaces.

Once the forwarding class map has been configured, you apply the map to the logical interface by using the **output-forwarding-class-map** statement at the **[edit class-of-service interfaces *interface-name* unit *logical-unit-number*]** hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
output-forwarding-class-map forwarding-class-map-name;
```

All parameters relating to the queues and forwarding class must be configured as well. For more information about configuring forwarding classes and queues, see *Configuring a Custom Forwarding Class for Each Queue*.

This example shows how to configure an interface-specific forwarding-class map named **FCMAP1** that restricts queues 5 and 6 to different queues on four-queue systems and then applies **FCMAP1** to **unit 0** of interface **ge-6/0/0**:

```
[edit class-of-service]
forwarding-class-map FCMAP1 {
   class FC1 queue-num 6 restricted-queue 3;
   class FC2 queue-num 5 restricted-queue 2;
   class FC3 queue-num 3;
   class FC4 queue-num 0;
   class FC3 queue-num 0;
   class FC4 queue-num 1;
}

[edit class-of-service]
interfaces {
   ge-6/0/0 unit 0 {
      output-forwarding-class-map FCMAP1;
   }
}
```

Note that without the **restricted-queue** option in **FCMAP1**, the example would assign **FC1** and **FC2** to queues 2 and 1, respectively, on a system restricted to four queues.

Use the **show class-of-service forwarding-class *forwarding-class-map-name*** command to display the forwarding-class map queue configuration:

```
user@host> show class-of-service forwarding-class FCMAP2

Forwarding class                          ID      Queue  Restricted queue
  FC1                                      0       6          3
  FC2                                      1       5          2
  FC3                                      2       3          3
  FC4                                      3       0          0
```

```
  FC5                                4        0          0
  FC6                                5        1          1
  FC7                                6        6          2
  FC8                                7        7          3
```

Use the **show class-of-service interface** *interface-name* command to display the forwarding-class maps (and other information) assigned to a logical interface:

```
user@host> show class-of-service interface ge-6/0/0

Physical interface: ge-6/0/0, Index: 128
Queues supported: 8, Queues in use: 8
 Scheduler map: <default>, Index: 2
 Input scheduler map: <default>, Index: 3
 Chassis scheduler map: <default-chassis>, Index: 4

Logical interface: ge-6/0/0.0, Index: 67
 Object                     Name             Type                 Index
 Scheduler-map              sch-map1         Output                6998
 Scheduler-map              sch-map1         Input                 6998
 Classifier                 dot1p            ieee8021p             4906
 forwarding-class-map       FCMAP1           Output                1221

Logical interface: ge-6/0/0.1, Index 68
 Object                     Name             Type                 Index
 Scheduler-map              <default>        Output                   2
 Scheduler-map              <default>        Input                    3

Logical interface: ge-6/0/0.32767, Index 69
 Object                     Name             Type                 Index
 Scheduler-map              <default>        Output                   2
 Scheduler-map              <default>        Input                    3
```

Related Documentation

- Examples: Configuring Administrative Scoping on page 899
- Examples: Configuring the Multicast Forwarding Cache on page 928

Examples: Configuring the Multicast Forwarding Cache

Understanding the Multicast Forwarding Cache

IP multicast protocols can create numerous entries in the multicast forwarding cache. If the forwarding cache fills up with entries that prevent the addition of higher-priority entries, applications and protocols might not function properly. You can manage the multicast forwarding cache properties by limiting the size of the cache and by controlling the length of time that entries remain in the cache. By managing timeout values, you can give preference to more important forwarding cache entries while removing other less important entries.

Example: Configuring the Multicast Forwarding Cache

When a routing device receives multicast traffic, it places the (S,G) route information in the multicast forwarding cache, **inet.1**. This example shows how to configure multicast forwarding cache limits to prevent the cache from filling up with entries.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol. See the *Junos OS Routing Protocols Library*.
- Configure a multicast protocol. This feature works with the following multicast protocols:
  - DVMRP
  - PIM-DM
  - PIM-SM
  - PIM-SSM

Overview

This example includes the following statements:

- **forwarding-cache**—Specifies how forwarding entries are aged out and how the number of entries is controlled.
- **timeout**—Specifies an idle period after which entries are aged out and removed from **inet.1**. You can specify a timeout in the range from 1 through 720 minutes.
- **threshold**—Enables you to specify threshold values on the forwarding cache to suppress (suspend) entries from being added when the cache entries reach a certain maximum and begin adding entries to the cache when the number falls to another threshold value. By default, no threshold values are enabled on the routing device.

  The suppress threshold suspends the addition of new multicast forwarding cache entries. If you do not specify a suppress value, multicast forwarding cache entries are created as necessary. If you specify a suppress threshold, you can optionally specify a reuse threshold, which sets the point at which the device resumes adding new multicast forwarding cache entries. During suspension, forwarding cache entries time out. After a certain number of entries time out, the reuse threshold is reached, and new entries are added. The range for both thresholds is from 1 through 200,000. If configured, the reuse value must be less than the suppression value. If you do not specify a reuse value, the number of multicast forwarding cache entries is limited to the suppression value.

A new entry is created as soon as the number of multicast forwarding cache entries falls below the suppression value.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set routing-options multicast forwarding-cache threshold suppress 150000
set routing-options multicast forwarding-cache threshold reuse 34
set routing-options multicast forwarding-cache timeout 60
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure the multicast forwarding cache:

1. Configure the maximum size of the forwarding cache.

   ```
   [edit routing-options multicast forwarding-cache]
   user@host# set threshold suppress 150000
   ```

2. Configure the amount of time (in minutes) entries can remain idle before being removed.

   ```
   [edit routing-options multicast forwarding-cache]
   user@host# set timeout 60
   ```

3. Configure the size of the forwarding cache when suppression stops and new entries can be added.

   ```
   [edit routing-options multicast forwarding-cache]
   user@host# set threshold reuse 70000
   ```

Results

Confirm your configuration by entering the **show routing-options** command.

```
user@host# show routing-options
multicast {
    forwarding-cache {
        threshold {
            suppress 150000;
            reuse 70000;
        }
        timeout 60;
    }
}
```

Verification

To verify the configuration, run the show multicast route **extensive** command.

```
user@host> show multicast route extensive
Family: INET
Group: 232.0.0.1
    Source: 11.11.11.11/32
    Upstream interface: fe-0/2/0.200
    Downstream interface list:
        fe-0/2/1.210
    Downstream interface list rejected by CAC:
        fe-0/2/1.220
    Session description: Source specific multicast
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 337
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 60 minutes
Wrong incoming interface notifications: 0
```

See Also

- Example: Configuring a Multicast Flow Map on page 931
- Bandwidth Management and Source Redundancy on page 908
- Understanding Bandwidth Management for Multicast on page 907
- Understanding the Multicast Forwarding Cache on page 928

Example: Configuring a Multicast Flow Map

This example shows how to configure a flow map to prevent certain forwarding cache entries from aging out, thus allowing for faster failover from one source to another. Flow maps enable you to configure bandwidth variables and multicast forwarding cache timeout values for entries defined by the flow map policy.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure an interior gateway protocol. See the *Junos OS Routing Protocols Library*.
- Configure a multicast protocol. This feature works with the following multicast protocols:

- DVMRP
- PIM-DM
- PIM-SM
- PIM-SSM

Overview

Flow maps are typically used for fast multicast source failover when there are multiple sources for the same group. For example, when one video source is actively sending the traffic, the forwarding states for other video sources are timed out after a few minutes. Later, when a new source starts sending the traffic again, it takes time to install a new forwarding state for the new source if the forwarding state is not already there. This switchover delay is worsened when there are many video streams. Using flow maps with longer timeout values or permanent cache entries helps reduce this switchover delay.

NOTE: The permanent forwarding state must exist on all routing devices in the path for fast source switchover to function properly.

This example includes the following statements:

- **bandwidth**—Specifies the bandwidth for each flow that is defined by a flow map to ensure that an interface is not oversubscribed for multicast traffic. If adding one more flow would cause overall bandwidth to exceed the allowed bandwidth for the interface, the request is rejected. A rejected request means that traffic might not be delivered out of some or all of the expected outgoing interfaces. You can define the bandwidth associated with multicast flows that match a flow map by specifying a bandwidth in bits per second or by specifying that the bandwidth is measured and adaptively modified.

  When you use the **adaptive** option, the bandwidth adjusts based on measurements made at 5-second intervals. The flow uses the maximum bandwidth value from the last 12 measured values (1 minute).

  When you configure a bandwidth value with the **adaptive** option, the bandwidth value acts as the starting bandwidth for the flow. The bandwidth then changes based on subsequent measured bandwidth values. If you do not specify a bandwidth value with the **adaptive** option, the starting bandwidth defaults to 2 megabits per second (Mbps).

  For example, the **bandwidth 2m adaptive** statement is equivalent to the **bandwidth adaptive** statement because they both use the same starting bandwidth (2 Mbps, the default). If the actual flow bandwidth is 4 Mbps, the measured flow bandwidth changes to 4 Mbps after reaching the first measuring point (5 seconds). However, if the actual flow bandwidth rate is 1 Mbps, the measured flow bandwidth remains at 2 Mbps for the first 12 measurement cycles (1 minute) and then changes to the measured 1 Mbps value.

- **flow-map**—Defines a flow map that controls the forwarding cache timeout of specified source and group addresses, controls the bandwidth for each flow, and specifies redundant sources. If a flow can match multiple flow maps, the first flow map applies.

- **forwarding-cache**—Enables you to configure the forwarding cache properties of entries defined by a flow map. You can specify a timeout of **never** to make the forwarding entries permanent, or you can specify a timeout in the range from 1 through 720 minutes. If you set the value to **never**, you can specify the **non-discard-entry-only** option to make an exception for entries that are in the pruned state. In other words, the **never non-discard-entry-only** statement allows entries in the pruned state to time out, while entries in the forwarding state never time out.

- **policy**—Specifies source and group addresses to which the flow map applies.

- **redundant-sources**—Specify redundant (backup) sources for flows identified by a flow map. Outbound interfaces that are admitted for one of the forwarding entries are automatically admitted for any other entries identified by the redundant source configuration. in the example that follows, the two forwarding entries, (10.11.11.11) and (10.11.11.12,) match the flow map defined for **flowMap1**. If an outbound interface is admitted for entry (10.11.11.11), it is also automatically admitted for entry (10.11.11.12) so one source or the other can send traffic at any time.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set policy-options prefix-list permanentEntries1 232.1.1.0/24
set policy-options policy-statement policyForFlow1 from source-address-filter 11.11.11.11/32
  exact
set policy-options policy-statement policyForFlow1 from prefix-list-filter
  permanentEntries1 orlonger
set policy-options policy-statement policyForFlow1 then accept
set routing-options multicast flow-map flowMap1 policy policyForFlow1
set routing-options multicast flow-map flowMap1 bandwidth 2m
set routing-options multicast flow-map flowMap1 bandwidth adaptive
set routing-options multicast flow-map flowMap1 redundant-sources 10.11.11.11
set routing-options multicast flow-map flowMap1 redundant-sources 10.11.11.12
set routing-options multicast flow-map flowMap1 forwarding-cache timeout never
  non-discard-entry-only
```

Step-by-Step Procedure

Multicast flow maps enable you to manage a subset of multicast forwarding table entries. For example, you can specify that certain forwarding cache entries be permanent or have a different timeout value from other multicast flows that are not associated with the flow map policy.

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure a flow map:

1. Configure the flow map policy. This step creates a flow map policy called policyForFlow1. The policy statement matches the source address using the source-address-filter statement, and matches the group address using the prefix-list-filter. The addresses must match the configured policy for flow mapping to occur.

   ```
   [edit policy-options]
   user@host# set prefix-list permanentEntries1 232.1.1.0/24
   user@host# set policy policyForFlow1 from source-address-filter 11.11.11.11/32 exact
   user@host# set policy policyForFlow1 from prefix-list-filter permanentEntries1
     orlonger
   user@host# set policy policyForFlow1 then accept
   ```

2. Define a flow map, **flowMap1**, that references the flow map policy, **policyForFlow1**, we just created.

   ```
   [edit routing-options]
   user@host# set multicast flow-map flowMap1 policy policyForFlow1
   ```

3. Configure permanent forwarding entries (that is, entries that never time out), and enable entries in the pruned state to time out.

```
[edit routing-options]
user@host# set multicast flow-map flowMap1 forwarding-cache timeout never
  non-discard-entry-only
```

4. Configure the flow map bandwidth to be adaptive with a default starting bandwidth of 2 Mbps.

```
[edit routing-options]
user@host# set multicast flow-map flowMap1 bandwidth 2m adaptive
```

5. Specify backup sources.

```
[edit routing-options]
user@host# set multicast flow-map flowMap1 redundant-sources [ 10.11.11.11 10.11.11.12
  ]
```

6. Commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show policy-options** and **show routing-options** commands.

```
user@host# show policy-options
prefix-list permanentEntries1 {
  232.1.1.0/24;
}
policy-statement policyForFlow1 {
  from {
    source-address-filter 11.11.11.11/32 exact;
    prefix-list-filter permanentEntries1 orlonger;
  }
  then accept;
}
```

```
user@host# show routing-options
multicast {
  flow-map flowMap1 {
    policy policyForFlow1;
    bandwidth 2m adaptive;
    redundant-sources [ 10.11.11.11 10.11.11.12 ];
    forwarding-cache {
      timeout never non-discard-entry-only;
    }
  }
}
```

Verification

To verify the configuration, run the following commands:

- **show multicast flow-map**
- **show multicast route extensive**

See Also

- Example: Configuring the Multicast Forwarding Cache on page 929
- Bandwidth Management and Source Redundancy on page 908
- Understanding Bandwidth Management for Multicast on page 907
- Understanding the Multicast Forwarding Cache on page 928

Related Documentation

- Examples: Configuring Administrative Scoping on page 899
- Examples: Configuring Bandwidth Management on page 907

Example: Configuring Ingress PE Redundancy

- Understanding Ingress PE Redundancy on page 936
- Example: Configuring Ingress PE Redundancy on page 937

Understanding Ingress PE Redundancy

In many network topologies, point-to-multipoint label-switched paths (LSPs) are used to distribute multicast traffic over a virtual private network (VPN). When traffic engineering is added to the provider edge (PE) routers, a popular deployment option has been to use traffic-engineered point-to-multipoint LSPs at the origin PE. In these network deployments, the PE is a single point of failure. Network operators have previously provided redundancy by broadcasting duplicate streams of multicast traffic from multiple PEs, a practice which at least doubles the bandwidth required for each stream.

Ingress PE redundancy eliminates the bandwidth duplication requirement by configuring one or more ingress PEs as a group. Within a group, one PE is designated as the primary PE and one or more others become backup PEs for the configured traffic stream. The solution depends on a full mesh of point-to-point (P2P) LSPs among the primary and backup PEs. Also, you must configure a full set of point-to-multipoint LSPs at the backup PEs, even though these point-to-multipoint LSPs at the backup PEs are not sending any traffic or using any bandwidth. The P2P LSPs are configured with bidirectional forwarding detection (BFD). When BFD detects a failure on the primary PE, a new designated forwarder is elected for the stream.

See Also

- *MPLS Applications Feature Guide*

Example: Configuring Ingress PE Redundancy

This example shows how to configure one PE as part of a backup PE group to enable ingress PE redundancy for multicast traffic streams.

Requirements

Before you begin:

- Configure the router interfaces.
- Configure a full mesh of P2P LSPs between the PEs in the backup group.

Overview

Ingress PE redundancy provides a backup resource when point-to-multipoint LSPs are configured for multicast distribution. When point-to-multipoint LSPs are used for multicast traffic, the PE device can become a single point of failure. One way to provide redundancy is by broadcasting duplicate streams from multiple PEs, thus doubling the bandwidth requirements for each stream. This feature implements redundancy between two or more PEs by designating a primary and one or more backup PEs for each configured stream. The solution depends on the configuration of a full mesh of P2P LSPs between the primary and backup PEs. These LSPs are configured with Bidirectional Forwarding Detection (BFD) running on top of them. BFD is used on the backup PEs to detect failure on the primary PE routing device and to elect a new designated forwarder for the stream.

A full mesh is required so that each member of the group can make an independent decision about the health of the other PEs and determine the designated forwarder for the group. The key concept in a backup PE group is that of a designated PE. A designated PE is a PE that forwards data on the static route. All other PEs in the backup PE group do not forward any data on the static route. This allows you to have one designated forwarder. If the designated forwarder fails, another PE takes over as the designated forwarder, thus allowing the traffic flow to continue uninterrupted.

Each PE in the backup PE group makes its own local decision regarding the designated forwarder. Thus, there is no inter-PE communication regarding designated forwarder. A PE computes the designated forwarder based on the IP address of all PEs and the connectivity status of other PEs. Connectivity status is determined based on the state of the BFD session on the P2P LSP to a PE.

A PE chosen is as the designated forwarder if it satisfies the following conditions:

- The PE is in the UP state. Either it is the local PE, or the BFD session on the P2P LSP to that PE is in the UP state.
- The PE has the lowest IP address among all PEs that are in the UP state.

Because all PEs have P2P LSPs to each other, each PE can determine the UP state of each other PE, and all PEs converge to the same designated forwarder.

If the designated forwarder PE fails, then all other PEs lose connectivity with the designated forwarder, and their BFD session ends. Consequently, other PEs then choose another designated forwarder. The new forwarder starts forwarding traffic. Thus, the traffic loss is limited to the failure detection time, which is the BFD session detection time.

When a PE that was the designated forwarder fails and then resumes operating, all other PEs recognize this fact, rerun the designated forwarder algorithm, and choose the PE as the designated forwarder. Consequently, the backup designated forwarder stops forwarding traffic. Thus, traffic switches back to the most eligible designated forwarder.

This example includes the following statements:

- **associate-backup-pe-groups**—Monitors the health of the routing device at the other end of the LSP. You can configure multiple backup PE groups that contain the same routing device's address. Failure of this LSP indicates to all of these groups that the destination PE routing device is down. So, the **associate-backup-pe-groups** statement is not tied to any specific group but applies to all groups that are monitoring the health of the LSP to the remote address.

  If there are multiple LSPs with the **associate-backup-pe-groups** statement to the same destination PE, then the local routing device picks the first LSP to that PE for detection purposes.

  We do not recommend configuring multiple LSPs to the same destination. If you do, make sure that the LSP parameters (for example, liveliness detection) are similar to avoid false failure notification even when the remote PE is up.

- **backup-pe-group**—Configures ingress PE redundancy for multicast traffic streams.
- **bfd-liveness-detection**—Enables BFD for each LSP.
- **label-switched-path**—Configures an LSP. You must configure a full mesh of P2P LSPs between the primary and backup PEs.

  NOTE: We recommend that you configure the P2P LSPs with fast reroute and node link protection so that link failures do not result in the LSP failure. For the purpose of PE redundancy, a failure in the P2P LSP is treated as a PE failure. Redundancy in the inter-PE path is also encouraged.

- **p2mp-lsp-next-hop**—Enables you to associate a backup PE group with a static route.
- **static**—Applies the backup group to a static route on the PE. This ensures that the static route is active (installed in the forwarding table) when the local PE is the designated forwarder for the configured backup PE group.

Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, copy and paste the commands into the CLI at the **[edit]** hierarchy level, and then enter **commit** from configuration mode.

```
set policy-options policy-statement no-rpf from route-filter 225.1.1.1/32 exact
set policy-options policy-statement no-rpf then reject
set protocols mpls label-switched-path backup_PE1 to 10.255.16.61
set protocols mpls label-switched-path backup_PE1 oam bfd-liveness-detection
  minimum-interval 500
set protocols mpls label-switched-path backup_PE1 oam bfd-liveness-detection multiplier
  3
set protocols mpls label-switched-path backup_PE1 associate-backup-pe-groups
set protocols mpls label-switched-path dest1 to 10.255.16.57
set protocols mpls label-switched-path dest1 p2mp p2mp-lsp
set protocols mpls label-switched-path dest2 to 10.255.16.55
set protocols mpls label-switched-path dest2 p2mp p2mp-lsp
```

```
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set routing-options static route 1.1.1.1/32 p2mp-lsp-next-hop p2mp-lsp
set routing-options static route 1.1.1.1/32 backup-pe-group g1
set routing-options static route 225.1.1.1/32 p2mp-lsp-next-hop p2mp-lsp
set routing-options static route 225.1.1.1/32 backup-pe-group g1
set routing-options multicast rpf-check-policy no-rpf
set routing-options multicast interface fe-1/3/3.0 enable
set routing-options multicast backup-pe-group g1 backups 10.255.16.61
set routing-options multicast backup-pe-group g1 local-address 10.255.16.59
```

Step-by-Step Procedure

The following example requires you to navigate various levels in the configuration hierarchy. For information about navigating the CLI, see *Using the CLI Editor in Configuration Mode* in the *CLI User Guide*.

To configure ingress PE redundancy:

1. Configure the multicast settings.

   ```
   [edit routing-options multicast]
   user@host# set rpf-check-policy no-rpf
   user@host# set interface fe-1/3/3.0 enable
   ```

2. Configure the RPF policy.

   ```
   [edit policy-options policy-statement no-rpf]
   user@host# set from route-filter 225.1.1.1/32 exact
   user@host# set then reject
   ```

3. Configure the backup PE group.

   ```
   [edit routing-options multicast]
   user@host# set backup-pe-group g1 backups 10.255.16.61
   user@host# set backup-pe-group g1 local-address 10.255.16.59
   ```

4. Configure the static routes for the point-to-multipoint LSPs backup PE group.

   ```
   [edit routing-options static]
   user@host# set route 1.1.1.1/32 p2mp-lsp-next-hop p2mp-lsp
   user@host# set route 1.1.1.1/32 backup-pe-group g1
   user@host# set route 225.1.1.1/32 p2mp-lsp-next-hop p2mp-lsp
   user@host# set route 225.1.1.1/32 backup-pe-group g1
   ```

5. Configure the MPLS interfaces.

   ```
   [edit protocols mpls]
   user@host# set interface all
   user@host# set interface fxp0.0 disable
   ```

6. Configure the LSP to the redundant router.

   ```
   [edit protocols mpls]
   user@host# set label-switched-path backup_PE1 to 10.255.16.61
   ```

```
user@host# set label-switched-path backup_PE1 oam bfd-liveness-detection
  minimum-interval 500
user@host# set label-switched-path backup_PE1 oam bfd-liveness-detection
  multiplier 3
user@host# set label-switched-path backup_PE1 associate-backup-pe-groups
```

7. Configure LSPs to two traffic destinations.

```
[edit protocols mpls]
user@host# set label-switched-path dest1 to 10.255.16.57
user@host# set label-switched-path dest1 p2mp p2mp-lsp
user@host# set label-switched-path dest2 to 10.255.16.55
user@host# set label-switched-path dest2 p2mp p2mp-lsp
```

8. If you are done configuring the device, commit the configuration.

```
user@host# commit
```

Results

Confirm your configuration by entering the **show policy**, **show protocols**, and **show routing-options** commands.

```
user@host# show policy
policy-statement no-rpf {
  from {
    route-filter 225.1.1.1/32 exact;
  }
  then reject;
}
```

```
user@host# show protocols
mpls {
  label-switched-path backup_PE1 {
    to 10.255.16.61;
    oam {
      bfd-liveness-detection {
        minimum-interval 500;
        multiplier 3;
      }
    }
    associate-backup-pe-groups;
  }
  label-switched-path dest1 {
    to 10.255.16.57;
    p2mp p2mp-lsp;
  }
  label-switched-path dest2 {
    to 10.255.16.55;
    p2mp p2mp-lsp;
  }
  interface all;
  interface fxp0.0 {
    disable;
```

```
    }
}

user@host# show routing-options
static {
    route 1.1.1.1/32 {
        p2mp-lsp-next-hop p2mp-lsp;
        backup-pe-group g1;
    }
    route 225.1.1.1/32 {
        p2mp-lsp-next-hop p2mp-lsp;
        backup-pe-group g1;
    }
}
multicast {
    rpf-check-policy no-rpf;
    interface fe-1/3/3.0 enable;
    backup-pe-group g1 {
        backups 10.255.16.61;
        local-address 10.255.16.59;
    }
}
```

Verification

To verify the configuration, run the following commands:

- **show mpls lsp**
- **show multicast backup-pe-groups**
- **show multicast rpf**

See Also

- Example: Configuring RPF Policies on page 813

Related Documentation

- Examples: Configuring Administrative Scoping on page 899
- Examples: Configuring Bandwidth Management on page 907
- Examples: Configuring the Multicast Forwarding Cache on page 928

PART 7

Configuration Statements and Operational Commands

CHAPTER 27

Configuration Statements

accept-remote-source

Syntax accept-remote-source;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 13.2R2 for PTX Series routers but is not supported for services requiring tunnel-services.

Description Configure an incoming interface to accept multicast traffic from a remote source. A remote source is a source that is not on the same subnet as the incoming interface.

For example, say R1 and R2 are connected on one subnet, and R2 and R3 are connected on another subnet, and that you want R3 to receive multicast traffic from a source connected to R1.

{R1 — [R2} — R3]

In this example, R2 is a pass-through device that is not running PIM, so R3 is the first hop router for multicast packets sent from R1. Because R1 and R3 are in different subnets, the default behavior of R3 is to disregard R1 as a remote source. You can have R3 accept packets from R1, however, by enabling **accept-remote-source** on the target interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Interface to Accept Traffic from a Remote Source on page 403
- Example: Allowing MBGP MVPN Remote Sources on page 633

accounting (Protocols MLD)

Syntax accounting;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced in Junos OS Release 9.1.

Description Enable the collection of MLD join and leave event statistics on the system.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Recording MLD Join and Leave Events on page 70

accounting (Protocols MLD Interface)

Syntax (accounting | no-accounting);

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.1.

Description Enable or disable the collection of MLD join and leave event statistics for an interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Recording MLD Join and Leave Events on page 70

accounting (Protocols IGMP Interface)

Syntax (accounting | no-accounting);

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Enable or disable the collection of IGMP join and leave event statistics for an interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Recording IGMP Join and Leave Events on page 44

accounting (Protocols IGMP AMT Interface)

Syntax (accounting | no-accounting);

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description Enable or disable the collection of IGMP join and leave event statistics for an Automatic Multicast Tunneling (AMT) interface.

Default Disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Default IGMP Parameters for AMT Interfaces on page 425

accounting (Protocols IGMP)

Syntax accounting;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Enable the collection of IGMP join and leave event statistics on the system.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Recording IGMP Join and Leave Events on page 44

accounting (Protocols AMT Interface)

Syntax accounting;

Hierarchy Level [edit logical-systems *logical-system-name* protocols amt relay],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay],
[edit protocols amt relay],
[edit routing-instances *routing-instance-name* protocols amt relay]

Release Information Statement introduced in Junos OS Release 10.2.

Description Enable the collection of statistics for an Automatic Multicast Tunneling (AMT) interface.

Default Disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring the AMT Protocol on page 423

active-source-limit

Syntax

```
active-source-limit {
  log-interval seconds;
  log-warning value;
  maximum number;
  threshold number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* protocols msdp source *ip-address/prefix-length*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp source *ip-address/prefix-length*],
[edit protocols msdp],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit protocols msdp source *ip-address/prefix-length*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp source *ip-address/prefix-length*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Limit the number of active source messages the routing device accepts.

Default

If you do not include this statement, the router accepts any number of MSDP active source messages.

Options

The options are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

address (Local RPs)

Syntax address *address*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp local family (inet | inet6)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)],
[edit protocols pim rp local family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the local rendezvous point (RP) address.

Options ***address***—Local RP address.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Local PIM RPs on page 237

address (Anycast RPs)

Syntax address *address* <forward-msdp-sa>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp local (inet | inet6) anycast-pim rp-set],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local (inet | inet6) anycast-pim rp-set],
[edit protocols pim rp local (inet | inet6) anycast-pim rp-set],
[edit routing-instances *routing-instance-name* protocols pim rp local (inet | inet6) anycast-pim rp-set]

Release Information Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the anycast rendezvous point (RP) addresses in the RP set. Multiple addresses can be configured in an RP set. If the RP has peer Multicast Source Discovery Protocol (MSDP) connections, then the RP must forward MSDP source active (SA) messages.

Options ***address***—RP address in an RP set.

forward-msdp-sa—(Optional) Forward MSDP SAs to this address.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

address (Bidirectional Rendezvous Points)

Syntax

```
address address {
  group-ranges {
    destination-ip-prefix</prefix-length>;
  }
  hold-time seconds;
  priority number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp bidirectional],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp bidirectional],
[edit protocols pim rp bidirectional],
[edit routing-instances *routing-instance-name* protocols pim rp bidirectional]

Release Information

Statement introduced in Junos OS Release 12.1.

Description

Configure bidirectional rendezvous point (RP) addresses. The address can be a loopback interface address, an address of a link interface, or an address that is not assigned to an interface but belongs to a subnet that is reachable by the bidirectional PIM routers in the network.

Options

address—Bidirectional RP address.

Default: 232.0.0.0/8

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

address (Static RPs)

Syntax

```
address address {
  group-ranges {
    destination-ip-prefix</prefix-length>;
  }
  override;
  version version;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp static],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp static],
[edit protocols pim static],
[edit routing-instances *routing-instance-name* protocols pim rp static]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure static rendezvous point (RP) addresses. You can configure a static RP in a logical system only if the logical system is not directly connected to a source.

For each static RP address, you can optionally specify the PIM version and the groups for which this address can be the RP. The default PIM version is version 1.

Options

address—Static RP address.

Default: 224.0.0.0/4

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Static PIM RP Address on the Non-RP Routing Device on page 242

advertise-from-main-vpn-tables

Syntax advertise-from-main-vpn-tables;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp],
[edit protocols bgp],
[edit routing-instances *routing-instance-name* protocols bgp],

Release Information Statement introduced in Junos OS Release 12.3.

Description Advertise VPN routes from the main VPN tables in the master routing instance (for example, bgp.l3vpn.0, bgp.mvpn.0) instead of advertising VPN routes from the tables in the VPN routing instances (for example, *instance-name*.inet.0, *instance-name*.mvpn.0). Enable nonstop active routing (NSR) support for BGP multicast VPN (MVPN).

When this statement is enabled, before advertising a route for a VPN prefix, the path selection algorithm is run on all routes (local and received) that have the same route distinguisher (RD).

NOTE: Adding or removing this statement causes all BGP sessions that have VPN address families to be removed and then added again. On the other hand, having this statement in the configuration prevents BGP sessions from going down when route reflector (RR) or autonomous system border router (ASBR) functionality is enabled or disabled on a routing device that has VPN address families configured.

Default If you do not include this statement, VPN routes are advertised from the tables in the VPN routing instances.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Understanding Junos OS Routing Tables*
- *Types of VPNs*

algorithm

Syntax algorithm *algorithm-name*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection authentication]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the algorithm to use for BFD authentication.

Options ***algorithm-name***—Name of algorithm to use for BFD authentication:

- **simple-password**—Plain-text password. One to 16 bytes of plain text. One or more passwords can be configured.
- **keyed-md5**—Keyed Message Digest 5 hash algorithm for sessions with transmit and receive rates greater than 100 ms.
- **meticulous-keyed-md5**—Meticulous keyed Message Digest 5 hash algorithm.
- **keyed-sha-1**—Keyed Secure Hash Algorithm I for sessions with transmit and receive rates greater than 100 ms.
- **meticulous-keyed-sha-1**—Meticulous keyed Secure Hash Algorithm I.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 357
- Configuring BFD Authentication for PIM on page 196
- **authentication on page 974**

allow-maximum (Multicast)

Syntax allow-maximum;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit logical-systems *logical-system-name* routing-options multicast forwarding-cache],
[edit routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit routing-options multicast forwarding-cache]

Release Information Statement introduced in Junos OS Release 13.2.

Description Allow the larger of global and family-level threshold values to take effect.

This statement is optional when you configure a forwarding cache or PIM state limits. When this statement is included in the configuration and both a family-specific and a global configuration are present, the higher limits take precedence.

For example:

```
[edit routing-options multicast forwarding-cache]
allow-maximum;
family inet {
   threshold {
      suppress 100;
      reuse 75;
   }
}
family inet6 {
   threshold {
      suppress 600;
      reuse 500;
   }
}
threshold {
   suppress 400;
   reuse 450;
}
```

```
user@host# show multicast forwarding-cache statistics

Instance: master Family: INET
Suppress Threshold                    400
Reuse Value                           400
Currently Used Entries                  0

Instance: master Family: INET6
Suppress Threshold                    600
Reuse Value                           500
Currently Used Entries                  0
```

This statement can be useful in single-stack devices on which IPv4 traffic is expected or IPv6 traffic is expected, but not both.

Default By default, this statement is disabled.

When this statement is omitted from the configuration, a family-specific forwarding cache configuration and a global forwarding cache configuration cannot be configured together. Either the global-specific configuration or the family-specific configuration is allowed, but not both.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring the Multicast Forwarding Cache on page 929
- Example: Configuring PIM State Limits on page 754

amt (IGMP)

Syntax

```
amt {
  relay {
    defaults {
      (accounting | no-accounting);
      group-policy [ policy-names ];
      query-interval seconds;
      query-response-interval seconds;
      robust-count number;
      ssm-map ssm-map-name;
      version version;
    }
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp],
[edit protocols igmp],
[edit routing-instances *routing-instance-name* protocols igmp]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure Automatic Multicast Tunneling (AMT) relay attributes.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Default IGMP Parameters for AMT Interfaces on page 425

amt (Protocols)

Syntax

```
amt {
  relay {
    accounting;
    family {
      inet {
        anycast-prefix ip-prefix</prefix-length>;
        local-address ip-address;
      }
    }
    secret-key-timeout minutes;
    tunnel-limit number;
  }
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Enable Automatic Multicast Tunneling (AMT) on the router or switch. You must also configure the local address and anycast prefix for AMT to function.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

anycast-pim

Syntax

```
anycast-pim {
  rp-set {
    address address <forward-msdp-sa>;
  }
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp local family (inet | inet6)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)],
[edit protocols pim rp local family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)]

Release Information

Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure properties for anycast RP using PIM.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM Anycast With or Without MSDP on page 248

anycast-prefix

Syntax anycast-prefix *ip-prefix/<prefix-length>*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols amt relay family inet],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay family inet],
[edit protocols amt relay family inet],
[edit routing-instances *routing-instance-name* protocols amt relay family inet]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify an IP address prefix to use for the Automatic Multicast Tunneling (AMT) relay anycast address. The prefix is advertised by unicast routing protocols to route AMT discovery messages to the router from nearby AMT gateways. The IP address that the prefix is derived from can be configured on any interface in the system. Typically, the router's **lo0.0** loopback address prefix is used for configuring the AMT anycast prefix in the default routing instance, and the router's **lo0.*n*** loopback address prefix is used for configuring the AMT anycast prefix in VPN routing instances. However, the anycast address can be either the primary or secondary **lo0.0** loopback address.

Default None. The anycast prefix must be configured.

Options ***ip-prefix/<prefix-length>***—IP address prefix.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring the AMT Protocol on page 423

asm-override-ssm

Syntax asm-override-ssm;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Enable the routing device to accept any-source multicast join messages (*,G) for group addresses that are within the default or configured range of source-specific multicast groups.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316

assert-timeout

Syntax assert-timeout *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Multicast routing devices running PIM sparse mode often forward the same stream of multicast packets onto the same LAN through the rendezvous-point tree (RPT) and shortest-path tree (SPT). PIM assert messages help routing devices determine which routing device forwards the traffic and prunes the RPT for this group. By default, routing devices enter an assert cycle every 180 seconds. You can configure this assert timeout to be between 5 and 210 seconds.

Options ***seconds***—Time for routing device to wait before another assert message cycle.

Range: 5 through 210 seconds

Default: 180 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring the PIM Assert Timeout on page 291

authentication (Protocols PIM)

Syntax

```
authentication {
  algorithm algorithm-name;
  key-chain key-chain-name;
  loose-check;
}
```

Hierarchy Level

[edit protocols pim interface *interface-name* family (inet | inet6) bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface family (inet | inet6) *interface-name* bfd-liveness-detection]

Release Information

Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the algorithm, security keychain, and level of authentication for BFD sessions running on PIM interfaces.

The remaining statements are explained separately. See CLI Explorer.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD Authentication for PIM on page 196
- Configuring BFD for PIM on page 194
- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 357
- bfd-liveness-detection on page 984
- key-chain (Protocols PIM) on page 1119
- loose-check on page 1137

authentication-key

Syntax authentication-key *peer-key*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Associate a Message Digest 5 (MD5) signature option authentication key with an MSDP peering session.

Default If you do not include this statement, the routing device accepts any valid MSDP messages from the peer address.

Options
peer-key—MD5 authentication key. The peer key can be a text string up to 16 letters and digits long. Strings can include any ASCII characters with the exception of (,), &, and [. If you include spaces in an MSDP authentication key, enclose all characters in quotation marks (“ ”).

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396

auto-rp

Syntax

```
auto-rp {
  (announce | discovery | mapping);
  (mapping-agent-election | no-mapping-agent-election);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced in Junos OS Release 7.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure automatic RP announcement and discovery.

Options

announce—Configure the routing device to listen only for mapping packets and also to advertise itself if it is an RP.

discovery—Configure the routing device to listen only for mapping packets.

mapping—Configures the routing device to announce, listen for and generate mapping packets, and announce that the routing device is eligible to be an RP.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Auto-RP on page 258

autodiscovery

Syntax

```
autodiscovery {
  inet-mdt;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mvpn family *inet*],
[edit routing-instances *routing-instance-name* protocols pim mvpn family *inet*]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement moved to **[..protocols pim mvpn family inet]** from **[.. protocols pim mvpn]** in Junos OS Release 13.3.

Description

For draft-rosen 7, enable the PE routers in the VPN to discover one another automatically.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

autodiscovery-only

Syntax

```
autodiscovery-only {
  intra-as {
    inclusive;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn family *inet* | *inet6*],
[edit routing-instances *routing-instance-name* protocols mvpn family *inet* | *inet6*]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement moved to **[..protocols pim mvpn family inet]** from **[.. protocols mvpn]** in Junos OS Release 13.3.
Support for IPv6 added in Junos OS Release 17.3R1.

Description

Enable the Rosen multicast VPN to use the MDT-SAFI autodiscovery NLRI.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

backoff-period

Syntax backoff-period *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim interface *interface-name* bidirectional df-election],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional df-election],
[edit protocols pim interface *interface-name* bidirectional df-election],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional df-election]

Release Information Statement introduced in Junos OS Release 12.1.

Description Configure the designated forwarder (DF) election backoff period for bidirectional PIM. The **backoff-period** statement configures the period that the acting DF waits between receiving a better DF Offer and sending the Pass message to transfer DF responsibility.

NOTE: Junos OS checks rendezvous point (RP) unicast reachability before accepting incoming DF messages. DF messages for unreachable rendezvous points are ignored. This is needed to prevent the following example scenario. Routers A and B are downstream routers on the same LAN, and both are supposed to send DF election messages with an infinite metric on their upstream interfaces (reverse-path forwarding [RPF] interfaces). Router A has a higher IP address than Router B. When both routers lose the path to the RP, both send an Offer message with the infinite metric onto the LAN. Router A wins the election because it has a higher IP address, and Router B backs off as a result. After three Offer messages, according to RFC 5015, Router A looks up the RP and finds no path to the RP. As a result, Router A transitions to the Lose state and sends nothing. On the other hand, after backing off for an interval of 3 x the Offer period, Router B does not receive any messages, and resumes the DF election by sending a new Offer message. Hence, the pattern repeats indefinitely.

Options *milliseconds*—Period that the acting DF waits between receiving a better DF Offer and sending the Pass message to transfer DF responsibility.

Range: 100 through 65,535 milliseconds

Default: 1000

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

backup-pe-group

Syntax

```
backup-pe-group group-name {
  backups [ addresses ];
  local-address address;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure a backup provider edge (PE) group for ingress PE redundancy when point-to-multipoint label-switched paths (LSPs) are used for multicast distribution.

Options

group-name—Name of the group for PE backups.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Ingress PE Redundancy on page 937

backup (MBGP MVPN)

Syntax backup *address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn static-umh],
[edit routing-instances *routing-instance-name* protocols mvpn static-umh]

Release Information Statement introduced in Junos OS Release 15.1.

Description Define a backup upstream multicast hop (UMH) for type 7 (S,G) routes.

If the primary UMH is unavailable, the backup is used. If neither UMH is available, no UMH is selected.

Options ***address***—Address of the backup UMH.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- sender-based-rpf on page 1278
- static-umh (MBGP MVPN) on page 1315
- unicast-umh-election on page 1364

backups

Syntax backups [*addresses*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast backup-pe-group *group-name*],
[edit logical-systems *logical-system-name* routing-options multicast backup-pe-group *group-name*],
[edit routing-instances *routing-instance-name* routing-options multicast backup-pe-group *group-name*],
[edit routing-options multicast backup-pe-group *group-name*]

Release Information Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the address of backup PEs for ingress PE redundancy when point-to-multipoint label-switched paths (LSPs) are used for multicast distribution.

Options ***addresses***—Addresses of other PEs in the backup group.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Ingress PE Redundancy on page 937

bandwidth

Syntax bandwidth (*bps* | adaptive);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast flow-map],
[edit logical-systems *logical-system-name* routing-options multicast flow-map],
[edit routing-instances *routing-instance-name* routing-options multicast flow-map],
[edit routing-options multicast flow-map]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the bandwidth property for multicast flow maps.

Options **adaptive**—Specify that the bandwidth is measured for the flows that are matched by the flow map.

bps—Bandwidth, in bits per second, for the flow map.

Range: 0 through any amount of bandwidth

Default: 2 Mbps

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring a Multicast Flow Map on page 931

bfd-liveness-detection (Protocols PIM)

Syntax

```
bfd-liveness-detection {
  authentication {
    algorithm algorithm-name;
    key-chain key-chain-name;
    loose-check;
  }
  detection-time {
    threshold milliseconds;
  }
  minimum-interval milliseconds;
  minimum-receive-interval milliseconds;
  multiplier number;
  no-adaptation;
  transmit-interval {
    minimum-interval milliseconds;
    threshold milliseconds;
  }
  version (0 | 1 | automatic);
}
```

Hierarchy Level

[edit protocols pim interface *interface-name* family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* family (inet | inet6)]

Release Information

Statement introduced in Junos OS Release 8.1.
authentication option introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure bidirectional forwarding detection (BFD) timers and authentication for PIM.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- Configuring BFD Authentication for PIM on page 196

bidirectional (Interface)

Syntax

```
bidirectional {
  df-election {
    backoff-period milliseconds;
    offer-period milliseconds;
    robustness-count number;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 12.1.

Description

Configure parameters for bidirectional PIM.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

bidirectional (RP)

Syntax

```
bidirectional {
  address address {
    group-ranges {
      destination-ip-prefix</prefix-length>;
    }
    hold-time seconds;
    priority number;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced in Junos OS Release 12.1.

Description

Configure the routing device's rendezvous-point (RP) properties for bidirectional PIM.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

bootstrap

Syntax

```
bootstrap {
  family (inet | inet6) {
    export [ policy-names ];
    import [ policy-names ];
    priority number;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure parameters to control bootstrap routers and messages.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255

bootstrap-export

Syntax bootstrap-export [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply one or more export policies to control outgoing PIM bootstrap messages.

Options ***policy-names***—Name of one or more import policies.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255
- **bootstrap-import on page 989**

bootstrap-import

Syntax bootstrap-import [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply one or more import policies to control incoming PIM bootstrap messages.

Options ***policy-names***—Name of one or more import policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255
- **bootstrap-export on page 988**

bootstrap-priority

Syntax bootstrap-priority *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure whether this routing device is eligible to be a bootstrap router. In the case of a tie, the routing device with the highest IP address is elected to be the bootstrap router.

Options *number*—Priority for becoming the bootstrap router. A value of 0 means that the routing device is not eligible to be the bootstrap router.

Range: 0 through 255

Default: 0

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253

cmcast-joins-limit-inet (MVPN Selective Tunnels)

Syntax cmcast-joins-limit-inet *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective],
[edit routing-instances *instance-name* provider-tunnel selective]

Release Information Statement introduced in Junos OS Release 13.3.

Description Configure the maximum number of IPv4 customer multicast entries

The purpose of the **cmcast-joins-limit-inet** statement is to supplement the multicast forwarding-cache limit when the MVPN **rpt-spt** mode is configured and when traffic is flowing through selective service provider multicast service inteface (S-PMSI) tunnels and is forwarded by way of the (*,G) entry, even though the forwarding cache limit has already blocked the forwarding entries from being created.

The **cmcast-joins-limit-inet** statement limits the number of Type-6 and Type-7 routes. These routes contain customer-route control information.

You can configure the **cmcast-joins-limit-inet** statement only when the MVPN mode is **rpt-spt**.

This statement is independent of the leaf-tunnel-limit-inet statement and of the **forwarding-cache** threshold statement.

The **cmcast-joins-limit-inet** statement is applicable on the egress PE router. It limits the customer multiccast entries created in response to PIM (*,G) and (S,G) join messages. This statement is applicable to both type-6 and type-7 routes because the intention is to limit the egress forwarding entries, and in **rpt-spt** mode, an MVPN creates forwarding entries for both of these route types (in other words, for both (*,G) and (S,G) entries). However, this statement does not block BGP-created customer multicast entries because the purpose of this statement is to prevent the creation of forwarding entries on the egress PE router only and only for non-remote receivers. If remote-side customer multicast entries or forwarding entries need to be limited, you can use forwarding-cache threshold on the ingress routers, in which case this statement is not required.

By placing a limit on the customer multicast entries, you can ensure that when the limit is reached or the maximum forwarding state is created, all further local join messages will be blocked by the egress PE router. This ensures that traffic is flowing for only those multicast entries that are permitted.

If another PE router is interested in the traffic, it might pull the traffic from the ingress PE router by sending type-6 and type-7 routes. To prevent forwarding in this case, you can configure the leaf tunnel limit (**leaf-tunnel-limit-inet**). By preventing type-4 routes from being sent in response to type-3 routes, the formation of selective tunnels is blocked when the tunnel limit is reached. This ensures that traffic flows only for the routes within the tunnel limit . For all other routes, traffic flows only to the PE routers that have not reached the configured limit.

Setting the **cmcast-joins-limit-inet** statement or reducing the value of the limit does not alter or delete the already existing and installed routes. If needed, you can run the **clear pim join** command to force the limit to take effect. Those routes that cannot be processed because of the limit are added to a queue, and this queue is processed when the limit is removed or increased and when existing routes are deleted.

Default Unlimited

Options ***number***—Maximum number of customer multicast entries for IPv4.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring the Multicast Forwarding Cache on page 928
- Example: Configuring MBGP Multicast VPN Topology Variations on page 648

cmcast-joins-limit-inet6 (MVPN Selective Tunnels)

Syntax cmcast-joins-limit-inet6 *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective],
[edit routing-instances *instance-name* provider-tunnel selective]

Release Information Statement introduced in Junos OS Release 13.3.

Description Configure the maximum number of IPv4 customer multicast entries

The purpose of the **cmcast-joins-limit-inet6** statement is to supplement the multicast forwarding-cache limit when the MVPN **rpt-spt** mode is configured and when traffic is flowing through selective service provider multicast service inteface (S-PMSI) tunnels and is forwarded by way of the (*,G) entry, even though the forwarding cache limit has already blocked the forwarding entries from being created.

The **cmcast-joins-limit-inet6** statement limits the number of Type-6 and Type-7 routes. These routes contain customer-route control information.

You can configure the **cmcast-joins-limit-inet6** statement only when the MVPN mode is **rpt-spt**.

This statement is independent of the leaf-tunnel-limit-inet6 statement and of the **forwarding-cache** threshold statement.

The **cmcast-joins-limit-inet6** statement is applicable on the egress PE router. It limits the customer multiccast entries created in response to PIM (*,G) and (S,G) join messages. This statement is applicable to both type-6 and type-7 routes because the intention is to limit the egress forwarding entries, and in **rpt-spt** mode, an MVPN creates forwarding entries for both of these route types (in other words, for both (*,G) and (S,G) entries). However, this statement does not block BGP-created customer multicast entries because the purpose of this statement is to prevent the creation of forwarding entries on the egress PE router only and only for non-remote receivers. If remote-side customer multicast entries or forwarding entries need to be limited, you can use forwarding-cache threshold on the ingress routers, in which case this statement is not required.

By placing a limit on the customer multicast entries, you can ensure that when the limit is reached or the maximum forwarding state is created, all further local join messages will be blocked by the egress PE router. This ensures that traffic is flowing for only those multicast entries that are permitted.

If another PE router is interested in the traffic, it might pull the traffic from the ingress PE router by sending type-6 and type-7 routes. To prevent forwarding in this case, you can configure the leaf tunnel limit (**leaf-tunnel-limit-inet6**). By preventing type-4 routes from being sent in response to type-3 routes, the formation of selective tunnels is blocked when the tunnel limit is reached. This ensures that traffic flows only for the routes within the tunnel limit . For all other routes, traffic flows only to the PE routers that have not reached the configured limit.

Setting the **cmcast-joins-limit-inet6** statement or reducing the value of the limit does not alter or delete the already existing and installed routes. If needed, you can run the **clear pim join** command to force the limit to take effect. Those routes that cannot be processed because of the limit are added to a queue, and this queue is processed when the limit is removed or increased and when existing routes are deleted.

Default Unlimited

Options ***number***—Maximum number of customer multicast entries for IPv4.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Examples: Configuring the Multicast Forwarding Cache on page 928
- Example: Configuring MBGP Multicast VPN Topology Variations on page 648

create-new-ucast-tunnel

Syntax create-new-ucast-tunnel;

Hierarchy Level [edit routing-instances *routing-instance-name* provider-tunnel ingress-replication],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* ingress-replication]

Release Information Statement introduced in Junos OS Release 10.4.

Description One of two modes for building unicast tunnels when ingress replication is configured for the provider tunnel. When this statement is configured, each time a new destination is added to the multicast distribution tree, a new unicast tunnel to the destination is created in the ingress replication tunnel. The new tunnel is deleted if the destination is no longer needed. Use this mode for RSVP LSPs using ingress replication.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs on page 591
- mpls-internet-multicast on page 1165
- ingress-replication on page 1096

dampen

Syntax dampen *minutes*

Hierarchy Level [edit logical-systems *logical-system--name* protocols mvpn mvpn-mode spt-only source-active-advertisement],
[edit logical-systems *logical-system--name* routing-instances *instance-name* protocols mvpn mvpn-mode spt-only source-active-advertisement],
[edit routing-instances protocols mvpn mvpn-mode spt-only source-active-advertisement],
[edit routing-instances *instance-name* protocols mvpn mvpn-mode spt-only source-active-advertisement]

Release Information Statement introduced in Junos OS Release 17.1.

Description Time to wait before re-advertising the source-active route (1 to 30 minutes). After traffic on the ingress PE falls below the threshold set for min-rate, this is length of time that resuming traffic must continue to exceed the **min-rate** before the ingress PE can start re-advertising Source-Active A-D routes.

The default is 1 minute.

To verify that the value is set as expected, you can check whether the Type 5 (Source-Active route) has been advertised using the **show route table *vrf*.mvpn.0** command. It may take several minutes before you can see the changes in the Source-Active A-D route advertisement after making changes to the **min-rate**.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*

data-encapsulation

Syntax data-encapsulation (disable | enable);

Hierarchy Level [edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure a rendezvous point (RP) using MSDP to encapsulate multicast data received in MSDP register messages inside forwarded MSDP source-active messages.

Default If you do not include this statement, the RP encapsulates multicast data.

Options **disable**—(Optional) Do not use MSDP data encapsulation.

enable—Use MSDP data encapsulation.

Default: **enable**

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

data-forwarding

Syntax

```
data-forwarding {
  receiver {
    source-vlans vlan-list;
    install;
  }
  source {
    groups group-prefix;
  }
}
```

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*)]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for the QFX Series.

Description Configure the VLAN to be a multicast source VLAN (MVLAN) or a multicast VLAN registration (MVR) receiver VLAN. Each data-forwarding VLAN, which can be a multicast source VLAN (MVLAN) or a multicast receiver VLAN, must have exactly one source statement or exactly one receiver statement. A data-forwarding VLAN can operate only in IGMP version 2 (IGMPv2) mode.

The remaining statements are explained separately. See CLI Explorer.

Default Disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

data-mdt-reuse

Syntax data-mdt-reuse;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim mdt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt]

Release Information Statement introduced in Junos OS Release 10.0. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description Enable dynamic reuse of data MDT group addresses.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Enabling Dynamic Reuse of Data MDT Group Addresses on page 517

default-peer

Syntax default-peer;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Establish this peer as the default MSDP peer and accept source-active messages from the peer without the usual peer-reverse-path-forwarding (peer-RPF) check.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

default-vpn-source

Syntax

```
default-vpn-source {
  interface-name interface-name;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit protocols pim]

Release Information Statement introduced in Junos OS Release 10.1.

Description Enable the router to use the primary loopback address configured in the default routing instance as the source address when PIM hello messages, join messages, and prune messages are sent over multicast tunnel interfaces for interoperability with other vendors' routers.

The remaining statements are explained separately. See CLI Explorer.

Default By default, the router uses the loopback address configured in the VRF routing instance as the source address when sending PIM hello messages, join messages, and prune messages over multicast tunnel interfaces.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- interface-name on page 1114

defaults

Syntax

```
defaults {
  (accounting | no-accounting);
  group-policy [ policy-names ];
  query-interval seconds;
  query-response-interval seconds;
  robust-count number;
  ssm-map ssm-map-name;
  version version;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* statement-name protocols igmp amt relay],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* statement-name protocols igmp amt relay],
[edit protocols igmp amt relay],
[edit routing-instances *routing-instance-name* statement-name protocols igmp amt relay]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Configure default IGMP attributes for all Automatic Multicast Tunneling (AMT) interfaces.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

dense-groups

Syntax

```
dense-groups {
  addresses;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure which groups are operating in dense mode.

Options

addresses—Address of groups operating in dense mode.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Sparse-Dense Mode Properties on page 207

detection-time (BFD for PIM)

Syntax

```
detection-time {
  threshold milliseconds;
}
```

Hierarchy Level

[edit protocols pim interface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information

Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Enable BFD failure detection. The BFD failure detection timers are adaptive and can be adjusted to be faster or slower. The lower the BFD failure detection timer value, the faster the failure detection and vice versa. For example, the timers can adapt to a higher value if the adjacency fails (that is, the timer detects failures more slowly). Or a neighbor can negotiate a higher value for a timer than the configured value. The timers adapt to a higher value when a BFD session flap occurs more than three times in a span of 15 seconds. A back-off algorithm increases the receive (Rx) interval by two if the local BFD instance is the reason for the session flap. The transmission (Tx) interval is increased by two if the remote BFD instance is the reason for the session flap. You can use the **clear bfd adaptation** command to return BFD interval timers to their configured values. The **clear bfd adaptation** command is hitless, meaning that the command does not affect traffic flow on the routing device.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- bfd-liveness-detection on page 984
- threshold on page 1323

df-election

Syntax

```
df-election {
  backoff-period milliseconds;
  offer-period milliseconds;
  robustness-count number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim interface *interface-name* bidirectional],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional],
[edit protocols pim interface *interface-name* bidirectional],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional]

Release Information

Statement introduced in Junos OS Release 12.1.

Description

Optionally, configure the designated forwarder (DF) election parameters for bidirectional PIM.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

disable

Syntax disable;

Hierarchy: disable (Protocols IGMP)
[edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Hierarchy: disable (Protocols SAP)
[edit logical-systems *logical-system-name* protocols sap],
[edit protocols sap]

Hierarchy: disable (Protocols MSDP)
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Hierarchy: disable (Protocols MLD)
[edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

disable (PIM Graceful Restart)
[edit logical-systems *logical-system-name* protocols pim graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim graceful-restart],
[edit protocols pim graceful-restart],
[edit routing-instances *routing-instance-name* protocols pim graceful-restart]

Hierarchy: disable (Protocols DVMRP)
[edit logical-systems *logical-system-name* protocols dvmrp],
[edit logical-systems *logical-system-name* protocols dvmrp interface *interface-name*],
[edit protocols dvmrp],
[edit protocols dvmrp interface *interface-name*]

Hierarchy: disable (PIM)
[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* protocols pim family (inet | inet6)],
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* protocols pim rp local family (inet | inet6)],

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)],
[edit protocols pim],
[edit protocols pim family (inet | inet6)],
[edit protocols pim interface *interface-name*],
[edit protocols pim rp local family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim mvpn family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)]

disable (Multicast Snooping)

[edit multicast-snooping-options graceful-restart]

Hierarchy: disable (Protocols MLD Snooping)

[edit protocols mld-snooping vlan (all | *vlan-name*)]

disable (IGMP Snooping)

[edit protocols igmp-snooping vlan (all | *vlan-name*)]

disable (MLD Snooping)

[edit protocols mld-snooping vlan (all | *vlan-name*)]

Hierarchy: disable (IGMP Snooping)

[edit protocols igmp-snooping vlan *vlan-name*]

Release Information

address (Local RPs) and **disable (Protocols IGMP)** and **disable (Protocols SAP)** and **disable (PIM)** and **disable (Protocols MLD)** and **disable (Protocols MSDP)** introduced before Junos OS Release 7.4.
address (Local RPs) and **disable (Protocols IGMP)**introduced in Junos OS Release 9.0 for EX Series switches.
disable (IGMP Snooping)introduced in Junos OS Release 9.2 for EX Series switches.
disable statement extended to the **[family]** hierarchy level of disable (PIM) in Junos OS Release 9.6.
disable (IGMP Snooping) introduced in Junos OS Release 11.1 for the QFX Series.
disable (MLD Snooping) introduced in Junos OS Release 18.1R1 for the SRX1500 devices.
address (Local RPs) introduced in Junos OS Release 11.3 for the QFX Series.
disable (Protocols IGMP) and **disable (Protocols MLD Snooping)** and **disable (Protocols MSDP)**introduced in Junos OS Release 12.1 for the QFX Series.
disable (Protocols MLD Snooping)introduced in Junos OS Release 12.1 for EX Series switches.
disable (Multicast Snooping) introduced in Junos OS Release 12.3.
address (Local RPs) and **disable (Protocols MSDP)** introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Description

disable (Protocols IGMP)disables IGMP on the system.

disable (Protocols SAP)explicitly disables SAP.

disable (Protocols MSDP) explicitly disables MSDP.

disable (Protocols MLD)disables MLD on the system.

disable (PIM Graceful Restart)explicitly disables PIM sparse mode graceful restart.

disable (Protocols DVMRP)explicitly disables DVMRP on the system or on an interface.

disable (PIM)explicitly disable PIM at the protocol, interface or family hierarchy levels.

disable (Multicast Snooping)explicitly disables graceful restart for multicast snooping.

disable (Protocols MLD Snooping)disables MLD snooping on the VLAN. Multicast traffic will be flooded to all interfaces in the VLAN except the source interface.

disable (IGMP Snooping)disables IGMP snooping on the VLAN. Multicast traffic will be flooded to all interfaces on the VLAN except the source interface.

disable (IGMP Snooping)disables IGMP snooping on all interfaces in a VLAN.

Default If you do not include this statement, MLD snooping is enabled on all interfaces in the VLAN.

If you do not include this statement in the configuration for a VLAN, IGMP snooping is enabled on the VLAN.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **mld-snooping on page 1155**
- Disabling IGMP on page 49
- Disabling MLD on page 74
- Disabling PIM on page 297
- **family (Protocols PIM) on page 1031**
- Configuring the Session Announcement Protocol on page 415
- Configuring PIM Sparse Mode Graceful Restart on page 383
- Example: Configuring Multicast Snooping on page 874
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- **show mld-snooping vlans on page 1525**
- *show igmp-snooping vlans*

disable (IGMP Snooping)

Syntax disable;

Hierarchy Level [edit protocols igmp-snooping vlan *vlan-name*]

Release Information Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Disable IGMP snooping on the VLAN. Multicast traffic will be flooded to all interfaces on the VLAN except the source interface.

Default If you do not include this statement in the configuration for a VLAN, IGMP snooping is enabled on the VLAN.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on Switches on page 93
- Configuring IGMP Snooping on Switches on page 88
- *Configuring IGMP Snooping (CLI Procedure)*
- *show igmp-snooping vlans*

disable (Protocols MLD Snooping)

Syntax disable;

Hierarchy Level [edit protocols mld-snooping vlan (all | *vlan-name*)]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.

Description Disable MLD snooping on the VLAN. Multicast traffic will be flooded to all interfaces in the VLAN except the source interface.

Default If you do not include this statement, MLD snooping is enabled on all interfaces in the VLAN.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- show mld-snooping vlans on page 1525

disable (Multicast Snooping)

Syntax disable;

Hierarchy Level [edit multicast-snooping-options graceful-restart]

Release Information Statement introduced in Junos OS Release 12.3.

Description Explicitly disable graceful restart for multicast snooping.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast Snooping on page 874

disable (PIM)

Syntax disable;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* protocols pim family (inet | inet6)],
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* protocols pim rp local family (inet | inet6)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)],
[edit protocols pim],
[edit protocols pim family (inet | inet6)],
[edit protocols pim interface *interface-name*],
[edit protocols pim rp local family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim mvpn family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)]

Release Information
Statement introduced before Junos OS Release 7.4.
disable statement extended to the **[family]** hierarchy level in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Explicitly disable PIM at the protocol, interface or family hierarchy levels.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Disabling PIM on page 297
- family (Protocols PIM) on page 1031

disable (Protocols MLD)

Syntax disable;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*], [edit protocols mld interface *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Disable MLD on the system.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Disabling MLD on page 74

disable (Protocols MSDP)

Syntax disable;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Explicitly disable MSDP.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Disabling MSDP on page 412

disable (Protocols SAP)

Syntax disable;

Hierarchy Level [edit logical-systems *logical-system-name* protocols sap],
[edit protocols sap]

Release Information Statement introduced before Junos OS Release 7.4.

Description Explicitly disable SAP.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring the Session Announcement Protocol on page 415

distributed-dr

Syntax distributed-dr;

Hierarchy Level [edit dynamic-profiles *name* protocols pim interface *interface-name*],
[edit logical-systems *name* protocols pim interface *interface-name*],
[edit logical-systems *name* routing-instances *name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *name* protocols pim interface *interface-name*]

Release Information Statement introduced in Junos OS Release 17.2R1.

Description Enable PIM distributed designated-router (DR) functionality on IRB interfaces associated with EVPN virtual LANs (VLANs) that have been configured with IGMP snooping. By effectively disabling certain PIM features that are not required in this scenario, this statement supports using PIM to perform intersubnet, that is, inter-VLAN, multicast routing more efficiently.

When you configure this statement, PIM ignores the DR status of the interface when processing IGMP reports received on the interface. When the interface receives the IGMP report, the provider edge (PE) device sends PIM upstream join messages to pull the multicast stream and forward it to the interface—regardless of the DR status of the interface. The statement also disables the PIM assert mechanism on the interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Example: Preserving Bandwidth with IGMP Snooping in an EVPN-VXLAN Environment*

distributed (IGMP)

Syntax distributed;

Hierarchy Level [edit protocols igmp interface *interface-name*],
[edit dynamic-profiles protocols igmp interface $junos-interface-name]

Release Information Statement introduced in Junos OS Release 14.1X50.
Support added in Junos OS Release 18.2R1 for using **distributed** IGMP in conjunction with Multipoint LDP (mLDP) in-band signalling.

Description Enable distributed IGMP by moving IGMP processing from the Routing Engine to the Packet Forwarding Engine. Distributed IGMP reduces the join and leave latency of IGMP memberships.

Distributed IGMP is only available when **chassis network-services enhanced-ip** is configured.

NOTE: When you enable distributed IGMP, the following interface options are not supported on the Packet Forwarding Engine: **oif-map**, **group-limit**, **ssm-map**, and **static**. However, the **ssm-map-policy** option is supported on distributed IGMP interfaces. The **traceoptions** and **accounting** statements can only be enabled for IGMP operations still performed on the Routing Engine; they are not supported on the Packet Forwarding Engine. The **clear igmp membership** command is not supported when distributed IGMP is enabled.

When the **distributed** command is enabled in conjunction with *mldp-inband-signalling*, (so PIM act as a multipoint LDP inband edge router), it supports interconnecting separate PIM domains via a MPLS-based core.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling Distributed IGMP on page 76
- *Configuring Dynamic DHCP Client Access to a Multicast Network*
- For general information about configuring IGMP, see the *Multicast Protocols Feature Guide*

dr-election-on-p2p

Syntax dr-election-on-p2p;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced in Junos OS Release 9.1.
Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Enable PIM designated router (DR) election on point-to-point (P2P) links.

Default No PIM DR election is performed on point-to-point links.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Designated Router Election on Point-to-Point Links on page 303

dr-register-policy

Syntax dr-register-policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply one or more policies to control outgoing PIM register messages.

Options ***policy-names***—Name of one or more import policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Register Message Filters on a PIM RP and DR on page 279
- rp-register-policy on page 1264

dvmrp

Syntax

```
dvmrp {
  disable;
  export [ policy-names ];
  import [ policy-names ];
  interface interface-name {
    disable;
    hold-time seconds;
    metric metric;
    mode (forwarding | unicast-routing);
  }
  rib-group group-name;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit protocols]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Enable DVMRP on the router or switch.

Default

DVMRP is disabled on the router or switch.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring DVMRP on page 434

embedded-rp

Syntax

```
embedded-rp {
  group-ranges {
    destination-ip-prefix</prefix-length>;
  }
  maximum-rps limit;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure properties for embedded IP version 6 (IPv6) RPs.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Embedded RP for IPv6 on page 266

exclude (Protocols IGMP)

Syntax exclude;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static group *multicast-group-address*],
[edit protocols igmp interface *interface-name* static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 9.3.

Description Configure the static group to operate in exclude mode. In exclude mode all sources except the address configured are accepted for the group. If this statement is not included, the group operates in include mode.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Enabling IGMP Static Group Membership on page 37

exclude (Protocols MLD)

Syntax exclude;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name* static group *multicast-group-address*],
[edit protocols mld interface *interface-name* static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 9.3.

Description Configure the static group to operate in exclude mode. In exclude mode all sources except the address configured are accepted for the group. By default, the group operates in include mode.

Required Privilege Level view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Enabling MLD Static Group Membership on page 63

export (Protocols PIM)

Syntax

export [*policy-names*];

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Apply one or more export policies to control outgoing PIM join and prune messages. PIM join and prune filters can be applied to PIM-SM and PIM-SSM messages. PIM join and prune filters cannot be applied to PIM-DM messages.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Filtering Outgoing PIM Join Messages on page 271

export (Protocols DVMRP)

Syntax export [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description Apply one or more policies to routes being exported from the routing table into DVMRP. If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching policy is applied to the route. If no match is found, the routing table exports into DVMRP only the routes that it learned from DVMRP and direct routes.

Options ***policy-names***—Name of one or more policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- import on page 1089
- Example: Configuring DVMRP to Announce Unicast Routes on page 438

export (Protocols MSDP)

Syntax export [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply one or more policies to routes being exported from the routing table into MSDP.

Options ***policy-names***—Name of one or more policies.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396
- import on page 1090

export (Bootstrap)

Syntax export [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp bootstrap family (inet | inet6)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp bootstrap family (inet | inet6)],
[edit protocols pim rp bootstrap family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap family (inet | inet6)]

Release Information Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply one or more export policies to control outgoing PIM bootstrap messages.

Options ***policy-names***—Name of one or more import policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255
- import (Protocols PIM Bootstrap) on page 1092

export-target

Syntax

```
export-target {
  target target-community;
  unicast;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target]

Release Information

Statement introduced in Junos OS Release 8.4.

Description

Enable you to override the Layer 3 VPN import and export route targets used for importing and exporting routes for the MBGP MVPN network layer reachability information (NLRI).

Options

target *target-community*—Specify the export target community.

unicast—Use the same target community as specified for unicast.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

family (Local RP)

Syntax

```
family (inet | inet6) {
  disable;
  address address;
  anycast-pim {
    local-address address;
    rp-set {
      address address <forward-msdp-sa>;
    }
  }
  group-ranges {
    destination-ip-prefix</prefix-length>;
  }
  hold-time seconds;
  override;
  priority number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp local],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local],
[edit protocols pim rp local],
[edit routing-instances *routing-instance-name* protocols pim rp local]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure which IP protocol type local RP properties to apply.

Options

inet—Apply IP version 4 (IPv4) local RP properties.

inet6—Apply IPv6 local RP properties.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Local PIM RPs on page 237

family (Bootstrap)

Syntax

```
family (inet | inet6) {
  export [ policy-names ];
  import [ policy-names ];
  priority number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp bootstrap],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp bootstrap],
[edit protocols pim rp bootstrap],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap]

Release Information

Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure which IP protocol type bootstrap properties to apply.

Options

inet—Apply IP version 4 (IPv4) local RP properties.

inet6—Apply IPv6 local RP properties.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255

family (Protocols AMT Relay)

Syntax

```
family {
  inet {
    anycast-prefix ip-prefix/<prefix-length>;
    local-address ip-address;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols amt relay],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay],
[edit protocols amt relay],
[edit routing-instances *routing-instance-name* protocols amt relay]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Configure the protocol address family for Automatic Multicast Tunneling (AMT) relay functions. Only the **inet** family for IPv4 protocol addresses is supported.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

family (Protocols PIM Interface)

Syntax

```
family (inet | inet6) {
  bfd-liveness-detection {
    authentication {
      algorithm algorithm-name;
      key-chain key-chain-name;
      loose-check;
    }
    detection-time {
      threshold milliseconds;
    }
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier number;
    no-adaptation;
    transmit-interval {
      minimum-interval milliseconds;
      threshold milliseconds;
    }
    version (0 | 1 | automatic);
  }
  disable;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 9.6.
Support for the Bidirectional Forwarding Detection (BFD) Protocol statements was introduced in Junos OS Release 12.2.

Description

Configure one of the following PIM protocol settings for the specified family on the specified interface:

- BFD protocol settings
- Disable PIM

Options

inet—Enable the PIM protocol for the IP version 4 (IPv4) address family.

inet6—Enable the PIM protocol for the IP version 6 (IPv6) address family.

The remaining statements are explained separately. See CLI Explorer.

Related Documentation

- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 357
- Disabling PIM on page 297

family (VRF Advertisement)

Syntax

```
family {
  inet-mvpn;
  inet6-mvpn;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* vrf-advertise-selective],
[edit routing-instances *routing-instance-name* vrf-advertise-selective],

Release Information

Statement introduced in Junos OS Release 10.1.

Description

Explicitly enable IPv4 or IPv6 MVPN routes to be advertised from the VRF instance while preventing all other route types from being advertised.

The options are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring PIM-SSM GRE Selective Provider Tunnels*
- inet-mvpn on page 1100
- inet6-mvpn on page 1102

family (Protocols PIM)

Syntax

```
family (inet | inet6) {
    disable;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Disable the PIM protocol for the specified family.

Options

inet—Disable the PIM protocol for the IP version 4 (IPv4) address family.

inet6—Disable the PIM protocol for the IP version 6 (IPv6) address family.

Related Documentation

- Disabling PIM on page 297
- *disable (PIM Graceful Restart)*
- disable (PIM) on page 1011

flood-groups

Syntax flood-groups [*ip-addresses*];

Hierarchy Level [edit bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* multicast-snooping-options]

Release Information Statement introduced in Junos OS Release 8.5.

Description Establish a list of flood group addresses for multicast snooping.

Options ***ip-addresses***—List of IP addresses subject to flooding.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast Snooping on page 874

flow-map

Syntax

```
flow-map flow-map-name {
  bandwidth (bps | adaptive);
  forwarding-cache {
    timeout (never non-discard-entry-only | minutes);
  }
  policy [ policy-names ];
  redundant-sources [ addresses ];
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure multicast flow maps.

Options

flow-map-name—Name of the flow-map.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring a Multicast Flow Map on page 931

forwarding-cache (Flow Maps)

Syntax

```
forwarding-cache {
  timeout (minutes | never non-discard-entry-only );
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit logical-systems *logical-system-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-options multicast flow-map *flow-map-name*]

Release Information

Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure multicast forwarding cache properties for the flow map.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring a Multicast Flow Map on page 931

forwarding-cache (Bridge Domains)

Syntax

```
forwarding-cache {
    threshold suppress value <reuse value>;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* multicast-snooping-options]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Establish multicast snooping forwarding cache parameter values.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast Snooping on page 874

graceful-restart (Protocols PIM)

Syntax

```
graceful-restart {
  disable;
  no-bidirectional-mode;
  restart-duration seconds;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure PIM sparse mode graceful restart.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Sparse Mode Graceful Restart on page 383

graceful-restart (Multicast Snooping)

Syntax

```
graceful-restart {
  disable;
  restart-duration seconds;
}
```

Hierarchy Level [edit multicast-snooping-options]

Release Information Statement introduced in Junos OS Release 9.2.

Description Establish the graceful restart duration for multicast snooping. You can set this value between 0 and 300 seconds. If you set the duration to 0, graceful restart is effectively disabled. Set this value slightly larger than the IGMP query response interval.

Default 180 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast Snooping on page 874
- query-response-interval (Bridge Domains) on page 1236

group (Bridge Domains)

Syntax

```
group ip-address {
    source-address ip-address;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name* static],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name* static],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name* static],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping interface *interface-name* static]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Configure the IGMP multicast group address that receives data on an interface and (optionally) a source address for certain packets.

Options

ip-address—Group address.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

group (Distributed IGMP)

Syntax

```
group multicast-group-address {
  <distributed>;
  source source-address <distributed>;
}
```

Hierarchy Level [edit protocols pim static]

Release Information Statement introduced in Junos OS Release 14.1X50.

Description Specify the multicast group address for the multicast group that is statically configured on an interface.

Options

distributed—(Optional) Preprovision a specific multicast group address (G).

multicast-group-address—Specific multicast group address being statically configured on an interface.

The remaining statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling Distributed IGMP on page 76
- For general information about configuring IGMP, see the *Multicast Protocols Feature Guide*
- For information about enabling IGMP, see "Enabling IGMP" in the *Multicast Protocols Feature Guide*

group (IGMP Snooping)

Syntax group *ip-address*;

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) interface (all | *interface-name*) static]

Release Information Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure a static multicast group on an interface.

Options ***ip-address***—IP address of the multicast group receiving data on an interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring IGMP Snooping (CLI Procedure)*
- *show igmp-snooping membership*
- show igmp-snooping vlans on page 1494

group (Protocols PIM)

Syntax

```
group group-address {
    source source-address {
        rate threshold-rate;
    }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mdt threshold],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt threshold],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdtthreshold]

Release Information

Statement introduced before Junos OS Release 7.4. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description

Specify the explicit or prefix multicast group address to which the threshold limits apply. This is typically a well-known address for a certain type of multicast traffic.

Options

group-address—Explicit group address to limit.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

group (Protocols MSDP)

Syntax

```
group group-name {
  disable;
  export [ policy-names ];
  import [ policy-names ];
  local-address address;
  mode (mesh-group | standard);
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
  peer address; {
    disable;
    active-source-limit {
      maximum number;
      threshold number;
    }
    authentication-key peer-key;
    default-peer;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
      file filename <files number> <size size> <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Define an MSDP peer group. MSDP peers within groups share common tracing options, if present and not overridden for an individual peer with the peer statement. To configure multiple MSDP groups, include multiple **group** statements.

By default, the group's options are identical to the global MSDP options. To override the global options, include group-specific options within the **group** statement.

The group must contain at least one peer.

Options

group-name—Name of the MSDP group.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396

group (Protocols MLD)

Syntax

```
group multicast-group-address {
  exclude;
  group-count number;
  group-increment increment;
  source ip-address {
    source-count number;
    source-increment increment;
  }
}
```

Hierarchy Level
[edit logical-systems *logical-system-name* protocols mld interface *interface-name* static],
[edit protocols mld interface *interface-name* static]

Release Information
Statement introduced before Junos OS Release 7.4.

Description
The MLD multicast group address and (optionally) the source address for the multicast group being statically configured on an interface.

Options
multicast-group-address—Address of the group.

NOTE: You must specify a unique address for each group.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Enabling MLD Static Group Membership on page 63

group (Protocols IGMP)

Syntax

```
group multicast-group-address {
  exclude;
  group-count number;
  group-increment increment;
  source ip-address {
    source-count number;
    source-increment increment;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static], [edit protocols igmp interface *interface-name* static]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify the IGMP multicast group address and (optionally) the source address for the multicast group being statically configured on an interface.

NOTE: You must specify a unique address for each group.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP Static Group Membership on page 37

group (Protocols MLD Snooping)

Syntax

```
group multicast-group-address {
    source ip-address;
}
```

Hierarchy Level

[edit protocols mld-snooping vlan (all | *vlan-name*) interface (all | *interface-name*) static]
[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name* interface *interface-name* static]

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.
Support at the **[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name* interface *interface-name* static]** hierarhcy level introduced in Junos OS Release 13.3 for EX Series switches.
Support for the **source** statement introduced in Junos OS Release 13.3 for EX Series switches.

Description

Configure a static multicast group on an interface and (optionally) the source address for the multicast group.

Options

multicast-group-address—Valid IP multicast address for the multicast group.

source *ip-address*—Valid IP multicast address for the source of the multicast group.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

group (Routing Instances)

Syntax

```
group address {
  source source-address {
    inter-region-segmented {
      fan-out fan-out value;
      threshold rate-value;
    }
    ldp-p2mp;
      pim-ssm {
      group-range multicast-prefix;
    }
    rsvp-te {
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
      static-lsp lsp-name;
    }
    threshold-rate number;
  }
  wildcard-source {
    inter-region-segmented {
      fan-out fan-out value;
    }
    ldp-p2mp;
    pim-ssm {
      group-range multicast-prefix;
    }
    rsvp-te {
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
      static-lsp lsp-name;
    }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

Release Information

Statement introduced in Junos OS Release 8.5.
The **inter-region-segmented** statement added in Junos OS Release 15.1.

Description

Specify the IP address for the multicast group configured for point-to-multipoint label-switched paths (LSPs) and PIM-SSM GRE selective provider tunnels.

Options

address—Specify the IP address for the multicast group. This address must be a valid multicast group address.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- *Configuring PIM-SSM GRE Selective Provider Tunnels*

group (RPF Selection)

Syntax

```
group group-address{
  sourcesource-address{
    next-hop next-hop-address;
  }
  wildcard-source {
    next-hop next-hop-address;
  }
}
```

Hierarchy Level
[edit routing-instances *routing-instance-name* edit protocols pim rpf-selection]

Release Information
Statement introduced in JUNOS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Configure the PIM group address for which you configure RPF selectiongroup (RPF Selection).

Default
By default, PIM RPF selection is not configured.

Options
group-address—PIM group address for which you configure RPF selection.

Required Privilege Level
view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Example: Configuring PIM RPF Selection on page 816

group-address (Routing Instances Tunnel Group)

Syntax group-address *address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* pim-ssm]

Release Information Statement introduced in Junos OS Release 9.4.
In Junos OS Release 17.3R1, the **pim-ssm** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default multicast distribution tree (MDT) in Rosen 7, and data MDT for Rosen 6 and Rosen 7.

Description Configure the PIM-ASM (Rosen 6) or PIM-SSM (Rosen 7) provider tunnel group address. Each MDT is linked to a group address in the provider space.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

group-address (Routing Instances VPN)

Syntax group-address *address*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-asm],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-asm family inet],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-asm family inet6],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-ssm],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-ssm family inet],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-ssm family inet6],
[edit routing-instances *routing-instance-name* provider-tunnel pim-asm],
[edit routing-instances *routing-instance-name* provider-tunnel pim-asm family inet],
[edit routing-instances *routing-instance-name* provider-tunnel pim-asm family inet6],
[edit routing-instances *routing-instance-name* provider-tunnel pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel pim-ssm family inet],
[edit routing-instances *routing-instance-name* provider-tunnel pim-ssm family inet6]

Release Information Statement introduced before Junos OS Release 7.4.
Starting with Junos OS Release 11.4, to provide consistency with draft-rosen 7 and next-generation BGP-based multicast VPNs, configure the provider tunnels for draft-rosen 6 anysource multicast VPNs at the **[edit routing-instances *routing-instance-name* provider-tunnel]** hierarchy level. The **mdt**, **vpn-tunnel-source**, and **vpn-group-address** statements are deprecated at the **[edit routing-instances *routing-instance-name* protocols pim]** hierarchy level. Use **group-address** in place of **vpn-group-address**.

Description Specify a group address on which to encapsulate multicast traffic from a virtual private network (VPN) instance.

NOTE: IPv6 provider tunnels are not currently supported for draft-rosen MVPNs. They are supported for MBGP MVPNs.

Options ***address***—For IPv4, IP address whose high-order bits are 1110, giving an address range from 224.0.0.0 through 239.255.255.255, or simply 224.0.0.0/4. For IPv6, IP address whose high-order bits are FF00 (FF00::/8).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Any-Source Multicast for Draft-Rosen VPNs on page 449

- *Configuring Multicast Layer 3 VPNs*
- *Multicast Protocols Feature Guide*

group-count (Protocols IGMP)

Syntax group-count *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static group *multicast-group-address*],
[edit protocols igmp interface *interface-name* static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the number of static groups to be created.

Options *number*—Number of static groups.

Range: 1 through 512

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Enabling IGMP Static Group Membership on page 37

group-count (Protocols MLD)

Syntax group-count *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name* static group *multicast-group-address*],
[edit protocols mld interface *interface-name* static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 9.6.

Description Configure the number of static groups to be created.

Options *number*—Number of static groups.

Default: 1

Range: 1 through 512

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD Static Group Membership on page 63

group-increment (Protocols IGMP)

Syntax group-increment *increment*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static group *multicast-group-address*],
[edit protocols igmp interface *interface-name* static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the number of times the address should be incremented for each static group created. The increment is specified in dotted decimal notation similar to an IPv4 address.

Options *increment*—Number of times the address should be incremented.

Default: 0.0.0.1

Range: 0.0.0.1 through 255.255.255.255

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP Static Group Membership on page 37

group-increment (Protocols MLD)

Syntax group-increment *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name* static group *multicast-group-address*],
[edit protocols mld interface *interface-name* static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 9.6.

Description Configure the number of times the address should be incremented for each static group created. The increment is specified in a format similar to an IPv6 address.

Options *increment*—Number of times the address should be incremented.

Default: ::1

Range: ::1 through ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD Static Group Membership on page 63

group-limit (IGMP)

Syntax group-limit *limit*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 10.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure a limit for the number of multicast groups (or [S,G] channels in IGMPv3) allowed on an interface. After this limit is reached, new reports are ignored and all related flows are not flooded on the interface.

To confirm the configured group limit on the interface, use the show igmp interface command.

Default By default, there is no limit to the number of multicast groups that can join the interface.

Options *limit*—group limit value for the interface.

Range: 1 through 32767

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Limiting the Number of IGMP Multicast Group Joins on Logical Interfaces on page 45
- group-threshold on page 1063
- log-interval on page 1132

group-limit (IGMP and MLD Snooping)

Syntax group-limit *limit*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure a limit for the number of multicast groups (or [S,G] channels in IGMPv3) allowed on an interface. After this limit is reached, new reports are ignored and all related flows are not flooded on the interface.

Default By default, there is no limit to the number of multicast groups joining an interface.

Options *limit*—a 32-bit number for the limit on the interface.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring IGMP Snooping on page 106

group-limit (Protocols MLD)

Syntax group-limit *limit*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 10.4.

Description Configure a limit for the number of multicast groups (or [S,G] channels in MLDv2) allowed on a logical interface. After this limit is reached, new reports are ignored and all related flows are not flooded on the interface.

Default By default, there is no limit to the number of multicast groups that can join the interface.

Options *limit*—group value limit for the interface.

Range: 1 through 32767

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring MLD on page 50

group-policy (Protocols IGMP)

Syntax group-policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.1.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description When this statement is enabled on a router running IGMP version 2 (IGMPv2) or version 3 (IGMPv3), after the router receives an IGMP report, the router compares the group against the specified group policy and performs the action configured in that policy (for example, rejects the report).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Filtering Unwanted IGMP Reports at the IGMP Interface Level on page 32

group-policy (Protocols IGMP AMT Interface)

Syntax group-policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description When this statement is enabled on the Automatic Multicast Tunneling (AMT) interfaces running IGMP version 2 (IGMPv2) or version 3 (IGMPv3), after the router receives an IGMP report, the router compares the group against the specified group policy and performs the action configured in that policy (for example, rejects the report).

Options ***policy-names***—Name of the policy.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Default IGMP Parameters for AMT Interfaces on page 425

group-policy (Protocols MLD)

Syntax group-policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.1.

Description When a routing device running MLD version 1 or version 2 (MLDv1 or MLDv2), receives an MLD report, the routing device compares the group against the specified group policy and performs the action configured in that policy (for example, rejects the report).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Filtering Unwanted MLD Reports at the MLD Interface Level on page 60

group-range (Data MDTs)

Syntax group-range *multicast-prefix*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*mdt],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*mdt]

Release Information Statement introduced before Junos OS Release 7.4. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description Establish the group range to use for data MDTs created in this VRF instance. Only IPv4 address are valid for group range. This address range cannot overlap the default MDT addresses of any other VPNs on the router, nor can the group range specified under the **inet** and **inet6** hierarchies overlap. If you configure overlapping group ranges, the configuration commit fails. Up to 8000 MDT group ranges are supported for IPv4 and IPv6.

Options ***multicast-prefix***—Multicast address range to identify data MDTs.

Range: Any valid, nonreserved multicast address range

Default: None (No data MDTs are created for this VRF instance.)

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

group-range (MBGP MVPN Tunnel)

Syntax group-range *multicast-prefix*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *group-address* source *source-address* pim-ssm],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *group-address* wildcard-source pim-ssm],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source pim-ssm],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *group-address* source *source-address* pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *group-address* wildcard-source pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source pim-ssm]

Release Information Statement introduced in Junos OS Release 10.1.

Description Establish the multicast group address range to use for creating MBGP MVPN source-specific multicast selective PMSI tunnels.

Options ***multicast-prefix***—Multicast group address range to be used to create MBGP MVPN source-specific multicast selective PMSI tunnels.

Range: Any valid, nonreserved IPv4 multicast address range

Default: None

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

group-ranges

Syntax

```
group-ranges {
  destination-ip-prefix</prefix-length>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp bidirectional address *address*],
[edit logical-systems *logical-system-name* protocols pim rp embedded-rp],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp bidirectional address *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp embedded-rp],
[edit protocols pim rp bidirectional address *address*],
[edit protocols pim rp embedded-rp],
[edit protocols pim rp local family (inet | inet6)],
[edit protocols pim rp static address *address*],
[edit routing-instances *instance-name* protocols pim rp bidirectional address *address*],
[edit routing-instances *routing-instance-name* protocols pim rp embedded-rp],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp static address *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Support for bidirectional RP addresses introduced in Junos OS Release 12.1.
Statement introduced in Junos OS Release 13.3 for the PTX5000 router.

Description

Configure the address ranges of the multicast groups for which this routing device can be a rendezvous point (RP).

Default

The routing device is eligible to be the RP for all IPv4 or IPv6 groups (224.0.0.0/4 or FF70::/12 to FFF0::/12).

Options

destination-ip-prefix</prefix-length>—Addresses or address ranges for which this routing device can be an RP.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Local PIM RPs on page 237
- Configuring PIM Embedded RP for IPv6 on page 266
- Example: Configuring Bidirectional PIM on page 343

group-rp-mapping

Syntax

```
group-rp-mapping {
  family (inet | inet6) {
    log-interval seconds;
    maximum limit;
    threshold value;
  }
  log-interval seconds;
  maximum limit;
  threshold value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced in Junos OS Release 12.2.

Description

Configure a limit for the number of incoming group-to-RP mappings.

NOTE: The maximum limit settings that you configure with the **maximum** and the **family (inet | inet6) maximum** statements are mutually exclusive. For example, if you configure a global maximum group-to-RP mapping limit, you cannot configure a limit at the family level for IPv4 or IPv6. If you attempt to configure a limit at both the global level and the family level, the device will not accept the configuration.

Options

family (inet | inet6)—(Optional) Specify either IPv4 or IPv6 messages to be counted towards the configured group-to-RP mapping limit.

Default: Both IPv4 and IPv6 messages are counted towards the configured group-to-RP limit.

The remaining statements are described separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM State Limits on page 754

group-threshold (Protocols IGMP Interface)

Syntax group-threshold *value*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*], [edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.2.

Description Specify the threshold at which a warning message is logged for the multicast groups received on a logical interface. The threshold is a percentage of the maximum number of multicast groups allowed on a logical interface.

For example, if you configure a maximum number of 1,000 incoming multicast groups, and you configure a threshold value of 90 percent, warning messages are logged in the system log when the interface receives 900 groups.

To confirm the configured group threshold on the interface, use the show igmp interface command.

Default By default, there is no configured threshold value.

Options *value*—Percentage of the maximum number of multicast groups allowed on the interface that starts triggering the warning. You configure a percentage of the **group-limit** value that starts triggering the warnings. You must explicitly configure the group-limit to configure a threshold value.

Range: 1 through 100

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Limiting the Number of IGMP Multicast Group Joins on Logical Interfaces on page 45
- group-limit on page 1054
- log-interval on page 1132

group-threshold (Protocols MLD Interface)

Syntax group-threshold *value*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.2.

Description Specify the threshold at which a warning message is logged for the multicast groups received on a logical interface. The threshold is a percentage of the maximum number of multicast groups allowed on a logical interface.

For example, if you configure a maximum number of 1,000 incoming multicast groups, and you configure a threshold value of 90 percent, warning messages are logged in the system log when the interface receives 900 groups.

To confirm the configured group threshold on the interface, use the show mld interface command.

Default By default, there is no configured threshold value.

Options *value*—Percentage of the maximum number of multicast groups allowed on the interface that starts triggering the warning. You configure a percentage of the **group-limit** value that starts triggering the warnings. You must explicitly configure the group-limit to configure a threshold value.

Range: 1 through 100

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Number of MLD Multicast Group Joins on Logical Interfaces on page 72
- group-limit on page 1056
- log-interval on page 1133

groups (Multicast VLAN Registration)

Syntax groups *group-prefix*;

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) data-forwarding source]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.

Description Specify the IP address range of the multicast VLAN (MVLAN) source interfaces.

Default Disabled

Options *group-prefix*—IP address range of the source group. Each MVLAN must have exactly one **groups** statement. If there are multiple MVLANs on the switch, their group ranges must be unique.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

hello-interval

Syntax hello-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Specify how often the routing device sends PIM hello packets out of an interface.

Options ***seconds***—Length of time between PIM hello packets.

Range: 0 through 255

Default: 30 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- hold-time (Protocols PIM) on page 1069
- Modifying the PIM Hello Interval on page 190

hold-time (Protocols DVMRP)

Syntax hold-time *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols dvmrp interface *interface-name*], [edit protocols dvmrp interface *interface-name*]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description Specify the time period for which a neighbor is to consider the sending router (this router) to be operative (up).

Options *seconds*—Hold time.

Range: 1 through 255

Default: 35 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring DVMRP on page 434

hold-time (Protocols MSDP)

Syntax hold-time *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer address],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name* peer *address*]
[edit routing-instances *instance-name* protocols msdp peer *address*],

Release Information Statement introduced in Junos OS Release 12.3.

Description Specify the hold-time period to use when maintaining a connection with the MSDP peer. If a keepalive message is not received for the hold-time period, the MSDP peer connection is terminated. According to the RFC 3618, *Multicast Source Discovery Protocol (MSDP)*, the recommended value for the hold-time period is 75 seconds.

The hold-time period must be longer than the keepalive interval.

You might want to change the hold-time period and keepalive timer for consistency in a multi-vendor environment.

Default In Junos OS, the default hold-time period is 75 seconds, and the default keepalive interval is 60 seconds.

Options ***seconds***—Hold time.
Range: 15 through 150 seconds
Default: 75 seconds

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Examples: Configuring MSDP on page 393
- keep-alive (Protocols MSDP) on page 1118
- sa-hold-time (Protocols MSDP) on page 1271

hold-time (Protocols PIM)

Syntax hold-time *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp bidirectional address *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp bidirectional address *address*],
[edit protocols pim rp bidirectional address *address*],
[edit protocols pim rp local family (inet | inet6)],
[edit routing-instances *instance-name* protocols pim rp bidirectional address *address*],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional RP addresses introduced in Junos OS Release 12.1.

Description Specify the time period for which a neighbor is to consider the sending routing device (this routing device) to be operative (up).

Options *seconds*—Hold time.

Range: 1 through 65535

Default: 150 seconds

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Local PIM RPs on page 237
- Example: Configuring Bidirectional PIM on page 343

host-only-interface

Syntax host-only-interface;

Hierarchy Level [edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure an interface as a host-facing interface. IGMP queries received on these interfaces are dropped.

Default The interface can either be a host-side or multicast-router interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

host-outbound-traffic (Multicast Snooping)

Syntax

```
host-outbound-traffic {
  forwarding-class class-name;
  dot1p number;
}
```

Hierarchy Level

[edit multicast-snooping-options],
[edit bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]

Release Information

Statement introduced in Junos OS Release 12.3.

Description

On an MX Series router in a network enabled for CET service and IGMP snooping, configure multicast forwarding class and IEEE 802.1p value to rewrite of IGMP self generated packets.

Options

- ***class-name***—Name of the forwarding class.
- ***number***—802.1p priority number.

  Range: 0 through 7

  Default: 0

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Multicast Snooping on page 873
- Configuring IGMP Snooping on page 104

hot-root-standby (MBGP MVPN)

Syntax

```
hot-root-standby {
  min-rate <rate>;
  source-tree;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information

Statement introduced in Junos OS Release 16.1.

Description

In a BGP multicast VPN (MVPN) with RSVP-TE point-to-multipoint provider tunnels, configure hot-root standby, as defined in *Multicast VPN fast upstream failover*, draft-morin-l3vpn-mvpn-fast-failover-05.

Hot-root standby enables an egress PE router to select two upstream PE routers for an (S,G) and send C-multicast joins to both the PE routers. Multiple ingress PE routers then receive traffic from the source and forward into the core. The egress PE router uses sender-based RPF to forward the one stream received by the primary upstream PE router.

When **hot-root-standby** is configured, based on local policy, as soon as the PE router receives this standby BGP customer multicast route, the PE can install the VRF PIM state corresponding to this BGP source-tree join route. The result is that join messages are sent to the CE device toward the customer source (C-S0, and the PE router receives (C-S,C-G) traffic. Also, based on local policy, as soon as the PE router receives this standby BGP customer multicast route, the PE router can forward (C-S, C-G) traffic to other PE routers through a P-tunnel independently of the reachability of the C-S through some other PE router.

The receivers must join the source tree (SPT) to establish a hot-root standby. Customer multicast join messages continue to be sent to a single upstream provider edge (PE) router for shared-tree state, and duplicate data does not flow through the core in this case.

Section 4 of Draft Morin specifies that hot-root standby is limited to the case where the site that contains the C-S is connected to exactly two PE routers. In the case that there are more than two PE routers multihomed to the source, the backup PE router is the PE router chosen with the highest IP address (not including the primary upstream PE router). This is a local decision that is not specified in the specification.

There is no limitation in Junos OS on which upstream multicast hop (UMH) selection method is used. For example, you can use static-umh (MBGP MVPN) or unicast-umh-election.

PIM dense mode as the customer multicast protocol is not supported.

Hot-root standby is supported for RSVP point-to-multipoint provider tunnels. Other provider tunnels are not supported. A commit error results if **hot-root-standby** is configured and the provider-tunnel is not RSVP point-to-multipoint.

Fast failover (sub 50ms) is supported for C-multicast streams within NG-MVPNs in a hot-standby mode. The threshold to trigger fast failover must be set. See min-rate for information on fast failover.

Cold-root standby and warm-root standby, as specified in draft Morin, are not supported.

The backup attribute is not sent in the customer multicast routes, as this is only needed for warm and cold-root standby.

Internet multicast is not supported with hot-root standby.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- sender-based-rpf on page 1278
- unicast-umh-election on page 1364

idle-standby-path-switchover-delay

Syntax idle-standby-path-switchover-delay <*seconds*>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced in Junos OS Release 12.2.

Description Configure the time interval after which an ECMP join is moved to the standby path in the absence of traffic on the path.

In the absence of this statement, ECMP joins are not moved to the standby path until traffic is detected on the path.

Options ***<seconds>***—Time interval after which an ECMP join is moved to the standby RPF path in the absence of traffic on the path.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM Make-Before-Break Join Load Balancing on page 790
- Configuring PIM Join Load Balancing on page 218
- clear pim join-distribution on page 1418
- join-load-balance on page 1116
- standby-path-creation-delay on page 1306

igmp

Syntax

```
igmp {
  accounting;
  interface interface-name {
    (accounting | no-accounting);
    disable;
    distributed;
    group-limit limit;
    group-policy [ policy-names ];
    group-threshold
    immediate-leave;
    log-interval
    oif-map map-name;
    passive;
    promiscuous-mode;
    ssm-map ssm-map-name;
    ssm-map-policy ssm-map-policy-name;
    static {
      group multicast-group-address {
        exclude;
        group-count number;
        group-increment increment;
        source ip-address {
          source-count number;
          source-increment increment;
        }
      }
    }
    version version;
  }
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit protocols]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Statement introduced in Junos OS Release 12.3R2 for EX Series switches.

Description

Enable IGMP on the router or switch. IGMP must be enabled for the router or switch to receive multicast packets.

The remaining statements are explained separately. See CLI Explorer.

Default IGMP is disabled on the router or switch. IGMP is automatically enabled on all broadcast interfaces when you configure Protocol Independent Multicast (PIM) or Distance Vector Multicast Routing Protocol (DVMRP).

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP on page 28
- *Understanding Multicast Route Leaking for VRF and Virtual-Router Instances*

igmp-snooping

List of Syntax

Syntax (EX Series, QFX Series, and NFX Series)

```
igmp-snooping {
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable> <match
      regex>;
    flag flag (detail | disable | receive | send);
  }
  vlan (vlan-name | all) {
    data-forwarding {
      source {
        groups group-prefix;
      }
      receiver {
        source-vlans vlan-list;
        install ;
      }
    }
    disable;
    immediate-leave;
    interface interface-name {
      group-limit limit;
      host-only-interface;
      immediate-leave;
      multicast-router-interface;
      static {
        group multicast-ip-address;
      }
    }
    l2-querier {
      source-address ip-address;
      }
    proxy {
      source-address ip-address;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    version number;
  }
}
```

Syntax (MX Series)

```
igmp-snooping {
  immediate-leave;
  interface interface-name {
    group-limit limit;
    host-only-interface;
    immediate-leave;
```

```
    multicast-router-interface;
    static {
      group ip-address {
        source ip-address;
      }
    }
  }
  proxy {
    source-address ip-address;
  }
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
  vlan vlan-id {
    immediate-leave;
    interface interface-name {
      group-limit limit;
      host-only-interface;
      immediate-leave;
      multicast-router-interface;
      static {
        group ip-address {
          source ip-address;
        }
      }
    }
    proxy {
      source-address ip-address;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
  }
}
```

Syntax (SRX Series)

```
igmp-snooping {
  vlan (all | vlan-name) {
    immediate-leave;
    interface interface-name {
      group-limit range;
      host-only-interface;
      multicast-router-interface;
      immediate-leave;
      static {
        group multicast-ip-address {
        source ip-address;
        }
      }
    }
    l2-querier {
      source-address ip-address;
      }
    proxy {
      source-address ip-address;
    }
    qualified-vlan vlan-id;
    query-interval number;
    query-last-member-interval number;
    query-response-interval number;
  robust-count number;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier>;
    }
  }
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols]
[edit routing-instances *routing-instance-name* protocols]
[edit protocols]

Release Information

Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.
Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX Series.

Description

Configure IGMP snooping to constrain multicast traffic to only the ports that have receivers attached. IGMP snooping enables the device to selectively send out multicast packets on only the ports that need them. Without IGMP snooping, the device floods the packets on every port. The device listens for the exchange of IGMP messages by the device and the end hosts. In this way, the device builds an IGMP snooping table that has a list of all the ports that have requested a particular multicast group. The factory default configuration enables IGMP snooping on all VLANs.

NOTE: IGMP snooping must be disabled on the device before enabling ISSU.

Default IGMP snooping is disabled on the device.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *IGMP Snooping in MC-LAG Active-Active Mode*
- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- IGMP Snooping Overview on page 81

ignore-stp-topology-change

Syntax ignore-stp-topology-change;

Hierarchy Level [edit bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options]

Release Information Statement introduced in Junos OS Release 9.5.

Description Ignore messages about spanning tree topology changes. This statement is supported for the **virtual-switch** routing instance type only.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Multicast Snooping on page 874

immediate-leave (Bridge Domains)

Syntax immediate-leave;

Hierarchy Level [edit bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description The immediate leave setting is useful for minimizing the leave latency of IGMP memberships. When this setting is enabled, the routing device leaves the multicast group immediately after the last host leaves the multicast group.

The immediate-leave setting enables host tracking, meaning that the device keeps track of the hosts that send join messages. This allows IGMP to determine when the last host sends a leave message for the multicast group.

When the immediate leave setting is enabled, the device removes an interface from the forwarding-table entry without first sending IGMP group-specific queries to the interface. The interface is pruned from the multicast tree for the multicast group specified in the IGMP leave message. The immediate leave setting ensures optimal bandwidth management for hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When immediate leave is disabled and one host sends a leave group message, the routing device first sends a group query to determine if another receiver responds. If no receiver responds, the routing device removes all hosts on the interface from the multicast group. Immediate leave is disabled by default for both IGMP version 2 and IGMP version 3.

NOTE: Although host tracking is enabled for IGMPv2 and MLDv1 when you enable immediate leave, use immediate leave with these versions only when there is one host on the interface. The reason is that IGMPv2 and MLDv1 use a report suppression mechanism whereby only one host on an interface sends a group join report in response to a membership query. The other interested hosts suppress their reports. The purpose of this mechanism is to avoid a flood of reports for the same group. But it also interferes with host tracking, because the router only knows about the one interested host and does not know about the others.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

immediate-leave (Protocols IGMP)

Syntax immediate-leave;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*], [edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description The immediate leave setting is useful for minimizing the leave latency of IGMP memberships. When this setting is enabled, the routing device leaves the multicast group immediately after the last host leaves the multicast group.

The immediate leave setting enables host tracking, meaning that the device keeps track of the hosts that send join messages. This allows IGMP to determine when the last host sends a leave message for the multicast group.

When the immediate leave setting is enabled, the device removes an interface from the forwarding-table entry without first sending IGMP group-specific queries to the interface. The interface is pruned from the multicast tree for the multicast group specified in the IGMP leave message. The immediate leave setting ensures optimal bandwidth management for hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When immediate leave is disabled and one host sends a leave group message, the routing device first sends a group query to determine if another receiver responds. If no receiver responds, the routing device removes all hosts on the interface from the multicast group. Immediate leave is disabled by default for both IGMP version 2 and IGMP version 3.

NOTE: Although host tracking is enabled for IGMPv2 and MLDv1 when you enable immediate leave, use immediate leave with these versions only when there is one host on the interface. The reason is that IGMPv2 and MLDv1 use a report suppression mechanism whereby only one host on an interface sends a group join report in response to a membership query. The other interested hosts suppress their reports. The purpose of this mechanism is to avoid a flood of reports for the same group. But it also interferes with host tracking, because the router only knows about the one interested host and does not know about the others.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Specifying Immediate-Leave Host Removal for IGMP on page 31

immediate-leave (IGMP Snooping)

Syntax immediate-leave;

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description Configure IGMP snooping immediate leave for the specified VLAN. When you configure immediate leave, host tracking is enabled, which allows the switch to track the hosts that send membership reports. The switch can then determine when the last host on an interface leaves the multicast group and immediately stop forwarding multicast traffic to the interface.

Configuring immediate leave reduces the amount of time it takes for the switch to stop sending multicast traffic to an interface when the last host leaves the group. When immediate leave is disabled, the switch no longer tracks hosts. Instead, whenever it receives a leave report from a host, it sends out a group-specific query to all hosts. If it does not receive any membership reports on the interface in response to the group-specific query within a set interval, it stops forwarding multicast traffic to the interface.

After the device receives a leave group membership message from a host, immediately remove the group membership from the interface and suppress the sending of any group-specific queries for the multicast group.

NOTE: Immediate leave is supported for both IGMP version 2 (IGMPv2) and IGMPv3. However, with IGMPv2, we recommend that you configure immediate leave only when there is only one IGMP host on an interface. In IGMPv2, only one host on a interface sends a membership report in response to a general query—any other interested hosts suppress their reports. Report suppression avoids a flood of reports for the same group, but it also interferes with host tracking because the device knows only about one interested host on the interface at any given time.

Default The immediate-leave feature is disabled.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on Switches on page 93
- Configuring IGMP Snooping on Switches on page 88
- show igmp-snooping vlans on page 1494

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- IGMP Snooping Overview on page 81

immediate-leave (Protocols MLD)

Syntax immediate-leave;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*], [edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.3.

Description The immediate leave setting is useful for minimizing the leave latency of MLD memberships. When this setting is enabled, the routing device leaves the multicast group immediately after the last host leaves the multicast group.

The immediate-leave setting enables host tracking, meaning that the device keeps track of the hosts that send join messages. This allows MLD to determine when the last host sends a leave message for the multicast group.

When the immediate leave setting is enabled, the device removes an interface from the forwarding-table entry without first sending MLD group-specific queries to the interface. The interface is pruned from the multicast tree for the multicast group specified in the MLD leave message. The immediate leave setting ensures optimal bandwidth management for hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When immediate leave is disabled and one host sends a leave group message, the routing device first sends a group query to determine if another receiver responds. If no receiver responds, the routing device removes all hosts on the interface from the multicast group. Immediate leave is disabled by default for both MLD version 1 and MLD version 2.

NOTE: Although host tracking is enabled for IGMPv2 and MLDv1 when you enable immediate leave, use immediate leave with these versions only when there is one host on the interface. The reason is that IGMPv2 and MLDv1 use a report suppression mechanism whereby only one host on an interface sends a group join report in response to a membership query. The other interested hosts suppress their reports. The purpose of this mechanism is to avoid a flood of reports for the same group. But it also interferes with host tracking, because the routing device only knows about the one interested host and does not know about the others.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Specifying Immediate-Leave Host Removal for MLD on page 59

immediate-leave (Protocols MLD Snooping)

Syntax immediate-leave;

Hierarchy Level
[edit protocols mld-snooping vlan (all | *vlan-name*)]
[edit protocols mld-snooping vlan *vlan-name* interface *interface-name*]
[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name* interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 18.1R1 for the SRX1500 devices.
Support at the **[edit protocols mld-snooping vlan *vlan-name* interface *interface-name*]** and the **[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name* interface *interface-name*]** hierarchy levels introduced in Junos OS Release 13.3 for EX Series switches.

Description
Configure MLD snooping immediate leave for the specified VLAN or interface. When you configure immediate leave, host tracking is enabled, which allows the switch to track the hosts that send join messages. The switch can then determine when the last host on an interface leaves the multicast group and immediately stop forwarding multicast traffic to the interface.

Configuring immediate leave reduces the amount of time it takes for the switch to stop sending multicast traffic to an interface when the last host leaves the group. When immediate leave is disabled, the switch no longer tracks hosts. Instead, whenever it receives a leave message from a host, it sends out a group membership query to all hosts. If it does not receive any join group reports on the interface in response to the group membership query within a set interval, it then stops forwarding multicast traffic to the interface.

NOTE: Immediate leave is supported for both MLD version 1 (MLDv1) and MLDv2. However, with MLDv1, we recommend that you configure immediate leave only when there is only one MLD host on an interface. In MLDv1, only one host on a interface sends a join report in response to a group membership query—any other interested hosts suppress their reports. This report-suppression feature means that the switch only knows about one interested host on the interface at any given time.

Default The immediate-leave feature is disabled.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- mld-snooping on page 1155
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Understanding MLD Snooping on page 125

import (Protocols DVMRP)

Syntax

import [*policy-names*];

Hierarchy Level

[edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description

Apply one or more policies to routes being imported into the routing table from DVMRP. If you specify more than one policy, they are evaluated in the order specified, from first to last, and the first matching policy is applied to the route. If no match is found, DVMRP shares with the routing table only those routes that were learned from DVMRP routers.

Options

policy-names—Name of one or more policies.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- export on page 1022
- Example: Configuring DVMRP to Announce Unicast Routes on page 438

import (Protocols MSDP)

Syntax import [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply one or more policies to routes being imported into the routing table from MSDP.

Options ***policy-names***—Name of one or more policies.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396
- export on page 1023

import (Protocols PIM)

Syntax import [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply one or more policies to routes being imported into the routing table from PIM. Use the **import** statement to filter PIM join messages and prevent them from entering the network.

Options ***policy-names***—Name of one or more policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Filtering Incoming PIM Join Messages on page 275

import (Protocols PIM Bootstrap)

Syntax import [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp bootstrap (inet | inet6)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp bootstrap (inet | inet6)],
[edit protocols pim rp bootstrap (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap (inet | inet6)]

Release Information
Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply one or more import policies to control incoming PIM bootstrap messages.

Options ***policy-names***—Name of one or more import policies.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255
- export (Bootstrap) on page 1024

import-target

Syntax

```
import-target {
  target {
    target-value;
    receiver target-value;
    sender target-value;
  }
  unicast {
    receiver;
    sender;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target]

Release Information Statement introduced in Junos OS Release 8.4.

Description Enable you to override the Layer 3 VPN import and export route targets used for importing and exporting routes for the MBGP MVPN NLRI.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

inclusive

Syntax inclusive;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn family *inet* | *inet6* autodiscovery-onlyintra-as],
[edit routing-instances *routing-instance-name* protocols mvpn family *inet* | *inet6* autodiscovery-only intra-as],

Release Information Statement introduced in Junos OS Release 9.4.
Statement moved to **[..protocols mvpn family inet]** from **[.. protocols mvpn]** in Junos OS Release 13.3.
Support for IPv6 added in Junos OS Release 17.3R1.

Description For Rosen 7, enable the MVPN control plane for autodiscovery only, using intra-AS autodiscovery routes over an inclusive provider multicast service interface (PMSI).

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

infinity

Syntax infinity [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim spt-threshold],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim spt-threshold],
[edit protocols pim spt-threshold],
[edit routing-instances *routing-instance-name* protocols pim spt-threshold]

Release Information Statement introduced in Junos OS Release 8.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply one or more policies to set the SPT threshold to infinity for a source-group address pair. Use the **infinity** statement to prevent the last-hop routing device from transitioning from the RPT rooted at the RP to an SPT rooted at the source for that source-group address pair.

Options ***policy-names***—Name of one or more policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring the PIM SPT Threshold Policy on page 293

ingress-replication

Syntax

```
ingress-replication {
  create-new-ucast-tunnel;
  label-switched-path {
    label-switched-path-template {
      (template-name | default-template);
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit protocols mvpn inter-region-template template *template-name* all-regions],
[edit protocols mvpn inter-region-template template *template-name* region *region-name*],
[edit routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]

Release Information

Statement introduced in Junos OS Release 10.4.

Description

A provider tunnel type used for passing multicast traffic between routers through the MPLS cloud, or between PE routers when using MVPN. The ingress replication provider tunnel uses MPLS point-to-point LSPs to create the multicast distribution tree.

Optionally, you can specify a label-switched path template. If you configure **ingress-replication label-switched-path** and do not include **label-switched-path-template**, ingress replication works with existing LDP or RSVP tunnels. If you include **label-switched-path-template**, the tunnels must be RSVP.

Options

existing-unicast-tunnel—An existing tunnel to the destination is used for ingress replication. If an existing tunnel is not available, the destination is not added. Default mode if no option is specified.

create-new-ucast-tunnel—When specified, a new unicast tunnel to the destination is created and used for ingress replication. The unicast tunnel is deleted later if the destination is no longer included in the multicast distribution tree.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs on page 591
- create-new-ucast-tunnel on page 994
- mpls-internet-multicast on page 1165

inet (AMT Protocol)

Syntax

```
inet {
  anycast-prefix ip-prefix/<prefix-length>;
  local-address ip-address;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols amt relay family],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay family],
[edit protocols amt relay family],
[edit routing-instances *routing-instance-name* protocols amt relay family]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Specify the IPv4 local address and anycast prefix for Automatic Multicast Tunneling (AMT) relay functions.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

inet-mdt

Syntax inet-mdt;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mvpn family *inet* | *inet6* autodiscovery],
[edit routing-instances *routing-instance-name* protocols pim mvpn family *inet* | *inet6* autodiscovery]

Release Information Statement introduced in Junos OS Release 9.4.
Statement moved to **[..protocols pim mvpn family inet]** from **[.. protocols mvpn]** in Junos OS Release 13.3.
Support for IPv6 added in Junos OS Release 17.3R1.

Description For Rosen 7, configure the PE router in a VPN to use an SSM multicast distribution tree (MDT) subsequent address family identifier (SAFI) NLRI .

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

inet-mvpn (BGP)

Syntax

```
inet-mvpn {
  signaling {
    accepted-prefix-limit {
      maximum number;
      teardown percentage {
        idle-timeout (forever | minutes);
      }
    }
    damping;
    loops number;
    prefix-limit {
      maximum number;
      teardown percentage {
        idle-timeout (forever | minutes);
      }
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp family],
[edit protocols bgp family],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family],
[edit protocols bgp group *group-name* family]

Release Information

Statement introduced in Junos OS Release 8.4.

Description

Enable the **inet-mvpn** address family in BGP.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

inet-mvpn (VRF Advertisement)

Syntax inet-mvpn;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* vrf-advertise-selective family],
[edit routing-instances *routing-instance-name* vrf-advertise-selective family]

Release Information Statement introduced in Junos OS Release 10.1.

Description Enable IPv4 MVPN routes to be advertised from the VRF instance.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Limiting Routes to Be Advertised by an MVPN VRF Instance*

inet6-mvpn (BGP)

Syntax

```
inet6-mvpn {
  signaling {
    accepted-prefix-limit {
      maximum number;
      teardown percentage {
        idle-timeout (forever | minutes);
      }
    }
    loops number
    prefix-limit {
      maximum number;
      teardown percentage {
        idle-timeout (forever | minutes);
      }
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols bgp family],
[edit protocols bgp family],
[edit logical-systems *logical-system-name* protocols bgp group *group-name* family],
[edit protocols bgp group *group-name* family]

Release Information

Statement introduced in Junos OS Release 10.0.

Description

Enable the **inet6-mvpn** address family in BGP.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *BGP Configuration Overview*

inet6-mvpn (VRF Advertisement)

Syntax inet6-mvpn;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* vrf-advertise-selective family],
[edit routing-instances *routing-instance-name* vrf-advertise-selective family],

Release Information Statement introduced in Junos OS Release 10.1.

Description Enable IPv6 MVPN routes to be advertised from the VRF instance.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

install (Multicast VLAN Registration)

Syntax install;

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) data-forwarding receiver]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.

Description Install forwarding entries in the multicast receiver VLAN. By default, the multicast VLAN (MVLAN) installs forwarding entries for MVLAN groups only.

Default Disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

interface (Bridge Domains)

Syntax

```
interface interface-name {
  group-limit limit;
  host-only-interface;
  static {
    group ip-address {
      source ip-address;
    }
  }
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Enable IGMP snooping on an interface and configure interface-specific properties.

Options

interface-name—Name of the interface. Specify the full interface name, including the physical and logical address components. To configure all interfaces, you can specify **all**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

interface (IGMP Snooping)

Syntax

```
interface interface-name {
  group-limit limit;
  host-only-interface;
  immediate-leave;
  multicast-router-interface;
  static {
    group multicast-group-address {
      source ip-address;
    }
  }
}
```

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*)]

Release Information

Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description For IGMP snooping, configure an interface as either a multicast-router interface or as a static member of a multicast group with optional interface-specific properties.

Options

all—All interfaces in the VLAN.

interface-name—Name of the interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- IGMP Snooping Overview on page 81
- **igmp-snooping on page 1077**

interface (MLD Snooping)

Syntax

```
interface (all | interface-name) {
  group-limit limit;
  host-only-interface;
  immediate-leave;
  multicast-router-interface;
  static {
    group ip-address {
      source ip-address;
    }
  }
}
```

Hierarchy Level

[edit protocols mld-snooping vlan (all | *vlan-name*)]
[edit routing-instances *instance-name* protocols mld-snooping vlan (*vlan-name*)]

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 18.1R1 for the SRX1500 devices.
Support at the **[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name*]** hierarchy introduced in Junos OS Release 13.3 for EX Series switches.
Support for the **group-limit**, **host-only-interface**, and the **immediate-leave** statements introduced in Junos OS Release 13.3 for EX Series switches.

Description

For MLD snooping, configure an interface as a static multicast-router interface, a host-side interface, or a static member of a multicast group.

Options

all—(All EX Series switches except EX9200) All interfaces in the VLAN.

interface-name—Name of the interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- **mld-snooping on page 1155**
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Understanding MLD Snooping on page 125

interface (Protocols DVMRP)

Syntax

```
interface interface-name {
  disable;
  hold-time seconds;
  metric metric;
  mode (forwarding | unicast-routing);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description

Enable DVMRP on an interface and configure interface-specific properties.

Options

interface-name—Name of the interface. Specify the full interface name, including the physical and logical address components. To configure all interfaces, you can specify **all**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring DVMRP on page 434

interface (Protocols IGMP)

Syntax

```
interface interface-name {
  (accounting | no-accounting);
  disable;
  distributed;
  group-limit limit;
  group-policy [ policy-names ];
  immediate-leave;
  oif-map map-name;
  passive;
  promiscuous-mode;
  ssm-map ssm-map-name;
  ssm-map-policy ssm-map-policy-name;
  static {
    group multicast-group-address {
      exclude;
      group-count number;
      group-increment increment;
      source ip-address {
        source-count number;
        source-increment increment;
      }
    }
  }
  version version;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Enable IGMP on an interface and configure interface-specific properties.

Options

interface-name—Name of the interface. Specify the full interface name, including the physical and logical address components. To configure all interfaces, you can specify **all**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP on page 28

interface (Protocols MLD)

Syntax

```
interface interface-name {
  (accounting | no-accounting);
  disable;
  distributed;
  group-limit limit;
  group-policy [ policy-names ];
  group-threshold value;
  immediate-leave;
  log-interval seconds;
  oif-map [ map-names ];
  passive;
  ssm-map ssm-map-name;
  ssm-map-policy ssm-map-policy-name;
  static {
    group multicast-group-address {
      exclude;
      group-count number
      group-increment increment
      source ip-address {
        source-count number;
        source-increment increment;
      }
    }
  }
  version version;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Enable MLD on an interface and configure interface-specific properties.

Options

interface-name—Name of the interface. Specify the full interface name, including the physical and logical address components. To configure all interfaces, you can specify **all**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD on page 54

interface (Protocols PIM)

Syntax

```
interface (Protocols PIM) (all | interface-name) {
  accept-remote-source;
  disable;
  bfd-liveness-detection {
    authentication {
      algorithmalgorithm-name;
      key-chainkey-chain-name;
      loose-check;
    }
    detection-time {
      threshold milliseconds;
    }
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    multiplier number;
    no-adaptation;
    transmit-interval {
      minimum-interval milliseconds;
      threshold milliseconds;
    }
    version (0 | 1 | automatic);
  }
  bidirectional {
    df-election {
      backoff-period milliseconds;
      offer-period milliseconds;
      robustness-count number;
    }
  }
  family (inet | inet6) {
    disable;
  }
  hello-interval seconds;
  mode (bidirectional-sparse | bidirectional-sparse-dense | dense | sparse | sparse-dense);
  neighbor-policy [ policy-names ];
  override-interval milliseconds;
  priority number;
  propagation-delay milliseconds;
  reset-tracking-bit;
  version version;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Enable PIM on an interface and configure interface-specific properties.

Options ***interface-name***—Name of the interface. Specify the full interface name, including the physical and logical address components. To configure all interfaces, you can specify **all**.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- PIM on Aggregated Interfaces on page 188

interface (Routing Options)

Syntax

```
interface interface-names {
  maximum-bandwidth bps;
  no-qos-adjust;
  reverse-oif-mapping {
    no-qos-adjust;
  }
  subscriber-leave-timer seconds;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Enable multicast traffic on an interface.

TIP: You cannot enable multicast traffic on an interface by using the **routing-options multicast interface** statement and configure PIM on the interface.

Options

interface-name—Names of the physical or logical interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Defining Interface Bandwidth Maximums on page 909
- Example: Configuring Multicast with Subscriber VLANs on page 912

interface (Scoping)

Syntax interface [*interface-names*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast scope *scope-name*],
[edit logical-systems *logical-system-name* routing-options multicast scope *scope-name*],
[edit routing-instances *routing-instance-name* routing-options multicast scope *scope-name*],
[edit routing-options multicast scope *scope-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Configure the set of interfaces for multicast scoping.

Options ***interface-names***—Names of the interfaces to scope. Specify the full interface name, including the physical and logical address components. To configure all interfaces, you can specify **all**.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast Snooping on page 874

interface (Virtual Tunnel in Routing Instances)

Syntax

```
interface vt-fpc/pic/port.unit-number {
  multicast;
  primary;
  unicast;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced in Junos OS Release 9.4.

Description In a multiprotocol BGP (MBGP) multicast VPN (MVPN), configure a virtual tunnel (VT) interface.

VT interfaces are needed for multicast traffic on routing devices that function as combined provider edge (PE) and provider core (P) routers to optimize bandwidth usage on core links. VT interfaces prevent traffic replication when a P router also acts as a PE router (an exit point for multicast traffic).

In an MBGP MVPN extranet, if there is more than one VRF routing instance on a PE router that has receivers interested in receiving multicast traffic from the same source, VT interfaces must be configured on all instances.

Starting in Junos OS Release 12.3, you can configure multiple VT interfaces in each routing instance. This provides redundancy. A VT interface can be used in only one routing instance.

Options ***vt-fpc/pic/port.unit-number***—Name of the VT interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 744
- Example: Configuring MBGP MVPN Extranets on page 669

interface-name

Syntax interface-name *interface-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim default-vpn-source],
[edit protocols pim default-vpn-source]

Release Information Statement introduced in Junos OS Release 10.1.

Description Specify the primary loopback address configured in the default routing instance to use as the source address when PIM hello messages, join messages, and prune messages are sent over multicast tunnel interfaces for interoperability with other vendors' routers.

Options **interface-name**—Primary loopback address configured in the default routing instance to use as the source address when PIM control messages are sent. Typically, the **lo0.0** interface is specified for this purpose.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

intra-as

Syntax

```
intra-as {
    inclusive;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn family *inet* | *inet6* autodiscovery-only],
[edit routing-instances *routing-instance-name* protocols mvpn family *inet* | *inet6* autodiscovery-only,]

Release Information

Statement introduced in Junos OS Release 9.4.
Statement moved to **[..protocols mvpn family inet]** from **[.. protocols mvpn]** in Junos OS Release 13.3.
Support for IPv6 added in Junos OS Release 17.3R1.

Description

For Rosen 7, enable the MVPN control plane for autodiscovery only, using intra-AS autodiscovery routes.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

join-load-balance

Syntax

```
join-load-balance {
  automatic;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Enable load balancing of PIM join messages across interfaces and routing devices.

Options

automatic—Enables automatic load balancing of PIM join messages. When a new interface or neighbor is introduced into the network, ECMP joins are redistributed with minimal disruption to traffic.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM Make-Before-Break Join Load Balancing on page 790
- Configuring PIM Join Load Balancing on page 218
- clear pim join-distribution on page 1418

join-prune-timeout

Syntax join-prune-timeout *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced in Junos OS Release 8.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the timeout for the join state. If the periodic join refresh message is not received before the timeout expires, the join state is removed.

Options **seconds**—Number of seconds to wait for the periodic join message to arrive.

Range: 210 through 240 seconds

Default: 210 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the Join State Timeout on page 222

keep-alive (Protocols MSDP)

Syntax keep-alive *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer address],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name* peer *address*]
[edit routing-instances *instance-name* protocols msdp peer *address*],

Release Information Statement introduced in Junos OS Release 12.3.

Description Specify the keepalive interval to use when maintaining a connection with the MSDP peer. If a keepalive message is not received for the hold-time period, the MSDP peer connection is terminated. According to the RFC 3618, *Multicast Source Discovery Protocol (MSDP)*, the recommended value for the keepalive timer is 60 seconds.

The hold-time period must be longer than the keepalive interval.

You might want to change the keepalive interval and hold-time period for consistency in a multi-vendor environment.

Default In Junos OS, the default hold-time period is 75 seconds, and the default keepalive interval is 60 seconds.

Options ***seconds***—Keepalive interval.

Range: 10 through 60 seconds

Default: 60 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring MSDP on page 393
- hold-time (Protocols MSDP) on page 1068
- sa-hold-time (Protocols MSDP) on page 1271

key-chain (Protocols PIM)

Syntax key-chain *key-chain-name*;

Hierarchy Level [edit protocols pim interface *interface-name* family {inet | inet6} bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* family {inet | inet6} bfd-liveness-detection authentication]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement modified in Junos OS Release 12.2 to include **family** in the hierarchy level.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the security keychain to use for BFD authentication.

Options *key-chain-name*—Name of the security keychain to use for BFD authentication. The name is a unique integer between **0** and **63**. This must match one of the keychains in the **authentication-key-chains** statement at the **[edit security]** hierarchy level.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD Authentication for PIM on page 196
- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 357
- authentication on page 974

l2-querier

Syntax

```
l2-querier {
  source-address ip-address;
}
```

Hierarchy Level [edit protocols igmp-snooping vlan],

Release Information Statement introduced in Junos OS Release 13.2 for the QFX Series.
Statement introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description Configure the device to be an IGMP querier. IGMP querier allows the device to proxy for a multicast router and send out periodic IGMP queries in the network. This action causes the device to consider itself an multicast router port. The remaining devices in the network simply define their respective multicast router ports as the interface on which they received this IGMP query. Use the **source-address** statement to configure the source address to use for IGMP snooping queries.

Options *seconds*—Time interval.

Range: 1 through 1024

Default: 125 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- IGMP Snooping Overview on page 81
- igmp-snooping on page 1077

label-switched-path-template (Multicast)

Syntax

```
label-switched-path-template {
    (default-template | lsp-template-name);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel ingress-replication label-switched-path],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit logical-systems *logical-system-name* routing-options dynamic-tunnels *tunnel-name* rsvp-te *entry-name*],
[edit protocols mvpn inter-region-segmented template *template-name* region *region-name* ingress-replication label-switched-path],
[edit protocols mvpn inter-region-segmented template *template-name* region *region-name* rsvpe-te],
[edit protocols mvpn inter-region-template template *template-name* all-regions ingress-replication label-switched-path],
[edit protocols mvpn inter-region-template template *template-name* all-regions rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel ingress-replication label-switched-path],
[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit routing-options dynamic-tunnels *tunnel-name* rsvp-te *entry-name*]
[edit routing-instances *instance-name* provider-tunnel]

Release Information

Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 18.2. under the heirarchy level [edit routing-instances *instance-name* provider-tunnel]

Description

Specify the LSP template. An LSP template is used as the basis for other dynamically generated LSPs. This feature can be used for a number of applications, including point-to-multipoint LSPs, flooding VPLS traffic, configuring ingress replication for IP multicast using MBGP MVPNs, and to enable RSVP automatic mesh. There is no default setting for the **label-switched-path-template** statement, so you must configure either the default-template using the **default-template** option, or you must specify the name of your preconfigured LSP template.

Options

default-template—Specify that the default LSP template be used for the dynamically generated LSPs.

lsp-template-name—Specify the name of an LSP to be used as a template for the dynamically generated LSPs.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs on page 591
- *Configuring RSVP-Signaled Inclusive Point-to-Multipoint LSPs for an MBGP MVPN*
- *Configuring Dynamic Point-to-Multipoint Flooding LSPs*
- *Configuring RSVP Automatic Mesh*

ldp-p2mp

Syntax ldp-p2mp;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel],
[edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective group *group-prefix* wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective group *group-prefix* source *source-prefix*],
[edit protocols mvpn inter-region-template template *template-name* all-regions],
[edit protocols mvpn inter-region-template template *template-name* region *region-name*],
[edit routing-instances *instance-name* provider-tunnel],
[edit routing-instances *instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit routing-instances *instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source],
[edit routing-instances *instance-name* provider-tunnel selective group *group-prefix* wildcard-source],
[edit routing-instances *instance-name* provider-tunnel selective group *group-prefix* source *source-prefix*]
[edit routing-instances *instance-name* provider-tunnel]

Release Information
Statement introduced in Junos OS Release 11.2.
Statement introduced in Junos OS Release 18.2. under the heirarchy level [edit routing-instances *instance-name* provider-tunnel]

Description Specify a point-to-multipoint provider tunnel with LDP signalling for an MBGP MVPN.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Point-to-Multipoint LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs on page 586

leaf-tunnel-limit-inet (MVPN Selective Tunnels)

Syntax leaf-tunnel-limit-inet *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective],
[edit routing-instances *instance-name* provider-tunnel selective]

Release Information Statement introduced in Junos OS Release 13.3.

Description Configure the maximum number of selective leaf tunnels for IPv4 control-plane routes.

The purpose of the **leaf-tunnel-limit-inet** statement is to supplement the multicast forwarding-cache limit when the MVPN **rpt-spt** mode is configured and when traffic is flowing through selective service provider multicast service inteface (S-PMSI) tunnels and is forwarded by way of the (*,G) entry, even though the forwarding cache limit has already blocked the forwarding entries from being created.

The **leaf-tunnel-limit-inet** statement limits the number of Type-4 leaf autodiscovery (AD) route messages that can be originated by receiver provider edge (PE) routers in response to receiving from the sender PE router S-PMSI AD routes with the leaf-information-required flag set. Thus, this statement limits the number of leaf nodes that are created when a selective tunnel is formed.

You can configure the statement only when the MVPN mode is **rpt-spt**.

This statement is independent of the **cmcast-joins-limit-inet** statement and of the **forwarding-cache** threshold statement.

Setting the **leaf-tunnel-limit-inet** statement or reducing the value of the limit does not alter or delete the already existing and installed routes. If needed, you can run the **clear pim join** command to force the limit to take effect. Those routes that cannot be processed because of the limit are added to a queue, and this queue is processed when the limit is removed or increased and when existing routes are deleted.

Default Unlimited

Options ***number***—Maximum number of selective leaf tunnels for IPv4.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring the Multicast Forwarding Cache on page 928
- Example: Configuring MBGP Multicast VPN Topology Variations on page 648

leaf-tunnel-limit-inet6 (MVPN Selective Tunnels)

Syntax leaf-tunnel-limit-inet6 *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* provider-tunnel selective],
[edit routing-instances *instance-name* provider-tunnel selective]

Release Information Statement introduced in Junos OS Release 13.3.

Description Configure the maximum number of selective leaf tunnels for IPv6 control-plane routes.

The purpose of the **leaf-tunnel-limit-inet6** statement is to supplement the multicast forwarding-cache limit when the MVPN **rpt-spt** mode is configured and when traffic is flowing through selective service provider multicast service inteface (S-PMSI) tunnels and is forwarded by way of the (*,G) entry, even though the forwarding cache limit has already blocked the forwarding entries from being created.

The **leaf-tunnel-limit-inet6** statement limits the number of Type-4 leaf autodiscovery (AD) route messages that can be originated by receiver provider edge (PE) routers in response to receiving from the sender PE router S-PMSI AD routes with the leaf-information-required flag set. Thus, this statement limits the number of leaf nodes that are created when a selective tunnel is formed.

You can configure the statement only when the MVPN mode is **rpt-spt**.

This statement is independent of the **cmcast-joins-limit-inet6** statement and of the **forwarding-cache** threshold statement.

Setting the **leaf-tunnel-limit-inet6** statement or reducing the value of the limit does not alter or delete the already existing and installed routes. If needed, you can run the **clear pim join** command to force the limit to take effect. Those routes that cannot be processed because of the limit are added to a queue, and this queue is processed when the limit is removed or increased and when existing routes are deleted.

Default Unlimited

Options ***number***—Maximum number of selective leaf tunnels for IPv6.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring the Multicast Forwarding Cache on page 928
- Example: Configuring MBGP Multicast VPN Topology Variations on page 648

listen

Syntax listen *address* <port *port*>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols sap],
[edit protocols sap]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify an address and optionally a port on which SAP and SDP listen, in addition to the default SAP address and port on which they always listen, 224.2.127.254:9875. To specify multiple additional addresses or pairs of address and port, include multiple **listen** statements.

Options ***address***—(Optional) Address on which SAP listens for session advertisements.

Default: 224.2.127.254

port *port*—(Optional) Port on which SAP listens for session advertisements.

Default: 9875

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Session Announcement Protocol on page 415

local

Syntax

```
local {
  disable;
  address address;
  family (inet | inet6) {
    disable;
    address address;
    anycast-pim {
      local-address address;
        rp-set {
        address address <forward-msdp-sa>;
      }
    }
    group-ranges {
      destination-ip-prefix</prefix-length>;
    }
    hold-time seconds;
    override;
    priority number;
  }
  group-ranges {
    destination-ip-prefix</prefix-length>;
  }
  hold-time seconds;
  override;
  priority number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the routing device's RP properties.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Local PIM RPs on page 237

local-address (Protocols AMT)

Syntax local-address *ip-address*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols amt relay family inet],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay family inet],
[edit protocols amt relay family inet],
[edit routing-instances *routing-instance-name* protocols amt relay family inet]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify the local unique IP address to send in Automatic Multicast Tunneling (AMT) relay advertisement messages, for use as the IP source of AMT control messages, and as the source of the data tunnel encapsulation. The address can be configured on any interface in the system. Typically, the router's **lo0.0** loopback address is used for configuring the AMT local address in the default routing instance, and the router's **lo0.*n*** loopback address is used for configuring the AMT local address in VPN routing instances.

Default None. The local address must be configured.

Options ***ip-address***—Unique unicast IP address.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

local-address (Protocols MSDP)

Syntax local-address *address*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the local end of an MSDP session. You must configure at least one peer for MSDP to function. When configuring a peer, you must include this statement. This address is used to accept incoming connections to the peer and to establish connections to the remote peer.

Options ***address***—IP address of the local end of the connection.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396

local-address (Protocols PIM)

Syntax local-address *address*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp local family (inet | inet6) anycast-pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6) anycast-pim],
[edit protocols pim rp local family (inet | inet6) anycast-pim],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6) anycast-pim]

Release Information
Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the routing device local address for the anycast rendezvous point (RP). If this statement is omitted, the router ID is used as this address.

Options ***address***—Anycast RP IPv4 or IPv6 address, depending on **family** configuration.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring PIM Anycast With or Without MSDP on page 248

local-address (Routing Options)

Syntax local-address *address*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast backup-pe-group *group-name*],
[edit logical-systems *logical-system-name* routing-options multicast backup-pe-group *group-name*],
[edit routing-instances *routing-instance-name* routing-options multicast backup-pe-group *group-name*],
[edit routing-options multicast backup-pe-group *group-name*]

Release Information
Statement introduced in Junos OS Release 9.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement added to the multicast hierarchy in Junos OS Release 13.2.

Description Configure the address of the local PE for ingress PE redundancy when point-to-multipoint LSPs are used for multicast distribution.

Options ***address***—Address of local PEs in the backup group.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Ingress PE Redundancy on page 937

log-interval (PIM Entries)

Syntax log-interval *value*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim sglimit],
[edit logical-systems *logical-system-name* protocols pim sglimit *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim sglimit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim sglimit *family*],
[edit protocols pim sglimit],
[edit protocols pim sglimit *family*],
[edit routing-instances *routing-instance-name* protocols pim sglimit],
[edit routing-instances *routing-instance-name* protocols pim sglimit *family*],
[edit logical-systems *logical-system-name* protocols pim rp group-rp-mapping],
[edit logical-systems *logical-system-name* protocols pim rp group-rp-mapping *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp group-rp-mapping],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp group-rp-mapping *family*],
[edit protocols pim rp group-rp-mapping],
[edit protocols pim rp group-rp-mapping *family*],
[edit routing-instances *routing-instance-name* protocols pim rp group-rp-mapping],
[edit routing-instances *routing-instance-name* protocols pim rp group-rp-mapping *family*],
[edit logical-systems *logical-system-name* protocols pim rp register-limit],
[edit logical-systems *logical-system-name* protocols pim rp register-limit *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp register-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp register-limit *family*],
[edit protocols pim rp register-limit],
[edit protocols pim rp register-limit *family*],
[edit routing-instances *routing-instance-name* protocols pim rp register-limit],
[edit routing-instances *routing-instance-name* protocols pim rp register-limit *family*],

Release Information Statement introduced in Junos OS Release 12.2.

Description Configure the amount of time between log messages.

Options *seconds*—Minimum time interval (in seconds) between log messages. To configure the time interval, you must explicitly configure the maximum number of entries received with the **maximum** statement. You can apply the log interval to incoming PIM join messages, PIM register messages, and group-to-RP mappings.

Range: 1 through 65,535

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- add new concept and example topic to related topic list.
- clear pim join on page 1416

log-interval (IGMP Interface)

Syntax log-interval *seconds*;

Hierarchy Level [edit dynamic-profiles *profile-name* protocols igmp interface *interface-name*]
[edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.2.

Description Specify the minimum time interval (in seconds) between sending consecutive log messages to the system log for multicast groups on static or dynamic IGMP interfaces. To configure the time interval, you must specify the maximum number of multicast groups allowed on the interface. You must configure the group-limit statement before you configure the **log-interval** statement.

To confirm the configured log interval on the interface, use the show igmp interface command.

Default By default, there is no configured time interval.

Options ***seconds***—Minimum time interval (in seconds) between log messages. You must explicitly configure the **group-limit** to configure a time interval to send log messages.

Range: 6 through 32,767 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Limiting the Number of IGMP Multicast Group Joins on Logical Interfaces on page 45
- group-limit on page 1054
- group-threshold on page 1063

log-interval (MLD Interface)

Syntax log-interval *seconds*;

Hierarchy Level [edit dynamic-profiles *profile-name* protocols mld interface *interface-name*]
[edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.2.

Description Specify the minimum time interval (in seconds) between sending consecutive log messages to the system log for multicast groups on static or dynamic MLD interfaces. To configure the time interval, you must specify the maximum number of multicast groups allowed on the interface.

To confirm the configured log interval on the interface, use the show mld interface command.

Default By default, there is no configured time interval.

Options ***seconds***—Minimum time interval (in seconds) between log messages. You must explicitly configure the **group-limit** to configure a time interval to send log messages.

Range: 6 through 32,767 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Number of MLD Multicast Group Joins on Logical Interfaces on page 72
- group-limit on page 1056
- group-threshold on page 1064

log-interval (Protocols MSDP)

Syntax log-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols msdp active-source-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp active-source-limit],
[edit protocols msdp active-source-limit],
[edit routing-instances *routing-instance-name* protocols msdp active-source-limit]

Release Information Statement introduced in Junos OS Release 12.2

Description Specify the minimum time interval (in seconds) between sending consecutive log messages to the system log for MSDP active source messages. To configure the time interval, you must specify the maximum number of MSDP active source messages received by the device.

To confirm the configured log interval, use the show msdp source-active command.

Options ***seconds***—Minimum time interval (in seconds) between log messages. You must explicitly configure the maximum value to configure a time interval to send log messages.

Range: 6 through 32,767 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404
- log-warning
- maximum on page 1139

log-warning (Protocols MSDP)

Syntax log-warning *value*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols msdp active-source-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp active-source-limit],
[edit protocols msdp active-source-limit],
[edit routing-instances *routing-instance-name* protocols msdp active-source-limit]

Release Information Statement introduced in Junos OS Release 12.2

Description Specify the threshold at which the device logs a warning message in the system log for received MSDP active source messages. This threshold is a percentage of the maximum number of MSDP active source messages received by the device.

To confirm the configured warning threshold, use the show msdp source-active command.

Options *value*—Percentage of the number of active source messages that starts triggering the warnings. You must explicitly configure the maximum value to configure a warning threshold value.

Range: 1 through 100

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404
- log-interval
- maximum on page 1139

log-warning (Multicast Forwarding Cache)

Syntax log-warning *value*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache threshold],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache family (inet | inet6)threshold],
[edit logical-systems *logical-system-name* routing-options multicast forwarding-cache threshold],
[edit logical-systems *logical-system-name* routing-options multicast forwarding-cache family (inet | inet6) threshold],
[edit routing-instances *routing-instance-name* routing-options multicast forwarding-cache threshold],
[edit routing-instances *routing-instance-name* routing-options multicast forwarding-cache family (inet | inet6) threshold],
[edit routing-options multicast forwarding-cache threshold],
[edit routing-options multicast forwarding-cache family (inet | inet6) threshold]

Release Information Statement introduced in Junos OS Release 12.2.

Description Specify the threshold at which the device logs a warning message in the system log for multicast forwarding cache entries. This threshold is a percentage of the maximum number of multicast forwarding cache entries received by the device. Configuring the **threshold** statement globally for the multicast forwarding cache or including the **family** statement to configure the thresholds for the IPv4 and IPv6 multicast forwarding caches are mutually exclusive.

To confirm the configured warning threshold, use the show multicast forwarding-cache statistics command.

Options *value*—Percentage of the number of multicast forwarding cache entries that can be added to the cache that starts triggering the warning. You must explicitly configure the **suppress** value to configure a warning threshold value.

Range: 1 through 100

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring the Multicast Forwarding Cache on page 929

loose-check

Syntax loose-check;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection authentication],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection authentication]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify loose authentication checking on the BFD session. Use loose authentication for transitional periods only when authentication might not be configured at both ends of the BFD session.

By default, strict authentication is enabled and authentication is checked at both ends of each BFD session. Optionally, to smooth migration from nonauthenticated sessions to authenticated sessions, you can configure *loose checking*. When loose checking is configured, packets are accepted without authentication being checked at each end of the session.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD Authentication for PIM on page 196
- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 357
- **authentication on page 974**

mapping-agent-election

Syntax (mapping-agent-election | no-mapping-agent-election);

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp auto-rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp auto-rp],
[edit protocols pim rp auto-rp],
[edit routing-instances *routing-instance-name* protocols pim rp auto-rp]

Release Information Statement introduced in Junos OS Release 7.5.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the routing device mapping announcements as a mapping agent.

Options

mapping-agent-election—Mapping agents do not announce mappings when receiving mapping messages from a higher-addressed mapping agent.

no-mapping-agent-election—Mapping agents always announce mappings and do not perform mapping agent election.

Default: mapping-agent-election

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Auto-RP on page 258

maximum (MSDP Active Source Messages)

Syntax maximum *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols msdp active-source-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp active-source-limit],
[edit protocols msdp active-source-limit],
[edit routing-instances *routing-instance-name* protocols msdp active-source-limit]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the maximum number of MSDP active source messages the router accepts.

Options *number*—Maximum number of active source messages.

Range: 1 through 1,000,000

Default: 25,000

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404
- threshold (MSDP Active Source Messages) on page 1320

maximum (PIM Entries)

Syntax maximum *limit*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim sglimit],
[edit logical-systems *logical-system-name* protocols pim sglimit *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim sglimit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim sglimit *family*],
[edit protocols pim sglimit],
[edit protocols pim sglimit *family*],
[edit routing-instances *routing-instance-name* protocols pim sglimit],
[edit routing-instances *routing-instance-name* protocols pim sglimit *family*],
[edit logical-systems *logical-system-name* protocols pim rp group-rp-mapping],
[edit logical-systems *logical-system-name* protocols pim rp group-rp-mapping *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp group-rp-mapping],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp group-rp-mapping *family*],
[edit protocols pim rp group-rp-mapping],
[edit protocols pim rp group-rp-mapping *family*],
[edit routing-instances *routing-instance-name* protocols pim rp group-rp-mapping],
[edit routing-instances *routing-instance-name* protocols pim rp group-rp-mapping *family*],
[edit logical-systems *logical-system-name* protocols pim rp register-limit],
[edit logical-systems *logical-system-name* protocols pim rp register-limit *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp register-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp register-limit *family*],
[edit protocols pim rp register-limit],
[edit protocols pim rp register-limit *family*],
[edit routing-instances *routing-instance-name* protocols pim rp register-limit],
[edit routing-instances *routing-instance-name* protocols pim rp register-limit *family*],

Release Information Statement introduced in Junos OS Release 12.2.

Description Configure the maximum number of specified PIM entries received by the device. If the device reaches the configured limit, no new entries are received.

NOTE: The maximum limit settings that you configure with the **maximum** and the **family (inet | inet6) maximum** statements are mutually exclusive. For example, if you configure a global maximum PIM join state limit, you cannot configure a limit at the family level for IPv4 or IPv6 joins. If you attempt to configure a limit at both the global level and the family level, the device will not accept the configuration.

Options *limit*—Maximum number of PIM entries received by the device. If you configure both the **log-interval** and the **maximum** statements, a warning is triggered when the maximum limit is reached.

Depending on your configuration, this limit specifies the maximum number of PIM joins, PIM register messages, or group-to-RP mappings received by the device.

Range: 1 through 65,535

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- add new concept and example topic to related topic list.
- clear pim join on page 1416

maximum-bandwidth

Syntax maximum-bandwidth *bps*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit logical-systems *logical-system-name* routing-options multicast interface *interface-name*],
[edit routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit routing-options multicast interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the multicast bandwidth for the interface.

Options *bps*—Bandwidth rate, in bits per second, for the multicast interface.

Range: 0 through any amount of bandwidth

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Defining Interface Bandwidth Maximums on page 909

maximum-rps

Syntax maximum-rps *limit*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim rp embedded-rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp embedded-rp],
[edit protocols pim rp embedded-rp],
[edit routing-instances *routing-instance-name* protocols pim rp embedded-rp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Limit the number of RPs that the routing device acknowledges.

Options *limit*—Number of RPs.

Range: 1 through 500

Default: 100

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Embedded RP for IPv6 on page 266

maximum-transmit-rate (Protocols IGMP)

Syntax maximum-transmit-rate *packets-per-second*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced in Junos OS Release 9.3.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Limit the transmission rate of IGMP packets

Options **packets-per-second**—Maximum number of IGMP packets transmitted in one second by the routing device.

Range: 1 through 10000

Default: 500 packets

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Limiting the Maximum IGMP Message Rate on page 36

maximum-transmit-rate (Protocols MLD)

Syntax maximum-transmit-rate *packets-per-second*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced in Junos OS Release 9.3.

Description Limit the transmission rate of MLD packets.

Options **packets-per-second**—Maximum number of MLD packets transmitted in one second by the routing device.

Range: 1 through 10000

Default: 500 packets

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Limiting the Maximum MLD Message Rate on page 63

mdt

Syntax

```
mdt {
  data-mdt-reuse;
  group-range multicast-prefix;
  threshold {
    group group-address {
      source source-address {
        rate threshold-rate;
      }
    }
    tunnel-limit limit;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*]

Release Information

Statement introduced before Junos OS Release 7.4. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description

Establish the group address range for data MDTs, the threshold for the creation of data MDTs, and tunnel limits for a multicast group and source. A multicast group can have more than one source of traffic.

The remaining statements are explained separately. .

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

metric (Protocols DVMRP)

Syntax metric *metric*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols dvmrp interface *interface-name*], [edit protocols dvmrp interface *interface-name*]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description Define the DVMRP metric value.

Options ***metric***—Metric value.

Range: 1 through 31

Default: 1

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring DVMRP on page 434

minimum-interval (PIM BFD Liveness Detection)

Syntax minimum-interval *milliseconds*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the minimum interval after which the local routing device transmits hello packets and then expects to receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can specify the minimum transmit and receive intervals separately using the **transmit-interval minimum-interval** and **minimum-receive-interval** statements.

Options ***milliseconds***—Minimum transmit and receive interval.

Range: 1 through 255,000 milliseconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194

minimum-interval (PIM BFD Transmit Interval)

Syntax minimum-interval *milliseconds*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection transmit-interval]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the minimum interval after which the local routing device transmits hello packets to a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can configure the minimum transmit interval using the **minimum-interval** statement at the **[edit protocols pim interface *interface-name* bfd-liveness-detection]** hierarchy level.

Options ***milliseconds***—Minimum transmit interval value.

Range: 1 through 255,000

NOTE: The threshold value specified in the **threshold** statement must be greater than the value specified in the **minimum-interval** statement for the **transmit-interval** statement.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- bfd-liveness-detection on page 984
- minimum-interval on page 1147
- threshold on page 1324

min-rate

Syntax

```
min-rate {
  rate bps;
  revert-delay seconds;
}
```

Hierarchy Level [edit routing-instances *routing-instance-name* protocols mvpn hot-root-standby]

Release Information Statement introduced in Junos OS Release 16.1.

Description Fast failover (that is, sub-50ms switch over for C-multicast streams as defined in *Draft Morin L3VPN Fast Failover 05,*) is supported for MPC cards operating in **enhanced-ip** mode that are running next generation (NG) MVPNs with **hot-root-standby** enabled.

Live-live NG MVPN traffic is available by enabling both sender-based reverse path forwarding (RPF) and hot-root standby. In this scenario, any upstream failure in the network can be repaired locally at the egress PE, and fast failover is triggered if the flow rate of monitored traffic falls below the threshold configured for **min-rate**.

On the egress PE, redundant multicast streams are received from a source that has been multihomed to two or more senders (upstream PEs). Only one stream is forwarded to the customer network, however, because the sender-based RPF running on the egress PE prevents any duplication.

Note that fast failover only supports VRF configured with a virtual tunnel (VT) interface, that is, anchored to a tunnel PIC to provide upstream tunnel termination. Label switched interfaces (LSI) are not supported.

NOTE: **min-rate is not strictly supported for MPC3 and MPC4 line cards (these cards have multiple lookup chips and an aggregate value is not calculated across chips). So, when setting the rate, choose a value that is high enough to ensure that lookup will be triggered at least once on each chip every 10 milliseconds or less. As a result, for line cards with multiple look up chips, a small percentage of duplicate multicast packets may be observed being leaked to the to the egress interface. This is normal behavior. The re-route is triggered when traffic rate on the primary tunnel hits zero. Likewise, if no packets are detected on any of the lookup chips during the configured interval, the tunnel will go down.**

Options

rate—Specify a rate to represent the typical flow rate of aggregate multicast traffic from the provider tunnel (P tunnel). Aggregate multicast traffic from the P tunnel is monitored, and if it falls below the threshold set here a failover to the hot-root standby is triggered.

Range: 4 Mb through 100 Gb

revert-delay *seconds*—Use the specified interval to allow time for the network to converge when and if the original link comes back online. You can specify a time, in seconds, for the router to wait before updating its multicast routes. For example, if the original link goes down and triggers the switchover to an alternative link, and then the original link comes back up, the update of multicast routes reflecting the new path can be delayed to accommodate the time it may take to for the network to converge back on the original link.

Range: 0 through 20 seconds

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- **hot-root-standby on page 1072**

min-rate (source-active-advertisement)

Syntax min-rate *bps*

Hierarchy Level [edit logical-systems *logical-system--name* routing-instances *instance-name* protocols mvpn mvpn-mode spt-only source-active-advertisement],
[edit routing-instances *instance-name* protocols mvpn mvpn-mode spt-only source-active-advertisement],
[edit routing-instances *instance-name* protocols mvpn mvpn-mode spt-only source-active-advertisement]

Release Information Statement introduced in Junos OS Release 17.1.

Description Minimum traffic rate required to advertise Source-Active route (1 to 1000000 bits per second), set on the ingress PEs.

Use the command, for example, to ensure that the egress PEs only receive Source-Active A-D route advertisements from ingress PEs that are receiving traffic at or above a minimum rate, regardless of how many ingress PEs there may be. Only one of the ingress PEs is chosen as the upstream multicast hop (UMH). Traffic flow continues because the egress PE removes its Type 7 advertisements to the old UMH and re-advertises a Type 7 to the new UMH.

The **min-rate** command works by polling traffic stats to determine the traffic rate of each flow on the ingress PE. Rather than advertising the Source-Active A-D route immediately upon learning of the S,G, the ingress PE waits until the traffic rate reaches the threshold set for **min-rate** before sending the Source-Active A-D route. If the rate then drops below the threshold, the Source-Active A-D route is withdrawn.

To verify that the value is set as expected, you can check whether the Type 5 (Source-Active route) has been advertised using the **show route table *vrf*.mvpn.0** command. It may take several minutes before you can see the changes in the Source-Active A-D route advertisement after making changes to the **min-rate**.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*
- dampen on page 995

minimum-receive-interval

Syntax minimum-receive-interval *milliseconds*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the minimum interval after which the local routing device must receive a reply from a neighbor with which it has established a BFD session. Optionally, instead of using this statement, you can configure the minimum receive interval using the **minimum-interval** statement at the **[edit protocols pim interface *interface-name* bfd-liveness-detection]** hierarchy level.

Options ***milliseconds***—Minimum receive interval.

Range: 1 through 255,000 milliseconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194

mld

Syntax

```
mld {
  accounting;
  interface interface-name {
    (accounting | no-accounting);
    disable;
    distributed;
    group-limit limit;
    group-policy [ policy-names ];
    immediate-leave;
    oif-map [ map-names ];
    passive;
    ssm-map ssm-map-name;
    ssm-map-policy ssm-map-policy-name;
    static {
      group multicast-group-address {
        exclude;
        group-count number;
        group-increment increment;
        source ip-address {
          source-count number;
          source-increment increment;
        }
      }
    }
    version version;
  }
  maximum-transmit-rate packets-per-second;
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols],
[edit protocols]

Release Information Statement introduced before Junos OS Release 7.4.

Description Enable MLD on the router. MLD must be enabled for the router to receive multicast packets.

Default MLD is disabled on the router. MLD is automatically enabled on all broadcast interfaces when you configure Protocol Independent Multicast (PIM) or Distance Vector Multicast Routing Protocol (DVMRP).

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD on page 54
- show mld group on page 1503
- show mld interface on page 1507
- show mld statistics on page 1511
- clear mld membership on page 1402
- clear mld statistics on page 1405

mld-snooping

List of Syntax

Syntax (SRX Series, EX Series) on page 1155
Syntax (MX Series) on page 1155

Syntax (SRX Series, EX Series)

```
mld-snooping {
  vlan (all | vlan-name) {
    immediate-leave;
    interface interface-name {
      group-limit;
      host-only-interface;
      immediate-leave;
      multicast-router-interface;
      static {
        group ip-address {
        source ip-address;
        }
      }
    }
    qualified-vlan vlan-id;
    query-interval;
    query-last-member-interval;
    query-response-interval;
    robust-count number;
    trace-options {
      file (files | no-word-readable | size | word-readable):
      flag (all | client-notification | general | group | host-notification | leave | noraml | packest
        | policy | query | report | route | report | state | task |timer):
    }
  }
}
```

Syntax (MX Series)

```
mld-snooping {
  immediate-leave;
  interface interface-name {
    group-limit limit;
    host-only-interface;
    immediate-leave;
    multicast-router-interface;
    static {
      group ip-address {
        source ip-address;
      }
    }
  }
  proxy {
    source-address ip-address;
  }
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
  }
```

```
    vlan vlan-id {
      immediate-leave;
      interface interface-name {
        group-limit limit;
        host-only-interface;
        immediate-leave;
        multicast-router-interface;
        static {
          group ip-address {
            source ip-address;
          }
        }
      }
      proxy {
        source-address ip-address;
      }
      query-interval seconds;
      query-last-member-interval seconds;
      query-response-interval seconds;
      robust-count number;
    }
  }
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 14.2 for MX Series routers with MPC.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description Enable and configure MLD snooping. MLD snooping constrains IPv6 multicast traffic at Layer 2 by configuring Layer 2 LAN ports dynamically to forward IPv6 multicast traffic only to those ports that want to receive it.

More Information: Multicast Listener Discovery (MLD) is a protocol built on ICMPv6 and used by IPv6 routers and hosts to discover and indicate interest in a multicast group. There are two versions, MLDv1 (RFC 2710) which is equivalent to IGMPv2, and MLDv2 (RFC 3810), which is equivalent to IGMPv3. Both MLDv1 and MLDv2 support Query, Report and Done messages, just as IGMP. MLDv2 further supports source-specific Queries/Reports and multi-record Reports.

For MX Series devices, MLD snooping restricts the forwarding of IPv6 multicast traffic to only those interfaces in a bridge-domain/VPLS that have interested listeners. Rather than flooding all interfaces in the bridge-domain/VPLS, MLD snooping restricts the forwarding of IPv6 multicast traffic to only those interfaces in a bridge-domain/VPLS that have interested listeners. These interfaces are identified by snooping MLD control packets, identifying the set of outgoing interfaces for a multicast stream, and building forwarding state accordingly. Queries will be snooped and flooded to all ports; Report and Done messages are snooped and selectively forwarded to multicast router ports only.

NOTE: **For MX Series devices, MLD snooping is supported on MPC-1, MPC-2, MPC-3, and MPC-4 linecards (Trio based). It is not supported on DPC linecards. The operational commands for mld-snooping, including defaults, functionality, logging, and tracing are the same as for igmp-snooping.**

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MLD Snooping on EX Series Switches on page 148
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Understanding MLD Snooping on page 125

mode (Protocols DVMRP)

Syntax mode (forwarding | unicast-routing);

Hierarchy Level [edit logical-systems *logical-system-name* protocols dvmrp interface *interface-name*], [edit protocols dvmrp interface *interface-name*]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description Configure DVMRP for multicast traffic forwarding or unicast routing.

Options **forwarding**—DVMRP performs unicast routing as well as multicast data forwarding.

unicast-routing—DVMRP performs unicast routing only. To forward multicast data, you must configure Protocol Independent Multicast (PIM) on the interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring DVMRP to Announce Unicast Routes on page 438

mode (Protocols MSDP)

Syntax mode (mesh-group | standard);

Hierarchy Level [edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure groups of peers in a full mesh topology to limit excessive flooding of source-active messages to neighboring peers. The default flooding mode is **standard**.

Default If you do not include this statement, default flooding is applied.

Options **mesh-group**—Group of peers that are mesh group members.

standard—Use standard MSDP source-active flooding rules.

Default: standard

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

mode (Protocols PIM)

Syntax mode (bidirectional-sparse | bidirectional-sparse-dense | dense | sparse | sparse-dense);

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
bidirectional-sparse and **bidirectional-sparse-dense** options introduced in Junos OS Release 12.1.

Description Configure the PIM mode on the interface.

Options The choice of PIM mode is closely tied to controlling how groups are mapped to PIM modes, as follows:

- **bidirectional-sparse**—Use if all multicast groups are operating in bidirectional, sparse, or SSM mode.
- **bidirectional-sparse-dense**—Use if multicast groups, except those that are specified in the **dense-groups** statement, are operating in bidirectional, sparse, or SSM mode.
- **dense**—Use if all multicast groups are operating in dense mode.
- **sparse**—Use if all multicast groups are operating in sparse mode or SSM mode.
- **sparse-dense**—Use if multicast groups, except those that are specified in the **dense-groups** statement, are operating in sparse mode or SSM mode.

Default: Sparse mode

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Dense Mode Properties on page 205
- Configuring PIM Sparse-Dense Mode Properties on page 207
- Example: Configuring Bidirectional PIM on page 343

mofrr-asm-starg (Multicast-Only Fast Reroute in a PIM Domain)

Syntax mofrr-asm-starg;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit logical-systems *logical-system-name* routing-options multicast stream-protection],
[edit routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit routing-options multicast stream-protection]

Release Information Statement introduced in Junos OS Release 14.1.
Statement introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description Enable **mofrr-asm-starg** to include any-source multicast (ASM) for (*,G) joins in the Multicast-only fast reroute (MoFRR).

NOTE: **mofrr-asm-starg** applies to IP-PIM only. When enabled for group G, *,G will undergo MoFRR as long as there is no S#,G for Group G. In other words, *,G MoFRR will cease and any old states will be torn down when S#,G is created. Note too, that **mofrr-asm-starg** is not supported for mLDP (since mLDP itself does not support *,G).

In a PIM domain with MoFRR enabled, the default for **stream-protection** is S,G routes only.

Context: Multicast-only fast reroute (MoFRR) can be used to reduce traffic loss in a multicast distribution tree in the event of link down. To employ MoFRR, a downstream router is configured with an alternative path back towards the source, over which it receives a backup live stream of the same multicast traffic. That router propagates the same (S,G) join toward both upstream neighbors in order to create duplicate multicast trees. If a failure is detected on the primary tree, the router switches to the backup tree to prevent packet loss.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

mofrr-disjoint-upstream-only (Multicast-Only Fast Reroute in a PIM Domain)

Syntax mofrr-disjoint-upstream-only;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit logical-systems *logical-system-name* routing-options multicast stream-protection],
[edit routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit routing-options multicast stream-protection]

Release Information Statement introduced in Junos OS Release 14.1.
Statement introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description When you configure multicast-only fast reroute (MoFRR) in a PIM domain, allow only a disjoint RPF (an RPF on a separate plane) to be selected as the backup RPF path.

In a multipoint LDP MoFRR domain, the same label is shared between parallel links to the same upstream neighbor. This is not the case in a PIM domain, where each link forms a neighbor. The **mofrr-disjoint-upstream-only** statement does not allow a backup RPF path to be selected if the path goes to the same upstream neighbor as that of the primary RPF path. This ensures that MoFRR is triggered only on a topology that has multiple RPF upstream neighbors.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

mofrr-no-backup-join (Multicast-Only Fast Reroute in a PIM Domain)

Syntax

```
mofrr-no-backup-join;
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit logical-systems *logical-system-name* routing-options multicast stream-protection],
[edit routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit routing-options multicast stream-protection]

Release Information

Statement introduced in Junos OS Release 14.1.
Statement introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description

When you configure multicast-only fast reroute (MoFRR) in a PIM domain, prevent sending join messages on the backup path, but retain all other MoFRR functionality.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

mofrr-primary-path-selection-by-routing (Multicast-Only Fast Reroute)

Syntax mofrr-primary-path-selection-by-routing;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit logical-systems *logical-system-name* routing-options multicast stream-protection],
[edit routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit routing-options multicast stream-protection]

Release Information Statement introduced in Junos OS Release 14.1.
Statement introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description MoFRR is supported on both equal-cost multipath (ECMP) paths and non-ECMP paths. Unicast loop-free alternate (LFA) routes need to be enabled to support MoFRR on non-ECMP paths. LFA routes are enabled with the link-protection statement in the interior gateway protocol (IGP) configuration. When you enable link protection on an OSPF or IS-IS interface, Junos OS creates a backup LFA path to the primary next hop for all destination routes that traverse the protected interface.

In the context of load balancing, MoFRR prioritizes the disjoint backup in favor of load balancing the available paths.

For Junos OS releases before 15.1R7, for both ECMP and Non-ECMP scenarios, the default MoFRR behavior was *sticky* , that is, if the Active link went down, the Active Path selection would give preference to Backup Path for the transition. The Active Path would not follow the unicast selected gateway

Starting in Junos OS Release 15.1R7 however, the default behavior for non-EMCP scenarios is to be *non*sticky, that is, the selection of Active Path strictly follows unicast selected gateway. MoFRR no longer chooses a unicast LFA path to become the MoFRR Active path; only a unicast LFA path can be selected to become MoFRR Backup.

Default By default, the backup path gets promoted to be the primary path when MoFRR is configured in a PIM domain.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

mpls-internet-multicast

Syntax mpls-internet-multicast;

Hierarchy Level [edit routing-instances *routing-instance-name* instance-type]
[edit protocols pim]

Release Information Statement introduced in Junos OS Release 11.1.

Description A nonforwarding routing instance type that supports Internet multicast over an MPLS network for the default master instance. No interfaces can be configured for it. Only one **mpls-internet-multicast** instance can be configured for each logical system.

The **mpls-internet-multicast** configuration statement is also explicitly required under PIM in the master instance.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Ingress Replication for IP Multicast Using MBGP MVPNs on page 591
- ingress-replication on page 1096

msdp

Syntax

```
msdp {
    disable;
    active-source-limit {
        log-interval seconds;
        log-warning value;
        maximum number;
        threshold number;
    }
    data-encapsulation (disable | enable);
    export [ policy-names ];
    group group-name {
        ... group-configuration ...
    }
    hold-time seconds;
    import [ policy-names ];
    local-address address;
    keep-alive seconds;
    peer address {
        ... peer-configuration ...
    }
    rib-group group-name;
    source ip-prefix</prefix-length> {
        active-source-limit {
            maximum number;
            threshold number;
        }
    }
    sa-hold-time seconds;
    traceoptions {
        file filename <files number> <size size> <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    group group-name {
        disable;
        export [ policy-names ];
        import [ policy-names ];
        local-address address;
        mode (mesh-group | standard);
        peer address {
            ... same statements as at the [edit protocols msdp peer address] hierarchy level shown
                just following ...
        }
        traceoptions {
            file filename <files number> <size size> <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
    peer address {
        disable;
        active-source-limit {
            maximum number;
            threshold number;
```

```
    }
    authentication-key peer-key;
    default-peer;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
      file filename <files number> <size size> <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.4 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description

Enable MSDP on the router or switch. You must also configure at least one peer for MSDP to function.

Default

MSDP is disabled on the router or switch.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP in a Routing Instance on page 396

multicast (Dynamic Profiles Routing Options)

Syntax

```
multicast {
  asm-override-ssm;
  backup-pe-group group-name {
    backups [ addresses ];
    local-address address;
  }
  flow-map flow-map-name {
    bandwidth (bps | adaptive);
    forwarding-cache {
      timeout (never non-discard-entry-only | minutes);
    }
    policy [ policy-names ];
    redundant-sources [ addresses ];
  }
  forwarding-cache {
    threshold suppress value <reuse value>;
    timeout minutes;
  }
  interface interface-name {
    maximum-bandwidth bps;
    no-qos-adjust;
    reverse-oif-mapping {
      no-qos-adjust;
    }
    subscriber-leave-timer seconds;
  }
  pim-to-igmp-proxy {
    upstream-interface [ interface-names ];
  }
  pim-to-mld-proxy {
    upstream-interface [ interface-names ];
  }
  rpf-check-policy [ policy-names ];
  scope scope-name {
    interface [ interface-names ];
    prefix destination-prefix;
  }
  scope-policy [ policy-names ];
  ssm-groups [ addresses ];
  ssm-map ssm-map-name {
    policy [ policy-names ];
    source [ addresses ];
  }
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <disable>;
  }
}
```

Hierarchy Level

[edit dynamic-profiles *profile-name* routing-options],
[edit dynamic-profiles *profile-name* routing-instances *routing-instance-name* routing-options],

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options],
[edit logical-systems *logical-system-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-options]

NOTE: You cannot apply a scope policy to a specific routing instance. That is, all scoping policies are applied to all routing instances. However, the **scope** statement does apply individually to a specific routing instance.

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
interface and **maximum-bandwidth** statements introduced in Junos OS Release 8.3.
interface and **maximum-bandwidth** statements introduced in Junos OS Release 9.0 for EX Series switches.
Statement added to **[edit dynamic-profiles routing-options]** and **[edit dynamic-profiles *profile-name* routing-instances *routing-instance-name* routing-options]** hierarchy levels in Junos OS Release 9.6.

Description

Configure multicast routing options properties.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring the Multicast Forwarding Cache on page 929
- Example: Configuring a Multicast Flow Map on page 931
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316

multicast (Virtual Tunnel in Routing Instances)

Syntax multicast;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* interface vt-*fpc/pic/port.unit-number*],
[edit routing-instances *routing-instance-name* interface vt-*fpc/pic/port.unit-number*]

Release Information Statement introduced in Junos OS Release 9.4.

Description In a multiprotocol BGP (MBGP) multicast VPN (MVPN), configure the virtual tunnel (VT) interface to be used for multicast traffic only.

Default If you omit this statement, the VT interface can be used for both multicast and unicast traffic.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 744
- Example: Configuring MBGP MVPN Extranets on page 669

multicast-replication

Syntax

```
multicast-replication {
  ingress;
  local-latency-fairness;
}
```

Hierarchy Level [edit forwarding-options]

Release Information Statement introduced in Junos OS Release 15.1 for MX Series routers.

Description Configure the mode of multicast replication that helps to optimize multicast latency.

NOTE: The multicast-replication statement is supported only on platforms with the enhanced-ip mode enabled.

Default This statement is disabled by default.

Options

ingress—Complete ingress replication of the multicast data packets where all the egress Packet Forwarding Engines receive packets from the ingress Packet Forwarding Engines directly.

local-latency-fairness—Complete parallel replication of the multicast data packets.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- *forwarding-options*

multicast-router-interface (IGMP Snooping)

Syntax multicast-router-interface;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit protocols igmp-snooping vlan (all | *vlan-name*) interface (all | *interface-name*)],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping interface *interface-name*]
[edit protocols igmp-snooping vlan *vlan-name* interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Statically configure the interface as an IGMP snooping multicast-router interface—that is, an interface that faces toward a multicast router or other IGMP querier.

NOTE: If the specified interface is a trunk port, the interface becomes a multicast-routing device interface for all VLANs configured on the trunk port. In addition, all unregistered multicast packets, whether they are IPv4 or IPv6 packets, are forwarded to the multicast routing device interface, even if the interface is configured as a multicast routing device interface only for IGMP snooping.

Configure an interface as a bridge interface toward other multicast routing devices.

Default Disabled. If this statement is disabled, the interface drops IGMP messages it receives.

The interface can either be a host-side or multicast-routing device interface.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring IGMP Snooping on page 106
- *IGMP Snooping in MC-LAG Active-Active Mode*
- host-only-interface on page 1070

multicast-router-interface (MLD Snooping)

Syntax multicast-router-interface;

Hierarchy Level
[edit protocols mld-snooping vlan (all | *vlan-name*) interface (all | *interface-name*)]
[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Support at the **[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name* interface *interface-name*]** hierarchy level introduced in Junos OS Release 13.3 for EX Series switches.

Description Statically configure the interface as a multicast-router interface—that is, an interface that faces towards a multicast router or other MLD querier.

NOTE: If the specified interface is a trunk port, the interface becomes a multicast-router interface for all VLANs configured on the trunk port. In addition, all unregistered multicast packets, whether they are IPv4 or IPv6 packets, are forwarded to the multicast router interface, even if the interface is configured as a multicast-router interface only for MLD snooping.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

multicast-snooping-options

Syntax

```
multicast-snooping-options {
  flood-groups [ ip-addresses ];
  forwarding-cache {
    threshold suppress value <reuse value>;
  }
  host-outbound-traffic (Multicast Snooping) {
    forwarding-class class-name;
    dot1p number;
  }
  graceful-restart <restart-duration seconds>;
  ignore-stp-topology-change;
  multichassis-lag-replicate-state;
  nexthop-hold-time milliseconds;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*],

Release Information Statement introduced in Junos OS Release 8.5.

Description Establish multicast snooping option values.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Multicast Snooping on page 873
- Enabling Bulk Updates for Multicast Snooping on page 879
- Example: Configuring Multicast Snooping on page 874

multichassis-lag-replicate-state

Syntax multichassis-lag-replicate-state;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options],
[edit routing-instances *routing-instance-name* multicast-snooping-options]

Release Information Statement introduced in Junos OS Release 10.2.

Description Provide multicast snooping for multichassis link aggregation group interfaces. Replicate IGMP join and leave messages from the active link to the standby link of a dual-link multichassis link aggregation group interface, enabling faster recovery of membership information after failover.

Default If not included, membership information is recovered using a standard IGMP network query.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Multicast Snooping on page 873
- multicast-snooping-options on page 1174

multiplier

Syntax multiplier *number*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the number of hello packets not received by a neighbor that causes the originating interface to be declared down.

Options *number*—Number of hello packets.

Range: 1 through 255

Default: 3

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194

mvpn (Draft-Rosen MVPN)

Syntax

```
mvpn {
  family {
    inet {
      autodiscovery {
        inet-mdt;
        }
      disable
    }
    inet6 {
      disable
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced in Junos OS Release 9.4.
The autodiscovery statement was moved from **[.. protocols pim mvpn]** to **[..protocols pim mvpn family inet]** in Junos OS Release 13.3.

Description

Configure the control plane to be used for PE routers in the VPN to discover one another automatically. From here, you can also disable IPv6 draft-rosen multicast VPN at this hierarchy by using the **disable** command at the **protocols pim mvpn family inet6** hierarchy.

Options

The other statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

mvpn

Syntax

```
mvpn {
  inter-region-template{
    template template-name {
      all-regions {
        incoming;
        ingress-replication {
          create-new-ucast-tunnel;
          label-switched-path {
            label-switched-path-template (Multicast) {
              (default-template | lsp-template-name);
            }
          }
        }
        ldp-p2mp;
        rsvp-te {
          label-switched-path-template (Multicast) {
            (default-template | lsp-template-name);
          }
        static-lsp static-lsp;
      region region-name{
        incoming;
        ingress-replication {
          create-new-ucast-tunnel;
          label-switched-path {
            label-switched-path-template (Multicast){
              (default-template | lsp-template-name);
            }
          }
        }
        ldp-p2mp;
        rsvp-te {
          label-switched-path-template (Multicast) {
            (default-template | lsp-template-name);
          }
          static-lsp static-lsp;
        }
      }
    }
  }
mvpn-mode (rpt-spt | spt-only);
receiver-site;
sender-site;
route-target {
  export-target {
    target target-community;
    unicast;
  }
  import-target {
    target {
      target-value;
      receiver target-value;
      sender target-value;
```

```
        }
        unicast {
          receiver;
          sender;
        }
      }
    }
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced in Junos OS Release 8.4.
Support for the **traceoptions** statement at the **[edit protocols mvpn]** hierarchy level introduced in Junos OS Release 13.3.
Support for the **inter-region-template** statement at the **[edit protocols mvpn]** hierarchy level introduced in Junos OS Release 15.1.

Description

Enable next-generation multicast VPNs in a routing instance.

Options

receiver-site—Allow sites with multicast receivers.

sender-site—Allow sites with multicast senders.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Routing Instances for an MBGP MVPN*

mvpn-iana-rt-import

Syntax mvpn-iana-rt-import;

Hierarchy Level [edit logical-systems *logical-system-name* protocols bgp group *group-name*],
[edit protocols bgp group *group-name*]

Release Information Statement introduced in Junos OS release 10.4R2.
Statement deprecated in Junos OS release 17.3, which means it no longer appears in the CLI but can be accessed by scripts or by typing the command name until it is finally removed.

Description Enables the use of IANA assigned rt-import type values (0x010b) for mutlicast VPNs. You can configure this statement on ingress PE routers only.

NOTE: If you configure the **mvpn-iana-rt-import** statement in Junos OS release 10.4R2 and later, the Juniper Networks router can inter-operate with other vendors routers for multicast VPNs. However, the Juniper Networks router cannot inter-operate with Juniper Networks routers running Junos OS release 10.4R1 and earlier.

If you do not configure the **mvpn-iana-rt-import** statement in Junos OS release 10.4R2 and later, the Juniper Networks router cannot inter-operate with other vendors routers for multicast VPNs. However, the Juniper Networks router can inter-operate with Juniper Networks routers running Junos OS release 10.4R1 and earlier.

Default The default rt-import type value is 0x010a.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

-

mvpn (NG-MVPN)

Syntax

```
mvpn {
  autodiscovery-only {
    intra-as {
      inclusive;
    }
  }
  receiver-site;
  route-target {
    export-target {
      target target-community;
      unicast;
    }
    import-target {
      target {
        target <target:number:number> <receiver | sender>;
        unicast <receiver | sender>;
      }
      unicast {
        receiver;
        sender;
      }
    }
  }
  sender-site;
  traceoptions {
    file filename <files number> <size maximum-file-size> <world-readable |
      no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
  unicast-umh-election;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced in Junos OS Release 9.4.

Description

Enable the MVPN control plane for autodiscovery only.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

mvpn-mode

Syntax mvpn-mode (rpt-spt | spt-only);

Hierarchy Level [edit logical-systems *profile-name* routing-instances *instance-name* protocols mvpn],
[edit routing-instances *instance-name* protocols mvpn]

Release Information Statement introduced in Junos OS Release 10.0.

Description Configure the mode for customer PIM (C-PIM) join messages. Mixing MVPN modes within the same VPN is not supported. For example, you cannot have spt-only mode on a source PE and rpt-spt mode on the receiver PE.

Default **spt-only**

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Shared-Tree Data Distribution Across Provider Cores for Providers of MBGP MVPNs*
- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*

neighbor-policy

Syntax neighbor-policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply a PIM interface-level policy to filter neighbor IP addresses.

Options ***policy-name***—Name of the policy that filters neighbor IP addresses.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Interface-Level PIM Neighbor Policies on page 270

nexthop-hold-time

Syntax nexthop-hold-time *milliseconds*;

Hierarchy Level [edit routing-instances *routing-instance-name* multicast-snooping-options]

Release Information Statement introduced in Junos OS Release 10.1.

Description Accumulate outgoing interface changes in order to perform bulk updates to the forwarding table and the routing table. Delete the statement to turn off bulk updates.

Options **milliseconds**—Set the hold time duration from 1 through 1000 milliseconds.
Range: 1 through 1000 milliseconds.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Enabling Bulk Updates for Multicast Snooping on page 879

next-hop (PIM RPF Selection)

Syntax next-hop *next-hop-address*;

Hierarchy Level
[edit routing-instances *routing-instance-name* protocols pim rpf-selection group *group-address* source *source-address*],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection group *group-address* wildcard-source],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection prefix-list *prefix-list-addresses* source *source-address*],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection prefix-list *prefix-list-addresses* wildcard-source]

Release Information
Statement introduced in JUNOS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the specific next-hop address for the PIM group source.

Options ***next-hop-address***—Specific next-hop address for the PIM group source.

Required Privilege Level
view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation
- Example: Configuring PIM RPF Selection on page 816

no-adaptation (PIM BFD Liveness Detection)

Syntax no-adaptation;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information Statement introduced in Junos OS Release 9.0
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure BFD sessions not to adapt to changing network conditions. We recommend that you *do not* disable BFD adaptation unless it is preferable to have BFD adaptation disabled in your network.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- bfd-liveness-detection on page 984

no-bidirectional-mode

Syntax no-birectional-mode;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim graceful-restart],
[edit protocols pim graceful-restart],
[edit routing-instances *routing-instance-name* protocols pim graceful-restart]

Release Information Statement introduced in Junos OS Release 12.1.

Description Disable forwarding for bidirectional PIM routes during graceful restart recovery, both in cases of a routing protocol process (rpd) restart and graceful Routing Engine switchover.

Bidirectional PIM accepts packets for a bidirectional route on multiple interfaces. This means that some topologies might develop multicast routing loops if all PIM neighbors are not synchronized with regard to the identity of the designated forwarder (DF) on each link. If one router is forwarding without actively participating in DF elections, particularly after unicast routing changes, multicast routing loops might occur.

If graceful restart for PIM is enabled and the forwarding of packets on bidirectional routes is disallowed (by including the **no-bidirectional-mode** statement in the configuration), PIM behaves conservatively to avoid multicast routing loops during the recovery period. When the routing protocol process (rpd) restarts, all bidirectional routes are deleted. After graceful restart has completed, the routes are re-added, based on the converged unicast and bidirectional PIM state. While graceful restart is active, bidirectional multicast flows drop packets.

Default If graceful restart for PIM is enabled and the bidirectional PIM is enabled, the default graceful restart behavior is to continue forwarding packets on bidirectional routes. If the gracefully restarting router was serving as a DF for some interfaces to rendezvous points, the restarting router sends a DF Winner message with a metric of 0 on each of these RP interfaces. This ensures that a neighbor router does not become the DF due to unicast topology changes that might occur during the graceful restart period. Sending a DF Winner message with a metric of 0 prevents another PIM neighbor from assuming the DF role until after graceful restart completes. When graceful restart completes, the gracefully restarted router sends another DF Winner message with the actual converged unicast metric.

NOTE: Graceful Routing Engine switchover operates independently of the graceful restart behavior. If graceful Routing Engine switchover is configured without graceful restart, all PIM routes for all modes are deleted when the rpd process restarts. If graceful Routing Engine switchover is configured with graceful restart, the behavior is the same as described here, except that the recovery happens on the Routing Engine that assumes mastership.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring PIM Sparse Mode Graceful Restart on page 383
- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

no-dr-flood (PIM Snooping)

Syntax
no-dr-flood;

Hierarchy Level
[edit routing-instances *<instance-name>* protocols pim-snooping traceoptions],
[edit logical-systems *<logical-system-name>* routing-instances *<instance-name>* protocols pim-snooping traceoptions],
[edit routing-instances *<instance-name>* protocols pim-snooping vlan *<vlan-id>*],
[edit logical-systems *<logical-system-name>* routing-instances *<instance-name>* protocols pim-snooping vlan*<vlan-id>*]

Release Information
Statement introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge Routers.
Statement introduced in Junos OS Release 13.2 for M Series Multiservice Edge Routers.

Description
Disable default flooding of multicast data on the PIM designated router port.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-qos-adjust

Syntax no-qos-adjust;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast interface *interface-name* reverse-oif-mapping],
[edit logical-systems *logical-system-name* routing-options multicast interface *interface-name*],
[edit logical-systems *logical-system-name* routing-options multicast interface *interface-name* reverse-oif-mapping],
[edit routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit routing-instances *routing-instance-name* routing-options multicast interface *interface-name* reverse-oif-mapping],
[edit routing-options multicast interface *interface-name*],
[edit routing-options multicast interface *interface-name* reverse-oif-mapping]

Release Information
Statement introduced in Junos OS Release 9.5.
Statement introduced in Junos OS Release 9.5 for EX Series switches.
Statement added to **[edit routing-instances *routing-instance-name* routing-options multicast interface *interface-name*]**, **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast interface *interface-name*]**, and **[edit routing-options multicast interface *interface-name*]** hierarchy levels in Junos OS Release 9.6.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Disable hierarchical bandwidth adjustment for all subscriber interfaces that are identified by their MLD or IGMP request from a specific multicast interface.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Multicast with Subscriber VLANs on page 912

offer-period

Syntax offer-period *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim interface *interface-name* bidirectional df-election],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional df-election],
[edit protocols pim interface *interface-name* bidirectional df-election],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional df-election]

Release Information
Statement introduced in Junos OS Release 12.1.
Statement introduced in Junos OS Release 13.3 for the PTX5000 router.

Description
Configure the designated forwarder (DF) election offer period for bidirectional PIM. When a DF election Offer or Winner message fails to be received, the message is retransmitted. The **offer-period** statement modifies the interval between repeated DF election messages. The robustness-count statement determines the minimum number of DF election messages that must fail to be received for DF election to fail. To prevent routing loops, all routing devices on the link must have a consistent view of the DF. When the DF election fails because DF election messages are not received, forwarding on bidirectional PIM routes is suspended.

If a router receives from a neighbor a better offer than its own, the router stops participating in the election for a period of robustness-count * **offer-period**. Eventually, all routers except the best candidate stop sending Offer messages.

Options
milliseconds—Interval to wait before retransmitting DF Offer and Winner messages.
Range: 100 through 10,000 milliseconds
Default: 100

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343
- robustness-count on page 1260

oif-map (IGMP Interface)

Syntax oif-map *map-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*], [edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Associates an outgoing interface (OIF) map to the IGMP interface. The OIF map is a routing policy statement that can contain multiple terms.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Multicast with Subscriber VLANs on page 912

oif-map (MLD Interface)

Syntax oif-map *map-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*], [edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.

Description Associate an outgoing interface (OIF) map to an MLD logical interface. The OIF map is a routing policy statement that can contain multiple terms.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Multicast with Subscriber VLANs on page 912

override (PIM Static RP)

Syntax override;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp local],
[edit logical-systems *logical-system-name* protocols pim rp local family inet],
[edit logical-systems *logical-system-name* protocols pim rp local family inet6],
[edit logical-systems *logical-system-name* protocols pim rp static address *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp local],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp local family inet],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp local family inet6],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp static address *address*],
[edit protocols pim rp local],
[edit protocols pim rp local family inet],
[edit protocols pim rp local family inet6],
[edit protocols pim rp static address *address*],
[edit routing-instances *instance-name* protocols pim rp local],
[edit routing-instances *instance-name* protocols pim rp local family inet],
[edit routing-instances *instance-name* protocols pim rp local family inet6],
[edit routing-instances *instance-name* protocols pim rp static address *address*]

Release Information Statement introduced in Junos OS Release 11.4.

Description When you configure both static RP mapping and dynamic RP mapping (such as auto-RP) in a single routing instance, allow the static mapping to take precedence for a given group range, and allow dynamic RP mapping for all other groups.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Static RP on page 237
- Configuring PIM Auto-RP on page 258

override-interval

Syntax override-interval *milliseconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim]
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 10.1.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Set the maximum time in milliseconds to delay sending override join messages for a multicast network that has join suppression enabled. When a router or switch sees a prune message for a join it is currently suppressing, it waits for the interval specified by the override timer before it sends an override join message.

Options This is a random timer with a value in milliseconds.

Range: 0 through maximum override value

Default: 2000 milliseconds

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Enabling Join Suppression on page 222
- propagation-delay on page 1221
- reset-tracking-bit on page 1247

p2mp (Protocols LDP)

Syntax

```
p2mp{
  root-address root-address{
  lsp-id id;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols ldp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols ldp],
[edit protocols ldp],
[edit routing-instances *routing-instance-name* protocols ldp]

Release Information

Statement introduced in Junos OS Release 11.2.

Description

Enable point-to-multipoint MPLS LSPs in an LDP-signaled LSP.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Point-to-Multipoint LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs on page 586
- *Point-to-Multipoint LSPs Overview*

passive (IGMP)

Syntax passive <allow-receive> <send-general-query> <send-group-query>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*], [edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.
allow-receive, **send-general-query**, and **send-group-query** options were added in Junos OS Release 10.0.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description When configured for passive IGMP mode, the interface listens for IGMP reports but it will not send or receive IGMP control traffic such as IGMP reports, queries and leaves. You can, however, configure exceptions to allow the interface to receive certain control traffic or queries.

NOTE: When an interface is configured for IGMP passive mode, Junos no longer processes static IGMP group membership on the interface.

Options You can selectively activate up to two out of the three available options for the **passive** statement while keeping the other functions passive (inactive). Activating all three options would be equivalent to not using the **passive** statement.

allow-receive—Enables IGMP to receive control traffic on the interface.

send-general-query—Enables IGMP to send general queries on the interface.

send-group-query—Enables IGMP to send group-specific and group-source-specific queries on the interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast with Subscriber VLANs on page 912
- Enabling IGMP on page 28

passive (MLD)

Syntax passive <allow-receive> <send-general-query> <send-group-query>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.6.
allow-receive, **send-general-query**, and **send-group-query** options added in Junos OS Release 10.0.

Description Specify that MLD run on the interface and either not send and receive control traffic or selectively send and receive control traffic such as MLD reports, queries, and leaves.

NOTE: You can selectively activate up to two out of the three available options for the passive statement while keeping the other functions passive (inactive). Activating all three options is equivalent to not using the passive statement.

Options **allow-receive**—Enables MLD to receive control traffic on the interface.

send-general-query—Enables MLD to send general queries on the interface.

send-group-query—Enables MLD to send group-specific and group-source-specific queries on the interface.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Multicast with Subscriber VLANs on page 912

peer (Protocols MSDP)

Syntax

```
peer address {
  disable;
  active-source-limit {
    maximum number;
    threshold number;
  }
  authentication-key peer-key;
  default-peer;
  export [ policy-names ];
  import [ policy-names ];
  local-address address;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Define an MSDP peering relationship. An MSDP routing device must know which routing devices are its peers. You define the peer relationships explicitly by configuring the neighboring routing devices that are the MSDP peers of the local routing device. After peer relationships are established, the MSDP peers exchange messages to advertise active multicast sources. To configure multiple MSDP peers, include multiple **peer** statements.

By default, the peer's options are identical to the global or group-level MSDP options. To override the global or group-level options, include peer-specific options within the peer (Protocols MSDP) statement.

At least one peer must be configured for MSDP to function. You must configure ***address*** and **local-address**.

Options

address—Name of the MSDP peer.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396

pim

Syntax

```
pim {
    disable;
    assert-timeout seconds;
    dense-groups {
        addresses;
    }
    dr-election-on-p2p;
    export;
    family (inet | inet6) {
        disable;
    }
    graceful-restart {
        disable;
        no-bidirectional-mode;
        restart-duration seconds;
    }
    import [ policy-names ];
    interface interface-name {
        family (inet | inet6) {
            disable;
            }
            bfd-liveness-detection {
                authentication {
                    algorithm algorithm-name;
                    key-chain key-chain-name;
                loose-check;
                detection-time {
                    threshold milliseconds;
                }
            minimum-interval milliseconds;
                minimum-receive-interval milliseconds;
                multiplier number;
                no-adaptation;
                transmit-interval {
                    minimum-interval milliseconds;
                    threshold milliseconds;
                }
            version (0 | 1 | automatic);
            }
            accept-remote-source;
            disable;
            bidirectional {
                df-election {
                    backoff-period milliseconds;
                    offer-period milliseconds;
                    robustness-count number;
                }
            }
            family (inet | inet6) {
                disable;
            }
            hello-interval seconds;
```

```
    mode (bidirectional-sparse | bidirectional-sparse-dense | dense | sparse |
      sparse-dense);
    neighbor-policy [ policy-names ];
    override-interval milliseconds;
    priority number;
    propagation-delay milliseconds;
    reset-tracking-bit;
    version version;
  }
  join-load-balance;
  join-prune-timeout;
  mdt {
    data-mdt-reuse;
    group-range multicast-prefix;
    threshold {
      group group-address {
        source source-address {
          rate threshold-rate;
        }
      }
      tunnel-limit limit;
    }
  }
  mvpn {
    autodiscovery {
      inet-mdt;
    }
  }
  nonstop-routing;
  override-interval milliseconds;
  propagation-delay milliseconds;
  reset-tracking-bit;
  rib-group group-name;
  rp {
    auto-rp {
      (announce | discovery | mapping);
      (mapping-agent-election | no-mapping-agent-election);
    }
    bidirectional {
      address address {
        group-ranges {
          destination-ip-prefix</prefix-length>;
        }
        hold-time seconds;
        priority number;
        }
      }
    bootstrap {
      family (inet | inet6) {
        export [ policy-names ];
        import [ policy-names ];
        priority number;
      }
    }
    bootstrap-import [ policy-names ];
    bootstrap-export [ policy-names ];
```

```
bootstrap-priority number;
dr-register-policy [ policy-names ];
embedded-rp {
  group-ranges {
    destination-ip-prefix</prefix-length>;
  }
  maximum-rps limit;
}
group-rp-mapping {
  family (inet | inet6) {
    log-interval seconds;
    maximum limit;
    threshold value;
    }
  }
  log-interval seconds;
  maximum limit;
  threshold value;
  }
}
local {
  family (inet | inet6) {
    address address;
    anycast-pim {
      rp-set {
        address address <forward-msdp-sa>;
      }
      disable;
      local-address address;
    }
    group-ranges {
      destination-ip-prefix</prefix-length>;
    }
    hold-time seconds;
    override;
    priority number;
  }
}
register-limit {
  family (inet | inet6) {
    log-interval seconds;
    maximum limit;
    threshold value;
    }
  }
  log-interval seconds;
  maximum limit;
  threshold value;
  }
}
rp-register-policy [ policy-names ];
spt-threshold {
  infinity [ policy-names ];
}
static {
  address address {
```

```
            override;
            version version;
            group-ranges {
              destination-ip-prefix</prefix-length>;
            }
          }
        }
      }
      rpf-selection {
        group group-address{
          sourcesource-address{
            next-hop next-hop-address;
          }
          wildcard-source {
            next-hop next-hop-address;
          }
        }
        prefix-list prefix-list-addresses {
          source source-address {
            next-hop next-hop-address;
          }
          wildcard-source {
            next-hop next-hop-address;
          }
        }
        sglimit {
          family (inet | inet6) {
            log-interval seconds;
            maximum limit;
            threshold value;
            }
          }
          log-interval seconds;
          maximum limit;
          threshold value;
          }
        }
        traceoptions {
          file filename <files number> <size size> <world-readable | no-world-readable>;
          flag flag <flag-modifier> <disable>;
        }
        tunnel-devices [ mt-fpc/pic/port ];
      }
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols],
[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information

Statement introduced before Junos OS Release 7.4.
family statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

Description Enable PIM on the routing device.

The remaining statements are explained separately. See CLI Explorer.

Default PIM is disabled on the routing device.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512
- Configuring PIM Dense Mode Properties on page 205
- Configuring PIM Sparse-Dense Mode Properties on page 207

pim-asm

Syntax

```
pim-asm {
  group-address (Routing Instances) address;
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel]

Release Information Statement introduced in Junos OS Release 8.3.

Description Specify a Protocol Independent Multicast (PIM) sparse mode provider tunnel for an MBGP MVPN or for a draft-rosen MVPN.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

pim-snooping

Syntax

```
pim-snooping {
  no-dr-flood;
    traceoptions{
      file [filename  files | no-word-readable | size | word-readable];
      flag [all | general | hello | join | normal | packets | policy | prune | route | state | task |
        timer];
    }
  vlan<vlan-id>{
    no-dr-flood;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *instance-name* instance-type *vpls* protocols],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols],
[edit routing-instances *instance-name* protocols]

Release Information

Statement introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge Routers.
Statement introduced in Junos OS Release 13.2 for M Series Multiservice Edge Routers.

Description

PIM snooping snoops PIM hello and join/prune packets on each interface to find interested multicast receivers and then populates the multicast forwarding tree with the information. PIM snooping is configured on PE routers connected using pseudowires and ensures that no new PIM packets are generated in the VPLS (with the exception of PIM messages sent through LDP on pseudowires). PIM snooping differs from PIM proxying in that PIM snooping floods both the PIM hello and join/prune packets in the VPLS, whereas PIM proxying only floods hello packets.

Default

PIM snooping is disabled on the device.

Options

no-dr-flood— Disable default flooding of multicast data on the PIM-designated router port.

traceoptions—Configure tracing options for PIM snooping.

vlan *<vlan-id>*—Configure PIM snooping parameters for a VLAN.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- PIM Snooping for VPLS on page 886

pim-ssm (Provider Tunnel)

Syntax

```
pim-ssm {
  group-address (Routing Instances) address;
  tunnel-source address;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*]

Release Information

Statement introduced in Junos OS Release 9.4.
In Junos OS Release 17.3R1, the **pim-ssm** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default multicast distribution tree (MDT) in Rosen 7, and data MDT for Rosen 6 and Rosen 7.

Description

Configure the PIM source-specific multicast (SSM) provider tunnel. Use **family inet6 pim-ssm** for Rosen 7 running on IPv6 . For Rosen 7 on IPv4, use **family inet pim-ssm**. The customer data-MDT can be configured on IPv4 or IPv6, but not both (the provider space always runs on IPv4). Enable Rosen IPv4 before enabling Rosen IPv6.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

pim-ssm (Selective Tunnel)

Syntax

```
pim-ssm {
    group-range multicast-prefix;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *group-address* source *source-address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *group-address* wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *group-address* source *source-address*],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *group-address* wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source]

Release Information

Statement introduced in Junos OS Release 10.1.

Description

Establish the multicast group address range to use for creating MBGP MVPN source-specific multicast selective PMSI tunnels.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

pim-to-igmp-proxy

Syntax

```
pim-to-igmp-proxy {
  upstream-interface [ interface-names ];
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Configure the rendezvous point (RP) routing device that resides between a customer edge-facing Protocol Independent Multicast (PIM) domain and a core-facing PIM domain to translate PIM join or prune messages into corresponding Internet Group Management Protocol (IGMP) report or leave messages. The routing device then transmits the report or leave messages by proxying them to one or two upstream interfaces that you configure on the RP routing device. Including the **pim-to-igmp-proxy** statement enables you to use IGMP to forward IPv4 multicast traffic across the PIM sparse mode domains.

The **pim-to-igmp-proxy** statement is not supported for routing instances.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM-to-IGMP Message Translation on page 386

pim-to-mld-proxy

Syntax

```
pim-to-mld-proxy {
  upstream-interface [ interface-names ];
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description

Configure the rendezvous point (RP) routing device that resides between a customer edge–facing Protocol Independent Multicast (PIM) domain and a core-facing PIM domain to translate PIM join or prune messages into corresponding Multicast Listener Discovery (MLD) report or leave messages. The routing device then transmits the report or leave messages by proxying them to one or two upstream interfaces that you configure on the RP routing device. Including the **pim-to-mld-proxy** statement enables you to use MLD to forward IPv6 multicast traffic across the PIM sparse mode domains.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM-to-MLD Message Translation on page 387

policy (Flow Maps)

Syntax policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit logical-systems *logical-system-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-options multicast flow-map *flow-map-name*]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure a flow map policy.

Options ***policy-names***—Name of one or more policies for flow mapping.

Required Privilege Level routing—To view this statement in the configuration.

policy (Multicast-Only Fast Reroute)

Syntax policy *policy-name*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit logical-systems *logical-system-name* routing-options multicast stream-protection],
[edit routing-instances *routing-instance-name* routing-options multicast stream-protection],
[edit routing-options multicast stream-protection]

Release Information Statement introduced in Junos OS Release 14.1.
Statement introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description When you configure multicast-only fast reroute (MoFRR), apply a routing policy that filters for a restricted set of multicast streams to be affected by your MoFRR configuration. You can apply filters that are based on source or group addresses.

For example:

```
routing-options {
  multicast {
    stream-protection {
      policy mofrr-select;
    }
  }
}
policy-statement mofrr-select {
  term A {
    from {
      source-address-filter 225.1.1.1/32 exact;
    }
    then {
      accept;
    }
  }
  term B {
    from {
      source-address-filter 226.0.0.0/8 orlonger;
    }
    then {
      accept;
    }
  }
  term C {
    from {
      source-address-filter 227.1.1.0/24 orlonger;
      source-address-filter 227.4.1.0/24 orlonger;
      source-address-filter 227.16.1.0/24 orlonger;
    }
    then {
      accept;
    }
  }
```

```
    term D {
      from {
        source-address-filter 227.1.1.1/32 exact;
      }
      then {
        reject; #MoFRR disabled
      }
    }
    term E {
      from {
        route-filter 227.1.1.0/24 orlonger;
      }
      then accept;
    }
    ...
}
```

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

policy (PIM rpf-vector)

Syntax policy [*policy-name*];

Hierarchy Level [edit dynamic-profiles *name* protocols pim rp rpf-vector],
[edit logical-systems *name* protocols pim rprpf-vector],
[edit logical-systems *name* routing-instances *name* protocols pim rp rpf-vector],
[edit protocols pim rp rpf-vector],
[edit routing-instances *name* protocols pim rp rpf-vector]

Release Information Statement introduced in Junos OS Release 17.3R1.

Description Create a filter policy. The configured device checks the policy configuration to determine whether or not to apply rpf-vector to (S,G).

RPF Vector Policy Example This example policy shows Source and Group, using Source, using Group.

```
policy-statement pim-rpf-vector-example {
        term A {
            from {
               source-address-filter <filter A>;
            }
            then {
               accept;
            }
        }
        term B {
            from {
               source-address-filter <filter A>;
               route-filter <filter D>;
            }
            then {
               p2mp-lsp-root {
                   address root address;
               }
               accept;
            }
        }
        term C {
            from {
               route-filter <filter D>;
            }
            then {
               accept;
            }
        }
        ...
    }
```

RPF Vector Policy Configuration statements This example policy using Source, Group.

```
set protocols pim rpf-vector policy rpf-vector-policy
set policy-options policy-statement rpf-vector-policy term 1 from route-filter
 232.0.0.1/32 exact
```

```
set policy-options policy-statement rpf-vector-policy term 1 from
source-address-filter 22.1.1.2/32 exact
set policy-options policy-statement rpf-vector-policy term 1 then p2mp-lsp-root
 address 200.1.1.2
set policy-options policy-statement rpf-vector-policy term 1 then accept
```

RPF Vector Policy Configuration statements

This example policy using Group, Source wildcard.

```
set protocols pim rpf-vector policy rpf-vector-policy
set policy-options policy-statement rpf-vector-policy term 1 from
source-address-filter 22.1.1.2/32 exact
set policy-options policy-statement rpf-vector-policy term 1 from route-filter
 0.0.0.0/0 longer
set policy-options policy-statement rpf-vector-policy term 1 then p2mp-lsp-root
 address 200.1.1.2
set policy-options policy-statement rpf-vector-policy term 1 then accept
```

Required Privilege Level

routing

Related Documentation

- show pim join on page 1642

policy (SSM Maps)

Syntax

policy [*policy-names*];

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast ssm-map *ssm-map-name*],
[edit logical-systems *logical-system-name* routing-options multicast ssm-map *ssm-map-name*],
[edit routing-instances *routing-instance-name* routing-options multicast ssm-map *ssm-map-name*],
[edit routing-options multicast ssm-map *ssm-map-name*]

Release Information

Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description

Apply one or more policies to an SSM map.

Options

policy-names—Name of one or more policies for SSM mapping.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To view this statement in the configuration.

Related Documentation

- Example: Configuring SSM Mapping on page 322

prefix

Syntax prefix *destination-prefix*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast scope *scope-name*],
[edit logical-systems *logical-system-name* routing-options multicast scope *scope-name*],
[edit routing-instances *routing-instance-name* routing-options multicast scope *scope-name*],
[edit routing-options multicast scope *scope-name*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the prefix for multicast scopes.

Options ***destination-prefix***—Address range for the multicast scope.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring Administrative Scoping on page 899
- Example: Creating a Named Scope for Multicast Scoping on page 901
- multicast on page 1168

prefix-list (PIM RPF Selection)

Syntax

```
prefix-list prefix-list-addresses {
  source source-address {
    next-hop next-hop-address;
  }
  wildcard-source {
    next-hop next-hop-address;
  }
}
```

Hierarchy Level

[edit routing-instances *routing-instance-name* protocols pim rpf-selection group *group-address* source *source-address*],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection group *group-address* wildcard-source],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection prefix-list *prefix-list-addresses* source *source-address*],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection prefix-list *prefix-list-addresses* wildcard-source]

Release Information

Statement introduced in Junos OS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

(Optional) Configure a list of prefixes (addresses) for multiple PIM groups.

Options

prefix-list-addresses—List of prefixes (addresses) for multiple PIM groups.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM RPF Selection on page 816

primary (Virtual Tunnel in Routing Instances)

Syntax primary;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* interface vt-*fpc/pic/port.unit-number*],
[edit routing-instances *routing-instance-name* interface vt-*fpc/pic/port.unit-number*]

Release Information Statement introduced in Junos OS Release 12.3.

Description In a multiprotocol BGP (MBGP) multicast VPN (MVPN), configure the virtual tunnel (VT) interface to be used as the primary interface for multicast traffic.

Junos OS supports up to eight VT interfaces configured for multicast in a routing instance to provide redundancy for MBGP (next-generation) MVPNs. This support is for RSVP point-to-multipoint provider tunnels as well as multicast Label Distribution Protocol (MLDP) provider tunnels. This feature works for extranets as well.

This statement allows you to configure one of the VT interfaces to be the primary interface, which is always used if it is operational. If a VT interface is configured as the primary, it becomes the nexthop that is used for traffic coming in from the core on the label-switched path (LSP) into the routing instance. When a VT interface is configured to be primary and the VT interface is used for both unicast and multicast traffic, only the multicast traffic is affected.

If no VT interface is configured to be the primary or if the primary VT interface is unusable, one of the usable configured VT interfaces is chosen to be the nexthop that is used for traffic coming in from the core on the LSP into the routing instance. If the VT interface in use goes down for any reason, another usable configured VT interface in the routing instance is chosen. When the VT interface in use changes, all multicast routes in the instance also switch their reverse-path forwarding (RPF) interface to the new VT interface to allow the traffic to be received.

To realize the full benefit of redundancy, we recommend that when you configure multiple VT interfaces, at least one of the VT interfaces be on a different Tunnel PIC from the other VT interfaces. However, Junos OS does not enforce this.

Default If you omit this statement, Junos OS chooses a VT interface to be the active interface for multicast traffic.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 744
- Example: Configuring MBGP MVPN Extranets on page 669

primary (MBGP MVPN)

Syntax primary *address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn static-umh],
[edit routing-instances *routing-instance-name* protocols mvpn static-umh]

Release Information Statement introduced in Junos OS Release 15.1.

Description Statically set the primary upstream multicast hop (UMH) for type 7 (S,G) routes.

If the primary UMH is unavailable, the backup UMH is used.

Options ***address***—Address of the primary UMH.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- sender-based-rpf on page 1278
- static-umh (MBGP MVPN) on page 1315
- unicast-umh-election on page 1364

priority (Bootstrap)

Syntax priority *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp bootstrap (inet | inet6)],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp bootstrap (inet | inet6)],
[edit protocols pim rp bootstrap (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap (inet | inet6)]

Release Information
Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the routing device's likelihood to be elected as the bootstrap router.

Options
number—Routing device's priority for becoming the bootstrap router. A higher value corresponds to a higher priority.

Range: 0 through a 32-bit number

Default: 0 (The routing device has the least likelihood of becoming the bootstrap router and sends packets with a priority of 0.)

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring PIM Bootstrap Properties for IPv4 on page 253
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 255
- bootstrap-priority on page 990

priority (PIM Interfaces)

Syntax priority *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the routing device's likelihood to be elected as the designated router.

Options
number—Routing device's priority for becoming the designated router. A higher value corresponds to a higher priority.

Range: 0 through 4294967295

Default: 1 (Each routing device has an equal probability of becoming the DR.)

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Interface Priority for PIM Designated Router Selection on page 302

priority (PIM RPs)

Syntax priority *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp bidirectional address *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim rp bidirectional address *address*],
[edit protocols pim rp bidirectional address *address*],
[edit protocols pim rp local family (inet | inet6)],
[edit routing-instances *instance-name* protocols pim rp bidirectional address *address*],
[edit routing-instances *routing-instance-name* protocols pim rp local family (inet | inet6)]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Support for bidirectional RP addresses introduced in Junos OS Release 12.1.
Statement introduced in Junos OS Release 13.3 for the PTX5000 router.

Description
For PIM-SM, configure this routing device's priority for becoming an RP.

For bidirectional PIM, configure this RP address' priority for becoming an RP.

The bootstrap router uses this field when selecting the list of candidate rendezvous points to send in the bootstrap message. A smaller number increases the likelihood that the routing device or RP address becomes the RP. A priority value of 0 means that bootstrap router can override the group range being advertised by the candidate RP.

Options
number—Priority for becoming an RP. A lower value corresponds to a higher priority.

Range: 0 through 255

Default: 1

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Local PIM RPs on page 237
- Example: Configuring Bidirectional PIM on page 343

promiscuous-mode (Protocols IGMP)

Syntax promiscuous-mode;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*], [edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify that the interface accepts IGMP reports from hosts on any subnetwork. Note that when enabling promiscuous-mode, all routing devices on the ethernet segment must be configured with the promiscuous mode statement. Otherwise, only the interface configured with lowest IPv4 address acts as the querier for IGMP for this Ethernet segment.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Accepting IGMP Messages from Remote Subnetworks on page 33

propagation-delay

Syntax propagation-delay *milliseconds*;

Hierarchy Level
[edit protocols pim],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 10.1.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Set a delay for implementing a PIM prune message on the upstream routing device on a multicast network for which join suppression has been enabled. The routing device waits for the prune pending period to detect whether a join message is currently being suppressed by another routing device.

Options ***milliseconds***—Interval for the prune pending timer, which is the sum of the **propagation-delay** value and the **override-interval** value.

Range: 250 through 2000 milliseconds

Default: 500 milliseconds

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Enabling Join Suppression on page 222
- override-interval on page 1192
- reset-tracking-bit on page 1247

provider-tunnel

Syntax

```
provider-tunnel {
  family {
    inet {
      ingress-replication {
        create-new-ucast-tunnel;
          label-switched-path-template {
            (default-template | lsp-template-name);
          }
        }
      ldp-p2mp;
      mdt {
        data-mdt-reuse;
        group-range multicast-prefix;
        threshold {
          group group-address {
            source source-address {
              rate threshold-rate;
            }
          }
          }
        tunnel-limit limit;
      }
      pim-asm {
        group-address (Routing Instances) address;
      }
      pim-ssm {
        group-address (Routing Instances) address;
      }
      rsvp-te {
        label-switched-path-template {
          (default-template | lsp-template-name);
        }
        static-lsp lsp-name;
      }
    }
  inet6 {
    ingress-replication {
      create-new-ucast-tunnel;
        label-switched-path-template {
          (default-template | lsp-template-name);
        }
      }
    ldp-p2mp;
    mdt {
      data-mdt-reuse;
      group-range multicast-prefix;
      threshold {
        group group-address {
          source source-address {
            rate threshold-rate;
          }
        }
```

```
        }
      tunnel-limit limit;
    }
  }
    pim-asm {
      group-address (Routing Instances) address;
    }
    pim-ssm {
      group-address (Routing Instances) address;
    }
    rsvp-te {
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
      static-lsp lsp-name;
    }
  ingress-replication {
    create-new-ucast-tunnel;
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
    }
  ldp-p2mp;
  pim-asm {
    group-address (Routing Instances) address;
  }
  pim-ssm {
    group-address (Routing Instances) address;
  }
  rsvp-te {
    label-switched-path-template {
      (default-template | lsp-template-name);
    }
    static-lsp lsp-name;
  }
  selective {
    group multicast--prefix/prefix-length {
      source ip--prefix/prefix-length {
        ldp-p2mp;
        create-new-ucast-tunnel;
          label-switched-path-template {
            (default-template | lsp-template-name);
          }
        }
        pim-ssm {
          group-range multicast-prefix;
        }
        rsvp-te {
          label-switched-path-template {
            (default-template | lsp-template-name);
          }
          static-lsp point-to-multipoint-lsp-name;
        }
        threshold-rate kbps;
      }
      wildcard-source {
```

```
            pim-ssm {
              group-range multicast-prefix;
            }
            rsvp-te {
              label-switched-path-template {
                (default-template | lsp-template-name);
              }
              static-lsp point-to-multipoint-lsp-name;
            }
            threshold-rate kbps;
          }
        }
        tunnel-limit number;
        wildcard-group-inet {
          wildcard-source {
            pim-ssm {
              group-range multicast-prefix;
            }
            rsvp-te {
              label-switched-path-template {
                (default-template | lsp-template-name);
              }
              static-lsp lsp-name;
            }
            threshold-rate number;
          }
        }
        wildcard-group-inet6 {
          wildcard-source {
            pim-ssm {
              group-range multicast-prefix;
            }
            rsvp-te {
              label-switched-path-template {
                (default-template | lsp-template-name);
              }
              static-lsp lsp-name;
            }
            threshold-rate number;
          }
        }
      }
    }
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information

Statement introduced in Junos OS Release 8.3.
The **selective** statement and substatements added in Junos OS Release 8.5.
The **ingress-replication** statement and substatements added in Junos OS Release 10.4.
In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description

Configure virtual private LAN service (VPLS) flooding of unknown unicast, broadcast, and multicast traffic using point-to-multipoint LSPs. Also configure point-to-multipoint LSPs for MBGP MVPNs.

Options

The remaining statements are explained separately. .

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Flooding Unknown Traffic Using Point-to-Multipoint LSPs in VPLS*
- *Configuring RSVP-Signaled Inclusive Point-to-Multipoint LSPs for an MBGP MVPN*
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502

proxy

Syntax

```
proxy {
  source-address ip-address;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Configure proxy mode and options, including source address. All the queries generated by IGMP snooping are sent using 0.0.0.0 as the source address in order to avoid participating in IGMP querier election. Also, all reports generated by IGMP snooping are sent with 0.0.0.0 as the source address unless there is a configured source address to use.

Default

By default, IGMP snooping does not employ proxy mode.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

proxy (Multicast VLAN Registration)

Syntax proxy source-address *ip-address*;

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*)]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.

Description Specify that the VLAN operate in proxy mode. The proxy option is supported only for a VLAN acting as a data-forwarding source.

Default Disabled

Options **source-address *ip-address***—IP address of the source VLAN to act as proxy.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

qualified-vlan

Syntax qualified-vlan *vlan-id*;

Hierarchy Level [edit protocols mld-snooping vlan *vlan-name*]
[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name*]
[edit protocols igmp-snooping vlan *vlan-name*]

Release Information Statement introduced in Junos OS Release 13.3 for EX Series switches.
Statement introduced in Junos OS Release 15.1X53-D10 for QFX10000 switches.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description Configure VLAN options for qualified learning.

Options *vlan-id*—VLAN ID of the learning domain.

Range: 0 through 1023

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure) on page 142
- Configuring IGMP Snooping on Switches on page 88
- *show mld snooping membership*
- IGMP Snooping Overview on page 81
- **igmp-snooping on page 1077**

query-interval (Bridge Domains)

Syntax query-interval *seconds*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* protocols mld-snooping] ,
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],[
edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols mld-snooping]
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*]
[edit routing-instances *routing-instance-name* protocols mld-snooping]
[edit protocols igmp-snooping vlan]

Release Information
Statement introduced before Junos OS Release 8.5.
Statement introduced in Junos OS Release 13.2 for the QFX series.
Statement introduced in Junos OS Release 14.2 for MX series Routers with MPC.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description Configure the interval for host-query message timeouts.

Options *seconds*—Time interval. This value must be greater than the interval set for query-response-interval.

Range: 1 through 1024

Default: 125 seconds

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106
- query-last-member-interval (Bridge Domains) on page 1233
- query-response-interval (Bridge Domains) on page 1236
- *mld-snooping*
- igmp-snooping on page 1077
- IGMP Snooping Overview on page 81

query-interval (Protocols IGMP)

Syntax query-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify how often the querier routing device sends general host-query messages.

Options *seconds*—Time interval.

Range: 1 through 1024

Default: 125 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the IGMP Host-Query Message Interval on page 29
- query-last-member-interval (Protocols IGMP) on page 1234
- query-response-interval (Protocols IGMP) on page 1237

query-interval (Protocols IGMP AMT)

Syntax query-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify how often the querier router sends IGMP general host-query messages through an Automatic Multicast Tunneling (AMT) interface.

Options *seconds*—Number of seconds between sending of general host query messages.

Range: 1 through 1024

Default: 125 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Default IGMP Parameters for AMT Interfaces on page 425

query-interval (Protocols MLD)

Syntax query-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before Junos OS Release 7.4.

Description Specify how often the querier router sends general host-query messages.

Options *seconds*—Time interval.

Range: 1 through 1024

Default: 125 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the MLD Host-Query Message Interval on page 56
- query-last-member-interval (Protocols MLD) on page 1235
- query-response-interval (Protocols MLD) on page 1239

query-last-member-interval (Bridge Domains)

Syntax query-last-member-interval *seconds*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* protocols mld-snooping] ,
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols mld-snooping]
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id*
[edit routing-instances *routing-instance-name* protocols mld-snooping]interface *interface-name*]
[edit protocols igmp-snooping vlan],

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 13.2 for the QFX series.
Statement introduced in Junos OS Release 14.2 for MX series Routers with MPC.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description Configure the interval for group-specific query timeouts.

Options
seconds—Time interval, in fractions of a second or seconds.
Range: 0.1 through 0.9, then in 1-second intervals 1 through 1024
Default: 1 second

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring IGMP Snooping on page 106
- query-interval on page 1229
- query-response-interval on page 1236
- *mld-snooping*
- igmp-snooping on page 1077
- Example: Configuring IGMP Snooping on SRX Series Devices on page 114

query-last-member-interval (Protocols IGMP)

Syntax query-last-member-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify how often the querier routing device sends group-specific query messages.

Options ***seconds***—Time interval, in fractions of a second or seconds.

Range: 0.1 through 0.9, then in 1-second intervals 1 through 999999

Default: 1 second

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the IGMP Last-Member Query Interval on page 34
- query-interval (Protocols IGMP) on page 1230
- query-response-interval (Protocols IGMP) on page 1237

query-last-member-interval (Protocols MLD)

Syntax query-last-member-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]
[edit protocols mld-snooping vlan *vlan-id*]
[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Support at the **[edit protocols mld-snooping vlan *vlan-id*]** and the **[edit routing-instances *instance-name*protocols mld-snooping vlan *vlan-id*]** hierarchy levels introduced in Junos OS Release 13.3 for EX Series switches.
Support at the **[edit protocols mld-snooping vlan *vlan-id*]** hierarchy level introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description Specify how often the querier routing device sends group-specific query messages.

Options ***seconds***—Time interval, in fractions of a second or seconds.

Range: 0.1 through 0.9, then in 1-second intervals from 1 through 1024

Default: 1 second

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- **mld-snooping on page 1155**
- Modifying the MLD Last-Member Query Interval on page 58
- **query-interval (Protocols MLD) on page 1232**
- **query-response-interval (Protocols MLD) on page 1239**
- Understanding MLD Snooping on page 125

query-response-interval (Bridge Domains)

Syntax query-response-interval *seconds*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snoopingvlan *vlan-id* interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols mld-snooping] ,
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snoopingvlan *vlan-id*
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols mld-snooping]interface *interface-name*]
[edit routing-instances *routing-instance-name* protocols mld-snooping]
[edit protocols igmp-snooping vlan],

Release Information
Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 13.2 for the QFX series.
Statement introduced in Junos OS Release 14.2 for MX series Routers with MPC.
Statement introduced in Junos OS Release 18.1R1 for SRX1500 devices.

Description Specify how long to wait to receive a response to a specific query message from a host.

Options *seconds*—Time interval. This interval should be less than the host-query interval.

Range: 1 through 1024

Default: 10 seconds

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- Example: Configuring IGMP Snooping on page 106
- query-interval (Bridge Domains) on page 1229
- query-last-member-interval (Bridge Domains) on page 1233
- *mld-snooping*
- igmp-snooping on page 1077

query-response-interval (Protocols IGMP)

Syntax query-response-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify how long the querier routing device waits to receive a response to a host-query message from a host.

Options ***seconds***—The query response interval must be less than the query interval.

Range: 1 through 1024

Default: 10 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the IGMP Query Response Interval on page 30
- query-interval (Protocols IGMP) on page 1230
- query-last-member-interval (Protocols IGMP) on page 1234

query-response-interval (Protocols IGMP AMT)

Syntax query-response-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify how long the IGMP querier router waits to receive a response to a host query message from a host through an Automatic Multicast Tunneling (AMT) interface.

Options *seconds*—Time to wait to receive a response to a host query message. The query response interval must be less than the query interval.

Range: 1 through 1024

Default: 10 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Default IGMP Parameters for AMT Interfaces on page 425

query-response-interval (Protocols MLD)

Syntax query-response-interval *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]
[edit protocols mld-snooping vlan *vlan-id*]
[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-id*]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support at the **[edit protocols mld-snooping vlan *vlan-id*]** and the **[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-id*]** hierarchy levels introduced in Junos OS Release 13.3 for EX Series switches.

Description Specify how long the querier routing device waits to receive a response to a host-query message from a host.

Options *seconds*—Time interval.

Range: 1 through 1024

Default: 10 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the MLD Query Response Interval on page 57
- query-interval (Protocols MLD) on page 1232
- query-last-member-interval (Protocols MLD) on page 1235

rate (Routing Instances)

Syntax rate *threshold-rate*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address* source *source-address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt threshold group *group-address* source *source-address*],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address* source *source-address*],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt threshold group *group-address* source *source-address*]

Release Information Statement introduced before Junos OS Release 7.4. **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description Apply a rate threshold to a multicast source to automatically create a data MDT.

Options ***threshold-rate***—Rate in kilobits per second (Kbps) to apply to source.

Range: 10 Kbps through 1 Gbps (1,000,000 Kbps)

Default: 10 Kbps

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

receiver

Syntax

```
receiver {
  source-vlans vlan-list;
  install;
}
```

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) data-forwarding]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for the QFX Series.

Description Configure a VLAN as a multicast receiver VLAN of the multicast VLAN (MVLAN).

The remaining statements are explained separately. See CLI Explorer.

Default Disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

redundant-sources

Syntax redundant-sources [*addresses*];

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit logical-systems *logical-system-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-options multicast flow-map *flow-map-name*]

Release Information
Statement introduced in Junos OS Release 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure a list of redundant sources for multicast flows defined by a flow map.

Options ***addresses***—List of IPv4 or IPv6 addresses for use as redundant (backup) sources for multicast flows defined by a flow map.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring a Multicast Flow Map on page 931

register-limit

Syntax

```
register-limit {
  family (inet | inet6) {
    log-interval seconds;
    maximum limit;
    threshold value;
  }
  log-interval seconds;
  maximum limit;
  threshold value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced in Junos OS Release 12.2.

Description

Configure a limit for the number of incoming (S,G) PIM registers.

NOTE: The maximum limit settings that you configure with the **maximum** and the **family (inet | inet6) maximum** statements are mutually exclusive. For example, if you configure a global maximum PIM register message limit, you cannot configure a limit at the family level for IPv4 or IPv6. If you attempt to configure a limit at both the global level and the family level, the device will not accept the configuration.

Options

family (inet | inet6)—(Optional) Specify either IPv4 or IPv6 messages to be counted towards the configured register message limit.

Default: Both IPv4 and IPv6 messages are counted towards the configured register message limit.

The remaining statements are described separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM State Limits on page 754
- clear pim join on page 1416
- clear pim register on page 1420

register-probe-time

Syntax register-probe-time *register-probe-time*;

Hierarchy Level [edit protocols pim rp]

Release Information Statement introduced in Junos OS Release 12.2 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D16 for QFX Series switches.

Description Specify the amount of time before the register suppression time (RST) expires when a designated switch can send a NULL-Register to the rendezvous point (RP).

Options ***register-probe-time***—Amount of time before the RST expires.

Default: **5** seconds

Range: **5** to **60** seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- PIM Overview on page 185
- Understanding PIM Sparse Mode on page 209

relay (AMT Protocol)

Syntax

```
relay {
  accounting;
  family {
    inet {
      anycast-prefix ip-prefix/<prefix-length>;
      local-address ip-address;
    }
  }
  secret-key-timeout minutes;
  tunnel-devicesvalue ;
  tunnel-limit number;
  unicast-stream-limitnumber;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols amt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt],
[edit protocols amt],
[edit routing-instances *routing-instance-name* protocols amt]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Configure the protocol address family, secret key timeout, and tunnel limit for Automatic Multicast Tunneling (AMT) relay functions.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

relay (IGMP)

Syntax

```
relay {
  defaults {
    (accounting | no-accounting);
    group-policy [ policy-names ];
    query-interval seconds;
    query-response-interval seconds;
    robust-count number;
    ssm-map ssm-map-name;
    version version;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* statement-name protocols igmp amt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* statement-name protocols igmp amt],
[edit protocols igmp amt],
[edit routing-instances *routing-instance-name* statement-name protocols igmp amt]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Configure default Automatic Multicast Tunneling (AMT) interface attributes.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Default IGMP Parameters for AMT Interfaces on page 425

reset-tracking-bit

Syntax reset-tracking-bit;

Hierarchy Level
[edit protocols pim],
[edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information
Statement introduced in Junos OS Release 10.1.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Change the value of a tracking bit (T-bit) field in the LAN prune delay hello option from the default of 1 to 0, which enables join suppression for a multicast interface. When the network starts receiving multiple identical join messages, join suppression triggers a random timer with a value of 66 through 84 milliseconds (1.1 × periodic through 1.4 × periodic, where periodic is 60 seconds). This creates an interval during which no identical join messages are sent. Eventually, only one of the identical messages is sent. Join suppression is triggered each time identical messages are sent for the same join.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Enabling Join Suppression on page 222
- override-interval on page 1192
- propagation-delay on page 1221

restart-duration (Multicast Snooping)

Syntax restart-duration *seconds*;

Hierarchy Level [edit multicast-snooping-options graceful-restart]

Release Information Statement introduced in Junos OS Release 9.2.

Description Configure the duration of the graceful restart interval.

Options ***seconds***— Graceful restart duration for multicast snooping.
Range: 0 through 300
Default: 180

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Multicast Snooping on page 874

restart-duration

Syntax restart-duration *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim graceful-restart],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim graceful-restart],
[edit protocols pim graceful-restart],
[edit routing-instances *routing-instance-name* protocols pim graceful-restart]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Configure the duration of the graceful restart interval.

Options ***seconds***—Time that the routing device waits (in seconds) to complete PIM sparse mode graceful restart.

Range: 30 through 300

Default: 60

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Sparse Mode Graceful Restart on page 383

reverse-oif-mapping

Syntax

```
reverse-oif-mapping {
    no-qos-adjust;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit logical-systems *logical-system-name* routing-options multicast interface *interface-name*],
[edit routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit routing-options multicast interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
The **no-qos-adjust** statement added in Junos OS Release 9.5.
The **no-qos-adjust** statement introduced in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description

Enable the routing device to identify a subscriber VLAN or interface based on an IGMP or MLD request it receives over the multicast VLAN.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast with Subscriber VLANs on page 912

rib-group (Protocols DVMRP)

Syntax rib-group *group-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description Associate a routing table group with DVMRP.

Options ***group-name***—Name of the routing table group. The name must be one that you defined with the **rib-groups** statement at the **[edit routing-options]** hierarchy level.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring DVMRP on page 434

rib-group (Protocols MSDP)

Syntax rib-group *group-name*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Associate a routing table group with MSDP.

Options *group-name*—Name of the routing table group. The name must be one that you defined with the **rib-groups** statement at the **[edit routing-options]** hierarchy level.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring MSDP in a Routing Instance on page 396

rib-group (Protocols PIM)

Syntax

```
rib-group {
    inet group-name;
    inet6 group-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Associate a routing table group with PIM.

Options

table-name—Name of the routing table. The name must be one that you defined with the **rib-groups** statement at the **[edit routing-options]** hierarchy level.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring a Dedicated PIM RPF Routing Table on page 806

robust-count (Bridge Domains)

Syntax robust-count *number*;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*]

Release Information Statement introduced in Junos OS Release 8.5.

Description Provide fine-tuning to allow for expected packet loss on a subnet. You can wait more intervals if subnet packet loss is high and IGMP report messages might be lost.

Options *number*—Robust interval.

Range: 2 through 10

Default: 2

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring IGMP Snooping on page 106

robust-count (Protocols IGMP)

Syntax robust-count *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Tune the expected packet loss on a subnet. This factor is used to calculate the group member interval, other querier present interval, and last-member query count.

Options *number*—Robustness variable.

Range: 2 through 10

Default: 2

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Modifying the IGMP Robustness Variable on page 34

robust-count (Protocols IGMP AMT)

Syntax robust-count *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description Configure the expected IGMP packet loss on an Automatic Multicast Tunneling (AMT) tunnel. If a tunnel is expected to have packet loss, increase the robust count.

Options *number*—Number of packets that can be lost before the AMT protocol deletes the multicast state.

Range: 2 through 10

Default: 2

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Default IGMP Parameters for AMT Interfaces on page 425

robust-count (IGMP Snooping)

Syntax robust-count *number*;

Hierarchy Level [edit protocols igmp-snooping vlan *vlan-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description Configure the number of intervals the device waits before removing a multicast group from the multicast forwarding table. Configure the length of each interval using the **query-interval** statement.

Default 2 intervals

Options *number*—Number of intervals the switch waits before timing out a multicast group.
Range: 2 through 10

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- IGMP Snooping Overview on page 81
- **igmp-snooping on page 1077**

robust-count (Protocols MLD)

Syntax robust-count *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before Junos OS Release 7.4.

Description Tune for the expected packet loss on a subnet.

Options *number*—Time interval. This interval must be less than the interval between general host-query messages.

Range: 2 through 10

Default: 2 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Modifying the MLD Robustness Variable on page 61

robust-count (MLD Snooping)

Syntax robust-count *number*;

Hierarchy Level [edit protocols mld-snooping vlan (all | *vlan-name*)]

[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 18.1R1 for the SRX1500 devices.
Support at the **[edit routing-instances *instance-name* protocols mld-snooping vlan *vlan-name*]** hierarchy level introduced in Junos OS Release 13.3 for EX Series switches.

Description Configure the number of queries the switch sends before removing a multicast group from the multicast forwarding table. We recommend that the robust count be set to the same value on all multicast routers and switches in the VLAN.

Default The default is the value of the **robust-count** statement configured for MLD. The default for the MLD **robust-count** statement is 2.

Options *number*—Number of queries the switch sends before timing out a multicast group.

Range: 2 through 10

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- **mld-snooping on page 1155**
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134
- Understanding MLD Snooping on page 125

robustness-count

Syntax robustness-count *number*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim interface *interface-name* bidirectional df-election],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional df-election],
[edit protocols pim interface *interface-name* bidirectional df-election],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bidirectional df-election]

Release Information
Statement introduced in Junos OS Release 12.1.
Statement introduced in Junos OS Release 13.3 for the PTX5000 router.

Description
Configure the designated forwarder (DF) election robustness count for bidirectional PIM. When a DF election Offer or Winner message fails to be received, the message is retransmitted. The **robustness-count** statement sets the minimum number of DF election messages that must fail to be received for DF election to fail. To prevent routing loops, all routers on the link must have a consistent view of the DF. When the DF election fails because DF election messages are not received, forwarding on bidirectional PIM routes is suspended.

If a router receives from a neighbor a better offer than its own, the router stops participating in the election for a period of **robustness-count** * offer-period. Eventually, all routers except the best candidate stop sending Offer messages.

Options
number—Number of transmission attempts for DF election messages.
Range: 1 through 10
Default: 3

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Understanding Bidirectional PIM on page 337
- Example: Configuring Bidirectional PIM on page 343

route-target (Protocols MVPN)

Syntax

```
route-target {
  export-target {
    target target-community;
    unicast;
  }
  import-target {
    target {
      target-value;
      receiver target-value;
      sender target-value;
    }
    unicast {
      receiver;
      sender;
    }
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information Statement introduced in Junos OS Release 8.4.

Description Enable you to override the Layer 3 VPN import and export route targets used for importing and exporting routes for the MBGP MVPN NLRI.

Default The multicast VPN routing instance uses the import and export route targets configured for the Layer 3 VPN.

Options The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring VRF Route Targets for Routing Instances for an MBGP MVPN*

rp

Syntax

```
register-probe-time {
  auto-rp {
    (announce | discovery | mapping);
    (mapping-agent-election | no-mapping-agent-election);
  }
  bidirectional {
    address address {
      group-ranges {
        destination-ip-prefix</prefix-length>;
      }
      hold-time seconds;
      priority number;
      }
    }
  bootstrap {
    family (inet | inet6) {
      export [ policy-names ];
      import [ policy-names ];
      priority number;
    }
  }
  bootstrap-export [ policy-names ];
  bootstrap-import [ policy-names ];
  bootstrap-priority number;
  dr-register-policy [ policy-names ];
  embedded-rp {
    group-ranges {
      destination-ip-prefix</prefix-length>;
    }
    maximum-rps limit;
  }
  group-rp-mapping {
    family (inet | inet6) {
      log-interval seconds;
      maximum limit;
      threshold value;
      }
    }
    log-interval seconds;
    maximum limit;
    threshold value;
    }
  }
  local {
    family (inet | inet6) {
      disable;
      address address;
      anycast-pim {
        local-address address;
          address address <forward-msdp-sa>;
          rp-set {
        }
```

```
        }
        group-ranges {
          destination-ip-prefix</prefix-length>;
        }
        hold-time seconds;
        override;
        priority number;
      }
    }
    register-limit {
      family (inet | inet6) {
        log-interval seconds;
        maximum limit;
        threshold value;
        }
      }
      log-interval seconds;
      maximum limit;
      threshold value;
      }
    }
    register-probe-time register-probe-time;
    }
    rp-register-policy [ policy-names ];
    static {
      address address {
        override;
        version version;
        group-ranges {
          destination-ip-prefix</prefix-length>;
        }
      }
    }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the routing device as an actual or potential RP. A routing device can be an RP for more than one group.

The remaining statements are explained separately. See CLI Explorer.

Default

If you do not include the **rp** statement, the routing device can never become the RP.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Understanding PIM Sparse Mode on page 209

rp-register-policy

Syntax
rp-register-policy [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information
Statement introduced in Junos OS Release 7.6.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Apply one or more policies to control incoming PIM register messages.

Options
policy-names—Name of one or more import policies.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Register Message Filters on a PIM RP and DR on page 279
- dr-register-policy on page 1017

rp-set

Syntax

```
rp-set {
    address address <forward-msdp-sa>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim local family (inet | inet6) anycast-pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim local family (inet | inet6) anycast-pim],
[edit protocols pim local family (inet | inet6) anycast-pim],
[edit routing-instances *routing-instance-name* protocols pim local family (inet | inet6) anycast-pim]

Release Information

Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure a set of rendezvous point (RP) addresses for anycast RP. You can configure up to 15 RPs.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM Anycast With or Without MSDP on page 248

rpf-check-policy (Routing Options RPF)

Syntax rpf-check-policy [*policy-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information
Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply policies for disabling RPF checks on arriving multicast packets. The policies must be correctly configured.

Options ***policy-names***—Name of one or more multicast RPF check policies.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring RPF Policies on page 813

rpf-selection

Syntax

```
rpf-selection {
  group group-address {
    sourcesource-address {
      next-hop next-hop-address;
    }
    wildcard-source {
      next-hop next-hop-address;
    }
  }
  prefix-list prefix-list-addresses {
    source source-address {
      next-hop next-hop-address;
    }
    wildcard-source {
      next-hop next-hop-address;
    }
  }
```

Hierarchy Level

[edit routing-instances *routing-instance-name* protocols pim]
[edit protocols pim]

Release Information

Statement introduced in JUNOS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the PIM RPF next-hop neighbor for a specific group and source for a VRF routing instance.

NOTE: Starting in Junos OS 17.4R1, you can configure **rpf-selection** statement at the **[edit protocols pim]** hierarchy level.

The remaining statements are explained separately. See CLI Explorer.

Default

If you omit the **rpf-selection** statement, PIM RPF checks typically choose the best path determined by the unicast protocol for all multicast flows.

Options

source-address—Specific source address for the PIM group.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM RPF Selection on page 816

rpf-vector (PIM)

Syntax

```
rpf-vector {
  policy (rpf-vector)[ policy-name];
}
```

Hierarchy Level

[edit dynamic-profiles *name* protocols pim],
[edit logical-systems *name* protocols pim],
[edit logical-systems *name* routing-instances *name* protocols pim],
[edit protocols pim],
[edit routing-instances *name* protocols pim]

Release Information

Statement introduced in Junos OS Release 17.3R1.

Description

This feature provides a way for PIM source-specific multicast (SSM) to resolve Vector Type Length (TLV) for multicast in a seamless Multiprotocol Label Switching (MPLS) networks. In other words, it enables PIM to build multicast trees through an MPLS core. **rpf-vector** implements RFC 5496, Reverse Path Forwarding (RPF) Vector TLV .

When **rpf-vector** is enabled on an edge router that sends PIM join messages into the core, the join message includes a vector specifying the IP address of the next edge router along the path to the root of the multicast distribution tree (MDT). The core routers can then process the join message by sending it towards the specified edge router (i.e., toward the Vector). The address of the edge router serves as the RPF vector in the PIM join message so routers in the core can resolve the next-hop towards the source without the need for BGP in the core.

Only the IPv4 address family is supported.

Options

policy— Create a filter policy to determine whether or not to apply **rpf-vector**.

Required Privilege Level

routing

Related Documentation

- show pim join on page 1642 extensive
- show pim neighbors on page 1657 detail
- policy (rpf-vector) on page 1211

rpt-spt

Syntax rpt-spt;

Hierarchy Level [edit logical-systems *profile-name* routing-instances *instance-name* protocols mvpn mvpn-mode],
[edit routing-instances *instance-name* protocols mvpn mvpn-mode]

Release Information Statement introduced in Junos OS Release 10.0.

Description Use rendezvous-point trees for customer PIM (C-PIM) join messages, and switch to the shortest-path tree after the source is known.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

rsvp-te (Routing Instances Provider Tunnel Selective)

Syntax

```
rsvp-te {
  label-switched-path-template {
    (default-template | lsp-template-name);
  }
  static-lsp lsp-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Configure the properties of the RSVP traffic-engineered point-to-multipoint LSP for MBGP MVPNs.

The remaining statements are explained separately. See CLI Explorer.

NOTE: Junos OS Release 11.2 and earlier do not support point-to-multipoint LSPs with next-generation multicast VPNs on MX80 routers.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*

sa-hold-time (Protocols MSDP)

Syntax sa-hold-time *seconds*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer address],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name* peer *address*]
[edit routing-instances *instance-name* protocols msdp peer *address*],

Release Information Statement introduced in Junos OS Release 12.3.

Description Specify the source address (SA) message hold time to use when maintaining a connection with the MSDP peer. Each entry in an SA cache has an associated hold time. The hold timer is started when an SA message is received by an MSDP peer. The timer is reset when another SA message is received before the timer expires. If another SA message is not received during the SA message hold-time period, the SA message is removed from the cache.

You might want to change the SA message hold time for consistency in a multi-vendor environment.

Options ***seconds***—Source address message hold time.

Range: 75 through 300 seconds

Default: 75 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring MSDP on page 393
- hold-time (Protocols MSDP) on page 1068
- keep-alive (Protocols MSDP) on page 1118

sap

Syntax

```
sap {
  disable;
  listen address <port port>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols],
[edit protocols]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

Enable the router to listen to session directory announcements for multimedia and other multicast sessions.

SAP and SDP always listen on the default SAP address and port, 224.2.127.254:9875. To have SAP listen on additional addresses or pairs of address and port, include a **listen** statement for each address or pair.

Options

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Session Announcement Protocol on page 415
- listen on page 1125

scope

Syntax

```
scope scope-name {
  interface [ interface-names ];
  prefix destination-prefix;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure multicast scoping.

Options

scope-name—Name of the multicast scope.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Multicast Snooping on page 873

scope-policy

Syntax scope-policy [*policy-names*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-options multicast]

NOTE: You can configure a scope policy at these two hierarchy levels only. You cannot apply a scope policy to a specific routing instance, because all scoping policies are applied to all routing instances. However, you can apply the **scope** statement to a specific routing instance at the **[edit routing-instances *routing-instance-name* routing-options multicast]** or **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast]** hierarchy level.

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

Description Apply policies for scoping. The policy must be correctly configured at the **edit policy-options policy-statement** hierarchy level.

Options ***policy-names***—Name of one or more multicast scope policies.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- scope on page 1273

secret-key-timeout

Syntax secret-key-timeout *minutes*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols amt relay],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay],
[edit protocols amt relay],
[edit routing-instances *routing-instance-name* protocols amt relay]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify the period in minutes after which the local opaque secret key used in the Automatic Multicast Tunneling (AMT) Message Authentication Code (MAC) times out and is regenerated.

Default 60 minutes

Options ***minutes***—Number of minutes to wait before generating a new MAC opaque secret key.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

selective

Syntax

```
selective {
  group multicast-prefix/prefix-length {
    source ip-prefix/prefix-length {
      ingress-replication {
        create-new-ucast-tunnel;
          label-switched-path-template {
            (default-template | lsp-template-name);
          }
        }
      ldp-p2mp;
      pim-ssm {
        group-range multicast-prefix;
      }
      rsvp-te {
        label-switched-path-template {
          (default-template | lsp-template-name);
        }
        static-lsp point-to-multipoint-lsp-name;
      }
      threshold-rate kbps;
    }
    wildcard-source {
      ldp-p2mp;
        pim-ssm {
          group-range multicast-prefix;
        }
        rsvp-te {
          label-switched-path-template {
            (default-template | lsp-template-name);
          }
        }
      static-lsp point-to-multipoint-lsp-name;
      }
      threshold-rate kbps;
    }
  }
  tunnel-limit number;
  wildcard-group-inet {
    wildcard-source {
      ldp-p2mp;
      pim-ssm {
        group-range multicast-prefix;
      }
      rsvp-te {
        label-switched-path-template {
          (default-template | lsp-template-name);
        }
        static-lsp lsp-name;
      }
      threshold-rate number;
    }
  }
```

```
wildcard-group-inet6 {
  wildcard-source {
    ldp-p2mp;
    pim-ssm {
      group-range multicast-prefix;
    }
    rsvp-te {
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
      static-lsp lsp-name;
    }
    threshold-rate number;
  }
 }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel],
[edit routing-instances *routing-instance-name* provider-tunnel]

Release Information

Statement introduced in Junos OS Release 8.5.
The **ingress-replication** statement and substatements added in Junos OS Release 10.4.

Description

Configure selective point-to-multipoint LSPs for an MBGP MVPN. Selective point-to-multipoint LSPs send traffic only to the receivers configured for the MBGP MVPNs, helping to minimize flooding in the service provider's network.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- *Configuring PIM-SSM GRE Selective Provider Tunnels*

sender-based-rpf (MBGP MVPN)

Syntax sender-based-rpf;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information Statement introduced in Junos OS Release 14.2.

Description In a BGP multicast VPN (MVPN) with RSVP-TE point-to-multipoint provider tunnels, configure a downstream provider edge (PE) router to forward multicast traffic only from a selected upstream sender PE router.

BGP MVPNs use an alternative to data-driven-event solutions and bidirectional mode DF election because, for one thing, the core network is not exactly a LAN. Because, in an MVPN scenario, it is possible to determine which PE router has sent the traffic, Junos OS uses this information to only forward the traffic if it is sent from the correct PE router. With sender-based RPF, the RPF check is enhanced to check whether data arrived on the correct incoming virtual tunnel (vt-) interface and that the data was sent from the correct upstream PE router.

More specifically, the data must arrive with the correct MPLS label in the outer header used to encapsulate data through the core. The label identifies the tunnel and, if the tunnel is point-to-multipoint, the upstream PE router.

Sender-based RPF is not a replacement for single-forwarder election, but is a complementary feature. Configuring a higher primary loopback address (or router ID) on one PE device (PE1) than on another (PE2) ensures that PE1 is the single-forwarder election winner. The **unicast-umh-election** statement causes the unicast route preference to determine the single-forwarder election. If single-forwarder election is not used or if it is not sufficient to prevent duplicates in the core, sender-based RPF is recommended.

For RSVP point-to-multipoint provider tunnels, the transport label identifies the sending PE router because it is a requirement that penultimate hop popping (PHP) is disabled when using point-to-multipoint provider tunnels with MVPNs. PHP is disabled by default when you configure the MVPN protocol in a routing instance. The label identifies the tunnel, and (because the RSVP-TE tunnel is point-to-multipoint) the sending PE router.

The sender-based RPF mechanism is described in RFC 6513, *Multicast in MPLS/BGP IP VPNs* in section 9.1.1.

Sender-based RPF prevents duplicates from being sent to the customer even if there is duplication in the provider network. Duplication could exist in the provider because of a hot-root standby configuration or if the single-forwarder election is not sufficient to prevent duplicates. Single-forwarder election is used to prevent duplicates to the core network, while sender-based RPF prevents duplicates to the customer even if there are duplicates in the core. There are cases in which single-forwarder election cannot prevent duplicate traffic from arriving at the egress PE router. One example of this (outlined in

section 9.3.1 of RFC 6513) is when PIM sparse mode is configured in the customer network and the MVPN is in RPT-SPT mode with an I-PMSI.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- unicast-umh-election on page 1364

sglimit

Syntax

```
sglimit {
  family (inet | inet6) {
    log-interval seconds;
    maximum limit;
    threshold value;
  }
  log-interval seconds;
  maximum limit;
  threshold value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced in Junos OS Release 12.2.

Description

Configure a limit for the number of accepted (*,G) and (S,G) PIM join states.

NOTE: The maximum limit settings that you configure with the **maximum** and the **family (inet | inet6) maximum** statements are mutually exclusive. For example, if you configure a global maximum PIM join state limit, you cannot configure a limit at the family level for IPv4 or IPv6 joins. If you attempt to configure a limit at both the global level and the family level, the device will not accept the configuration.

Options

family (inet | inet6)—(Optional) Specify either IPv4 or IPv6 join states to be counted towards the configured join state limit.

Default: Both IPv4 and IPv6 join states are counted towards the configured join state limit.

The remaining statements are described separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM State Limits on page 754
- clear pim join on page 1416

signaling

Syntax signaling;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp family inet-mdt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp family inet-mvpn],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* family inet-mdt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols bgp group *group-name* family inet-mvpn],
[edit routing-instances *routing-instance-name* protocols bgp family inet-mdt],
[edit routing-instances *routing-instance-name* protocols bgp family inet-mvpn],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* family inet-mdt],
[edit routing-instances *routing-instance-name* protocols bgp group *group-name* family inet-mvpn]

Release Information Statement introduced in Junos OS Release 9.4.
Statement introduced in Junos OS Release 11.1 for EX Series switches.

Description Enable signaling in BGP. For multicast distribution tree (MDT) subaddress family identifier (SAFI) NLRI signaling, configure signaling under the **inet-mdt** family. For multiprotocol BGP (MBGP) intra-AS NLRI signaling, configure signaling under the **inet-mvpn** family.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 491

snoop-pseudowires

Syntax snoop-pseudowires;

Hierarchy Level [edit routing-instances *routing-instance-name* igmp-snooping-options]
[edit logical-systems *logical-system -name* routing-instances
*routing-instance-name*igmp-snooping-options]

Release Information Statement introduced in Junos OS Release 15.1.

Description The default IGMP snooping implementation for a VPLS instance adds each pseudowire interface to its **oif** list. It includes traffic from the ingress PE that is sent to egress PE even if there is no interest. The **snoop-pseudowires** option prevents multicast traffic from traversing the pseudowire (to egress PEs) unless there are IGMP receivers for the traffic. In other words, multicast traffic is forwarded only to VPLS core interfaces that are router interfaces, or that are IGMP receivers. In addition to the benefit of sending traffic to only interested PEs, **snoop-pseudowires** also optimizes a common path between PE-P routers wherever possible (so if two PEs connect via the same P router, only one copy of packet is sent; the packet would be replicated only on P routers for which the path is divergent).

NOTE: Note that this option can only be enabled when **instance-type** is *vpls*. The **snoop-pseudowires** option cannot be enabled if **use-p2mp-lsp** is enabled for **igmp-snooping-options**.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *instance-type*
- Example: Configuring IGMP Snooping on page 106

source-active-advertisement

Syntax

```
source-active-advertisement {
    dampen minutes;
    min-rate seconds;
}
```

Hierarchy Level

[edit logical-systems *logical-system--name* protocols mvpn mvpn-mode spt-only],
[edit logical-systems *logical-system--name* routing-instances *instance-name* protocols mvpn mvpn-mode spt-only],
[edit routing-instances protocols mvpn mvpn-mode spt-only],
[edit routing-instances *instance-name* protocols mvpn mvpn-mode spt-only]

Release Information

Statement introduced in Junos OS Release 17.1.

Description

Attributes associated with advertising Source-Active A-D routes.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*

source (Bridge Domains)

Syntax

source *ip-address*;

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name* static group],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name* static group],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name* static group],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols vlan *vlan-id* igmp-snooping interface *interface-name* static group]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Statically define multicast group source addresses on an interface.

Options

ip-address—IP address to use as the source for the group.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

source (Distributed IGMP)

Syntax source *source-address* <distributed>;

Hierarchy Level [edit protocols pim static group *multicast-group-address*]

Release Information Statement introduced in Junos OS Release 14.1X50.

Description Specify an IP unicast source address for a multicast group being statically configured on an interface.

Options

distributed—(Optional) Enable a static join for multiple multicast address groups so that all Packet Forwarding Engines receive traffic, but preprovision only one multicast group.

source-address—Specific IP unicast source address for a multicast group.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling Distributed IGMP on page 76
- For general information about configuring IGMP, see the *Multicast Protocols Feature Guide*
- For information about enabling IGMP, see “Enabling IGMP” in the *Multicast Protocols Feature Guide*

source (Multicast VLAN Registration)

Syntax

```
source {
  groups group-prefix;
}
```

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) data-forwarding]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.

Description Configure a VLAN to be a multicast source VLAN (MVLAN).

The remaining statement is explained separately. See CLI Explorer.

Default Disabled

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

source (PIM RPF Selection)

Syntax

```
source source-address {
  next-hop next-hop-address;
}
```

Hierarchy Level

[edit routing-instances *routing-instance-name* protocols pim rpf-selection group *group-address*],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection prefix-list *prefix-list-addresses*]

Release Information

Statement introduced in JUNOS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the source address for the PIM group.

Options

source-address—Specific source address for the PIM group.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM RPF Selection on page 816

source (Protocols IGMP)

Syntax

```
source ip-address {
    source-count number;
    source-increment increment;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static group *multicast-group-address*],
[edit protocols igmp interface *interface-name* static group *multicast-group-address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Specify the IP version 4 (IPv4) unicast source address for the multicast group being statically configured on an interface.

Options

ip-address—IPv4 unicast address.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP Static Group Membership on page 37

source (Protocols MLD)

Syntax

```
source ip-address {
    source-count number;
    source-increment increment;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols mld interface *interface-name* static group *multicast-group-address*],
[edit protocols mld interface *interface-name* static group *multicast-group-address*]

Release Information

Statement introduced before Junos OS Release 7.4.

Description

IP version 6 (IPv6) unicast source address for the multicast group being statically configured on an interface.

Options

ip-address — One or more IPv6 unicast addresses.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD Static Group Membership on page 63

source (Protocols MSDP)

Syntax

```
source ip-address</prefix-length> {
  active-source-limit {
    maximum number;
    threshold number;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Limit the number of active source messages the routing device accepts from sources in this address range.

Default

If you do not include this statement, the routing device accepts any number of MSDP active source messages.

Options

The other statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404

source (Routing Instances)

Syntax

```
source source-address {
    rate threshold-rate;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *intet* | *inet6* mdt threshold group *group-address*],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address*],
[edit routing-instances *routing-instance-name* provider-tunnel family *intet* | *inet6* mdt threshold group *group-address*]

Release Information

Statement introduced before Junos OS Release 7.4. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description

Establish a threshold to trigger the automatic creation of a data MDT for the specified unicast address or prefix of the source of multicast information.

Options

source-address—Explicit unicast address of the multicast source.

The remaining statement is explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

source (Routing Instances Provider Tunnel Selective)

Syntax

```
source source-address {
  ldp-p2mp;
  pim-ssm {
    group-range multicast-prefix;
  }
  rsvp-te {
    label-switched-path-template {
      (default-template | lsp-template-name);
    }
    static-lsp lsp-name;
  }
  threshold-rate number;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address*],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address*]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Specify the IP address for the multicast source. This statement is a part of the point-to-multipoint LSP and PIM-SSM GRE selective provider tunnel configuration for MBGP MVPNs.

Options

source-address—IP address for the multicast source.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- *Configuring PIM-SSM GRE Selective Provider Tunnels*

source (Source-Specific Multicast)

Syntax source [*addresses*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast ssm-map *ssm-map-name*],
[edit logical-systems *logical-system-name* routing-options multicast ssm-map *ssm-map-name*],
[edit routing-instances *routing-instance-name* routing-options multicast ssm-map *ssm-map-name*],
[edit routing-options multicast ssm-map *ssm-map-name*]

Release Information Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description Specify IPv4 or IPv6 source addresses for an SSM map.

Options ***addresses***—IPv4 or IPv6 source addresses.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To view this statement in the configuration.

Related Documentation

- Example: Configuring SSM Mapping on page 322

source-address

Syntax source-address *ip-address*;

Hierarchy Level [edit bridge-domains *bridge-domain-name* protocols igmp-snooping proxy],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* proxy],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping proxy],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* proxy]

Release Information Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 13.2 for the QFX series.

Description Specify the IP address to use as the source for IGMP snooping reports in proxy mode. Reports are sent with address 0.0.0.0 as the source address unless there is a source address configured. You can also use this statement to configure the source address to use for IGMP snooping queries.

Options ***ip-address***—IP address to use as the source for proxy-mode IGMP snooping reports.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring IGMP Snooping on page 106

source-count (Protocols IGMP)

Syntax source-count *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static group *multicast-group-address* source],
[edit protocols igmp interface *interface-name* static group *multicast-group-address* source]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the number of multicast source addresses that should be accepted for each static group created.

Options *number*—Number of source addresses.

Default: 1

Range: 1 through 1024

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP Static Group Membership on page 37

source-count (Protocols MLD)

Syntax source-count *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name* static group *multicast-group-address* source],
[edit protocols mld interface *interface-name* static group *multicast-group-address* source]

Release Information Statement introduced in Junos OS Release 9.6.

Description Configure the number of multicast source addresses that should be accepted for each static group created.

Options *number*—Number of source addresses.

Default: 1

Range: 1 through 1024

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD Static Group Membership on page 63

source-increment (Protocols IGMP)

Syntax source-increment *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name* static group *multicast-group-address* source],
[edit protocols igmp interface *interface-name* static group *multicast-group-address* source]

Release Information Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the number of times the multicast source address should be incremented for each static group created. The increment is specified in dotted decimal notation similar to an IPv4 address.

Options *increment*—Number of times the source address should be incremented.

Default: 0.0.0.1

Range: 0.0.0.1 through 255.255.255.255

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP Static Group Membership on page 37

source-increment (Protocols MLD)

Syntax source-increment *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name* static group *multicast-group-address* source],
[edit protocols mld interface *interface-name* static group *multicast-group-address* source]

Release Information Statement introduced in Junos OS Release 9.6.

Description Configure the number of times the address should be incremented for each static group created. The increment is specified in a format similar to an IPv6 address.

Options *increment*—Number of times the source address should be incremented.

Default: ::1

Range: ::1 through ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD Static Group Membership on page 63

source-tree (MBGP MVPN)

Syntax source-tree;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn static-umh],
[edit routing-instances *routing-instance-name* protocols mvpn static-umh]

Release Information Statement introduced in Junos OS Release 15.1.

Description Specify that a statically selected upstream multicast hop (UMH) only affects type 7 (S,G) routes.

The **source-tree** option is mandatory. Type 6 routes are sent toward the rendezvous point (RP), and use the dynamic UMH selection that is configured with the unicast-umh-election statement, or the default method of highest IP address is used if **unicast-umh-election** is not configured.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- **sender-based-rpf on page 1278**
- **static-umh (MBGP MVPN) on page 1315**
- **unicast-umh-election on page 1364**

source-vlans

Syntax source-vlans *vlan-list*;

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) data-forwarding receiver]

Release Information Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 12.3 for the QFX Series.

Description Specify a list of multicast VLANs (MVLANs) from which this multicast receiver VLAN receives multicast traffic. Either all of these MVLANs must be in proxy mode or none of them can be in proxy mode.

Default Disabled

Options ***vlan-list***—Names of the MVLANs.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast VLAN Registration on EX Series Switches on page 178
- Configuring Multicast VLAN Registration on EX Series Switches (CLI Procedure) on page 177

spt-only

Syntax spt-only;

Hierarchy Level [edit logical-systems *profile-name* routing-instances *instance-name* protocols mvpn mvpn-mode],
[edit routing-instances *instance-name* protocols mvpn mvpn-mode]

Release Information Statement introduced in Junos OS Release 10.0.

Description Set the MVPN mode to learn about active multicast sources using multicast VPN source-active routes. This is the default mode.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring SPT-Only Mode for Multiprotocol BGP-Based Multicast VPNs*

spt-threshold

Syntax

```
spt-threshold {
  infinity [ policy-names ];
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced in Junos OS Release 8.0.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Set the SPT threshold to infinity for a source-group address pair. Last-hop multicast routing devices running PIM sparse mode can forward the same stream of multicast packets onto the same LAN through an RPT rooted at the RP or an SPT rooted at the source. By default, last-hop routing devices transition to a direct SPT to the source. You can configure this routing device to set the SPT transition value to infinity to prevent this transition for any source-group address pair.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring the PIM SPT Threshold Policy on page 293

ssm-groups

Syntax ssm-groups [*ip-addresses*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure source-specific multicast (SSM) groups.

By default, the SSM group multicast address is limited to the IP address range from 232.0.0.0 through 232.255.255.255. However, you can extend SSM operations into another Class D range by including the **ssm-groups** statement in the configuration. The default SSM address range from 232.0.0.0 through 232.255.255.255 cannot be used in the **ssm-groups** statement. This statement is for adding other multicast addresses to the default SSM group addresses. This statement does not override the default SSM group address range.

IGMPv3 supports SSM groups. By utilizing inclusion lists, only sources that are specified send to the SSM group.

Options ***ip-addresses***—List of one or more additional SSM group addresses separated by a space.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 316

ssm-map (Protocols IGMP)

Syntax ssm-map *ssm-map-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply an SSM map to an IGMP interface.

Options ***ssm-map-name***—Name of SSM map.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring SSM Mapping on page 322

ssm-map (Protocols IGMP AMT)

Syntax ssm-map *ssm-map-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description Apply a source-specific multicast (SSM) map to all Automatic Multicast Tunneling (AMT) interfaces.

Options ***ssm-map-name***—Name of the SSM map.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring Default IGMP Parameters for AMT Interfaces on page 425

ssm-map (Protocols MLD)

Syntax ssm-map *ssm-map-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*], [edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 7.4.

Description Apply an SSM map to an MLD interface.

Options ***ssm-map-name***—Name of SSM map.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring SSM Mapping on page 322

ssm-map (Routing Options Multicast)

Syntax

```
ssm-map ssm-map-name {
    policy [ policy-names ];
    source [ addresses ];
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure SSM mapping.

Options

ssm-map-name—Name of the SSM map.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring SSM Mapping on page 322

ssm-map-policy (MLD)

Syntax ssm-map-policy *ssm-map-policy-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.4.

Description Apply an SSM map policy to a statically configured MLD interface.

For dynamically-configured MLD interfaces, use the **ssm-map-policy (Dynamic MLD Interface)** statement.

Options ***ssm-map-policy-name***—Name of SSM map policy.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring SSM Maps for Different Groups to Different Sources on page 333

ssm-map-policy (IGMP)

Syntax ssm-map-policy *ssm-map-policy-name*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Apply an SSM map policy to a statically configured IGMP interface.

For dynamically-configured IGMP interfaces, use the **ssm-map-policy (Dynamic IGMP Interface)** statement.

Options ***ssm-map-policy-name***—Name of SSM map policy.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring SSM Maps for Different Groups to Different Sources on page 333

standby-path-creation-delay

Syntax standby-path-creation-delay <*seconds*>;

Hierarchy Level [edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced in Junos OS Release 12.2.

Description Configure the time interval after which a standby path is created, when a new ECMP interface or neighbor is added to the network.

In the absence of this statement, ECMP joins are redistributed as soon as a new ECMP interface or neighbor is added to the network.

Options ***<seconds>***—Time interval after which a standby path is created, when a new ECMP interface or neighbor is added to the network. Range is from 1 through 300.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM Make-Before-Break Join Load Balancing on page 790
- Configuring PIM Join Load Balancing on page 218
- clear pim join-distribution on page 1418
- join-load-balance on page 1116
- idle-standby-path-switchover-delay on page 1074

static (Bridge Domains)

Syntax

```
static {
  group multicast-group-address {
    source ip-address;
  }
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping interface *interface-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping vlan *vlan-id* interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 8.5.

Description

Define static multicast groups on an interface.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on page 106

static (Distributed IGMP)

Syntax

```
static {
  <distributed>;
  group multicast-group-address{
    <distributed>;
    source source- address<distributed>;
  }
}
```

Hierarchy Level [edit protocols pim]

Release Information Statement introduced in Junos OS Release 14.1X50.

Description Configure static source and group (S, G) addresses when distributed IGMP is enabled. Reduces the first join delay time and brings multicast traffic to the last-hop router. Specified (S, G) addresses join statically without waiting for the first join.

Options

distributed—(Optional) Enable static joins for specified (S,G) addresses and preprovision all of them so that all distributed IGMP Packet Forwarding Engines receive traffic.

The remaining statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling Distributed IGMP on page 76
- For general information about configuring IGMP, see the *Multicast Protocols Feature Guide*
- For information about enabling IGMP, see “Enabling IGMP” in the *Multicast Protocols Feature Guide*

static (IGMP Snooping)

Syntax

```
static {
  group ip-address;
}
```

Hierarchy Level [edit protocols igmp-snooping vlan (all | *vlan-name*) interface *interface-name*

Release Information Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Statically define multicast groups on an interface.

The remaining statement is explained separately. See CLI Explorer.

Default No multicast groups are statically defined.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring IGMP Snooping (CLI Procedure)*
- show igmp snooping membership on page 1483
- show igmp-snooping vlans on page 1494

static (Protocols IGMP)

Syntax

```
static {
  group multicast-group-address {
    exclude;
    group-count number;
    group-increment increment;
    source ip-address {
      source-count number;
      source-increment increment;
    }
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*], [edit protocols igmp interface *interface-name*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Test multicast forwarding on an interface without a receiver host.

The **static** statement simulates IGMP joins on a routing device statically on an interface without any IGMP hosts. It is supported for both IGMPv2 and IGMPv3 joins. This statement is especially useful for testing multicast forwarding on an interface without a receiver host.

NOTE: To prevent joining too many groups accidentally, the static statement is not supported with the interface all statement.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Enabling IGMP Static Group Membership on page 37

static (Protocols MLD)

Syntax

```
static {
  group multicast-group-address {
    exclude;
    group-count number;
    group-increment increment;
    source ip-address {
      source-count number;
      source-increment increment;
    }
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*], [edit protocols mld interface *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Test multicast forwarding on an interface.

The **static** statement simulates MLD joins on a routing device statically on an interface without any MLD hosts. It is supported for both MLDv1 and MLDv2 joins. This statement is especially useful for testing multicast forwarding on an interface without a receiver host.

NOTE: To prevent joining too many groups accidentally, the static statement is not supported with the interface all statement.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Enabling MLD Static Group Membership on page 63

static (Protocols PIM)

Syntax

```
static {
  address address {
    group-ranges {
      destination-ip-prefix</prefix-length>;
    }
    override;
    version version;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim rp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp],
[edit protocols pim rp],
[edit routing-instances *routing-instance-name* protocols pim rp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure static RP addresses. The default static RP address is 224.0.0.0/4. To configure other addresses, include one or more **address** statements. You can configure a static RP in a logical system only if the logical system is not directly connected to a source.

For each static RP address, you can optionally specify the PIM version and the groups for which this address can be the RP. The default PIM version is version 1.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the Static PIM RP Address on the Non-RP Routing Device on page 242

static-lsp

Syntax static-lsp *lsp-name*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* wildcard-source rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source rsvp-te],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address* rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* wildcard-source rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source rsvp-te],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source rsvp-te]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the name of the static point-to-multipoint (P2MP) LSP used for a specific MBGP MVPN; static P2MP LSP cannot be shared by multiple VPNs. Use this statement to specify the static LSP for both inclusive and selective point-to-multipoint LSPs.

Use a static P2MP LSP when you know all the egress PE router endpoints (receiver nodes) and you want to avoid the setup delay incurred by dynamically created P2MP LSPs (configured with the **label-switched-path-template**). These static LSPs are signaled before the MVPN requires or uses them, consequently avoiding any signaling latency and minimizing traffic loss due to latency.

If you add new endpoints after the static P2MP LSP is established, you must update the configuration on the ingress PE router. In contrast, a dynamic P2MP LSP learns new endpoints without any configuration changes.

BEST PRACTICE: Multiple multicast flows can share the same static P2MP LSP; this is the preferred configuration when the set of egress PE router endpoints on the LSP are all interested in the same set of multicast flows. When the set of relevant flows is different between endpoints, we recommend that you create a new static P2MP LSP to associate endpoints with flows of interest.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Point-to-Multipoint LSPs Overview*
- *Configuring Static LSPs*
- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- *Example: Configuring an RSVP-Signaled Point-to-Multipoint LSP on Logical Systems*

static-umh (MBGP MVPN)

Syntax

```
static-umh {
  primary address;
  backup address;
  source-tree;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information

Statement introduced in Junos OS Release 15.1.

Description

In a BGP multicast VPN (MVPN) with RSVP-TE point-to-multipoint provider tunnels, statically set the upstream multicast hop (UMH), instead of using one of the dynamic methods to choose the UMH routers, such as that described in unicast-umh-election.

The **static-umh** statement causes all type 7 (S,G) routes to use the configured primary and backup upstream multicast hops. If these UMHs are not available, no UMH is selected. If the primary is not available, but the backup UMH is available, the backup is used as the UMH.

The **static-umh** statement only affects type 7 (S,G) routes. Type 6 routes are sent toward the rendezvous point (RP), and use the dynamic UMH selection that is configured with the **unicast-umh-election** statement, or the default method of highest IP address is used if **unicast-umh-election** is not configured.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 542
- Example: Configuring Sender-Based RPF in a BGP MVPN with RSVP-TE Point-to-Multipoint Provider Tunnels on page 716
- sender-based-rpf on page 1278
- unicast-umh-election on page 1364

stream-protection (Multicast-Only Fast Reroute)

Syntax

```
stream-protection {
  mofrr-asm-starg;
  mofrr-disjoint-upstream-only;
  mofrr-no-backup-join;
  mofrr-primary-path-selection-by-routing;
  policy policy-name;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-systems *logical-system-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information

Statement introduced in Junos OS Release 14.1.
Statement introduced in Junos OS Release 17.4R1 for QFX Series switches.

Description

Enable multicast-only fast reroute (MoFRR) on a routing or switching device. MoFRR minimizes packet loss in a network when there is a link failure.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Understanding Multicast-Only Fast Reroute on page 822
- Understanding Multicast-Only Fast Reroute on Switches on page 829
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on Switches on page 844
- Example: Configuring Multicast-Only Fast Reroute in a Multipoint LDP Domain on page 852

subscriber-leave-timer

Syntax subscriber-leave-timer *seconds*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit logical-systems *logical-system-name* routing-options multicast interface *interface-name*],
[edit routing-instances *routing-instance-name* routing-options multicast interface *interface-name*],
[edit routing-options multicast interface *interface-name*]

Release Information Statement introduced in Junos OS Release 9.2.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Length of time before the multicast VLAN updates QoS data (for example, available bandwidth) for subscriber interfaces after it receives an IGMP leave message.

Options ***seconds***—Length of time before the multicast VLAN updates QoS data (for example, available bandwidth) for subscriber interfaces after it receives an IGMP leave message. Specifying a value of 0 results in an immediate update. This is the same as if the statement were not configured.

Range: 0 through 30

Default: 0 seconds

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

target (Routing Instances MVPN)

Syntax

```
target target-value {
    receiver target-value;
    sender target-value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target import-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target import-target]

Release Information

Statement introduced in Junos OS Release 8.4.

Description

Specify the target value when importing sender and receiver site routes.

Options

target-value—Specify the target value when importing sender and receiver site routes.

receiver—Specify the target community used when importing receiver site routes.

sender—Specify the target community used when importing sender site routes.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring VRF Route Targets for Routing Instances for an MBGP MVPN*

threshold (Bridge Domains)

Syntax threshold suppress *value* <reuse *value*>;

Hierarchy Level
[edit bridge-domains *bridge-domain-name* multicast-snooping-options forwarding-cache],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* multicast-snooping-options forwarding-cache],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options forwarding-cache],
[edit routing-instances *routing-instance-name* multicast-snooping-options forwarding-cache],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* multicast-snooping-options forwarding-cache]

Release Information Statement introduced in Junos OS Release 8.5.

Description Configure the suppression and reuse thresholds for multicast snooping forwarding cache limits.

Options

suppress *value*—Value to begin suppressing new multicast forwarding cache entries. This value is mandatory. This number must be greater than the reuse value.

Range: 1 through 200,000

reuse *value*—(Optional) Value to begin creating new multicast forwarding cache entries. If configured, this number must be less than the suppress value.

Range: 1 through 200,000

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Multicast Snooping on page 874

threshold (MSDP Active Source Messages)

Syntax threshold *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols msdp active-source-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp active-source-limit],
[edit protocols msdp active-source-limit],
[edit routing-instances *routing-instance-name* protocols msdp active-source-limit]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the random early detection (RED) threshold for MSDP active source messages. This number must be less than the configured or default maximum.

Options *number*—RED threshold for active source messages.

Range: 1 through 1,000,000

Default: 24,000

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring MSDP with Active Source Limits and Mesh Groups on page 404
- maximum (MSDP Active Source Messages) on page 1139

threshold (Multicast Forwarding Cache)

Syntax

```
threshold {
  log-warning value;
  suppress value;
  reuse value;
  mvpn-rpt-suppress value;
  mvpn-rpt-reuse value;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache family (inet | inet6)],
[edit logical-systems *logical-system-name* routing-options multicast forwarding-cache],
[edit logical-systems *logical-system-name* routing-options multicast forwarding-cache family (inet | inet6)],
[edit routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit routing-instances *routing-instance-name* routing-options multicast forwarding-cache (inet | inet6)],
[edit routing-options multicast forwarding-cache],
[edit routing-options multicast forwarding-cache family (inet | inet6)]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.2 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure the suppression, reuse, and warning log message thresholds for multicast forwarding cache limits. You can configure the thresholds globally for the multicast forwarding cache or individually for the IPv4 and IPv6 multicast forwarding caches. Configuring the **threshold** statement globally for the multicast forwarding cache or including the **family** statement to configure the thresholds for the IPv4 and IPv6 multicast forwarding caches are mutually exclusive.

When general forwarding-cache suppression is active, the multicast forwarding-cache prevents forwarding traffic on the shared RP tree (RPT). At the same time, MVPN (*,G) forwarding states are not created for new RPT c-mcast entires, and , (*,G) installed by BGP-MVPN protocol are deleted. When general forwarding-cache suppression ends, BGP-MVPN (*,G) entries are re-added in the RIB and restored to the FIB (up to the MVPN (*,G) limit).

When MVPN RPT suppression is active, for all PE routers in excess of the threshold (including RP PEs), MVPN will not add new (*,G) forwarding entries to the forwarding-cache. Changes are visible once the entries in the current forwarding-cache have timed out or are deleted.

To use **mvpn-rpt-suppress** and/or **mvpn-rpt-reuse**, you must first configure the general **suppress** threshold. If **suppress** is configured but **mvpn-rpt-suppress** is not, both

mvpn-rpt-suppress and **mvpn-rpt-reuse** will inherit *and use* the value set for the general **suppress**.

Options

reuse or **mvpn-rpt-reuse***value* (Optional) Value at which to begin creating new multicast forwarding cache entries. If configured, this number should be less than the **suppress** value.

Range: 1 through 200,000

suppress or **mvpn-rpt-suppress***value* —Value at which to begin suppressing new multicast forwarding cache entries. This value is mandatory. This number should be greater than the **reuse** value.

Range: 1 through 200,000

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Examples: Configuring the Multicast Forwarding Cache on page 928
- **show multicast forwarding-cache statistics on page 1563**

threshold (PIM BFD Detection Time)

Syntax threshold *milliseconds*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection detection-time],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection detection-time]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the threshold for the adaptation of the BFD session detection time. When the detection time adapts to a value equal to or greater than the threshold, a single trap and a single system log message are sent.

NOTE: The threshold value must be equal to or greater than the transmit interval.

The threshold time must be equal to or greater than the value specified in the minimum-interval or the minimum-receive-interval statement.

Options ***milliseconds***—Value for the detection time adaptation threshold.
Range: 1 through 255,000

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- bfd-liveness-detection on page 984
- detection-time on page 1003
- minimum-interval on page 1147
- minimum-receive-interval on page 1152

threshold (PIM BFD Transmit Interval)

Syntax threshold *milliseconds*;

Hierarchy Level [edit protocols pim interface *interface-name* bfd-liveness-detection transmit-interval],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection transmit-interval]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the threshold for the adaptation of the BFD session transmit interval. When the transmit interval adapts to a value greater than the threshold, a single trap and a single system message are sent.

Options ***milliseconds***—Value for the transmit interval adaptation threshold.
Range: 0 through 4,294,967,295 ($2^{32} - 1$)

NOTE: The threshold value specified in the threshold statement must be greater than the value specified in the minimum-interval statement for the transmit-interval statement.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- bfd-liveness-detection on page 984

threshold (PIM Entries)

Syntax threshold *value*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim sglimit],
[edit logical-systems *logical-system-name* protocols pim sglimit *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim sglimit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim sglimit *family*],
[edit protocols pim sglimit],
[edit protocols pim sglimit *family*],
[edit routing-instances *routing-instance-name* protocols pim sglimit],
[edit routing-instances *routing-instance-name* protocols pim sglimit *family*],
[edit logical-systems *logical-system-name* protocols pim rp group-rp-mapping],
[edit logical-systems *logical-system-name* protocols pim rp group-rp-mapping *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp group-rp-mapping],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp group-rp-mapping *family*],
[edit protocols pim rp group-rp-mapping],
[edit protocols pim rp group-rp-mapping *family*],
[edit routing-instances *routing-instance-name* protocols pim rp group-rp-mapping],
[edit routing-instances *routing-instance-name* protocols pim rp group-rp-mapping *family*],
[edit logical-systems *logical-system-name* protocols pim rp register-limit],
[edit logical-systems *logical-system-name* protocols pim rp register-limit *family*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp register-limit],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp register-limit *family*],
[edit protocols pim rp register-limit],
[edit protocols pim rp register-limit *family*],
[edit routing-instances *routing-instance-name* protocols pim rp register-limit],
[edit routing-instances *routing-instance-name* protocols pim rp register-limit *family*],

Release Information Statement introduced in Junos OS Release 12.2.

Description Configure a threshold at which a warning message is logged when a certain number of PIM entries have been received by the device.

Options *value*—Threshold at which a warning message is logged. This is a percentage of the maximum number of entries accepted by the device as defined with the **maximum** statement. You can apply this threshold to incoming PIM join messages, PIM register messages, and group-to-RP mappings.

For example, if you configure a maximum number of 1,000 incoming group-to-RP mappings, and you configure a threshold value of 90 percent, warning messages are logged in the system log when the device receives 900 group-to-RP mappings. The same formula applies to incoming PIM join messages and PIM register messages if configured with both the **maximum *limit*** and the **threshold *value*** statements.

Default: 1 through 100

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- add new concept and example topic to related topic list.
- clear pim join on page 1416

threshold (Routing Instances)

Syntax

```
threshold {
  group group-address {
    source source-address {
      rate threshold-rate;
    }
  }
}
```

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mdt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt],
[edit routing-instances *routing-instance-name* protocols pim mdt],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt]

Release Information
Statement introduced before Junos OS Release 7.4. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description
Establish a threshold to trigger the automatic creation of a data MDT.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

threshold-rate

Syntax threshold-rate *kbps*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *group-address* wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* source *source-address*]
[edit routing-instances *routing-instance-name* provider-tunnel selective group *address* wildcard-source]
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet wildcard-source],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6 wildcard-source]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify the data threshold required before a new tunnel is created for a dynamic selective point-to-multipoint LSP. This statement is part of the configuration for point-to-multipoint LSPs for MBGP MVPNs and PIM-SSM GRE or RSVP-TE selective provider tunnels.

Options *number*—Specify the data threshold required before a new tunnel is created.

Range: 0 through 1,000,000 kilobits per second. Specifying 0 is equivalent to not including the statement.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- *Configuring PIM-SSM GRE Selective Provider Tunnels*

timeout (Flow Maps)

Syntax timeout (never non-discard-entry-only | *minutes*);

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit logical-systems *logical-system-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit routing-options multicast flow-map *flow-map-name*]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the timeout value for multicast forwarding cache entries associated with the flow map.

Options ***minutes***—Length of time that the forwarding cache entry remains active.

Range: 1 through 720

never non-discard-entry-only—Specify that the forwarding cache entry always remain active. If you omit the **non-discard-entry-only** option, all multicast forwarding entries, including those in forwarding and pruned states, are kept forever. If you include the **non-discard-entry-only** option, entries with forwarding states are kept forever, and entries with pruned states time out.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

timeout (Multicast)

Syntax timeout *minutes* <family (inet | inet6)>;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit logical-systems *logical-system-name* routing-options multicast forwarding-cache],
[edit routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit routing-options multicast forwarding-cache]

Release Information Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure the timeout value for multicast forwarding cache entries.

Options ***minutes***—Length of time that the forwarding cache limit remains active.

Range: 1 through 720

family (inet | inet6)—(Optional) Apply the configured timeout to either IPv4 or IPv6 multicast forwarding cache entries. Configuring the **timeout** statement globally for the multicast forwarding cache or including the **family** statement to configure the timeout value for the IPv4 and IPv6 multicast forwarding caches are mutually exclusive.

Default: By default, the configured timeout applies to both IPv4 and IPv6 multicast forwarding cache entries.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring the Multicast Forwarding Cache on page 929

traceoptions (IGMP Snooping)

Syntax

```
traceoptions {
  file filename <files number> <no-stamp> <replace> <size size> <world-readable |
    no-world-readable>;
  flag flag <flag-modifier>;
}
```

Hierarchy Level [edit protocols igmp-snooping]

Release Information Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement introduced in Junos OS Release 13.2 for the QFX series.

Description Define tracing operations for IGMP snooping.

Default The **traceoptions** feature is disabled by default.

Options

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

files *number*—(Optional) Maximum number of trace files, including the active trace file. When a trace file reaches its maximum size, its contents are archived into a compressed file named ***filename*.0** and the trace file is emptied. When the trace file reaches its maximum size again, the ***filename*.0** archive file is renamed ***filename*.1** and a new ***filename*.0** archive file is created from the contents of the trace file. This process continues until the maximum number of trace files is reached, at which point the system starts overwriting the oldest archive file each time the trace file is archived. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 10 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—All tracing operations.
- **general**—Trace general IGMP snooping protocol events.
- **krt**—Trace communication over routing socket.
- **leave**—Trace leave group messages (IGMPv2 and IGMPv3 only).
- **nexthop**—Trace nexthop-related events.
- **normal**—Trace normal IGMP snooping protocol events. If you do not specify this flag, only unusual or abnormal operations are traced.
- **packets**—Trace all IGMP packets.
- **policy**—Trace policy processing.

- **query**—Trace IGMP membership query messages.
- **report**—Trace membership report messages.
- **route**—Trace routing information.
- **state**—Trace IGMP state transitions.
- **task**—Trace routing protocol task processing.
- **timer**—Trace routing protocol timer processing.
- **vlan**—Trace VLAN-related events.

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers per flag:

- **detail**—Provide detailed trace information
- **disable**—Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.
- **receive**—Packets being received.
- **send**—Packets being transmitted.

no-stamp—(Optional) Omit the timestamp at the beginning of each line in the trace file.

no-world-readable—(Optional) Restrict file access to the user who created the file.

replace—(Optional) Replace an existing trace file if there is one. If you do not include this option, tracing output is appended to an existing trace file.

size ***size*** —(Optional) Maximum size of each trace file, in bytes, kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named **trace-file** reaches its maximum size, it is zipped and renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum size, you also must specify a maximum number of files with the **files** option.

Syntax: ***x*** to specify bytes, ***x*****k** to specify KB, ***x*****m** to specify MB, or ***x*****g** to specify GB

Range: 10240 through 4294967295 bytes

Default: 128 KB

world-readable—(Optional) Allow unrestricted file access.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

traceoptions (Multicast Snooping Options)

Syntax

```
traceoptions {
  file filename<files number> <size size> <world-readable | no-world-readable>;
  flag flag <disable>;
}
```

Hierarchy Level [edit multicast-snooping-options]

Release Information Statement introduced in Junos OS Release 8.5.

Description Set multicast snooping tracing options.

Default Tracing operations are disabled.

Options

disable—(Optional) Disable the tracing operation. One use of this option is to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *name*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks. We recommend that you place multicast snooping tracing output in the file **/var/log/multicast-snooping-log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 1 trace file only

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

The following are the tracing options:

- **all**—All tracing operations
- **config-internal**—Trace configuration internals.
- **general**—Trace general events.
- **normal**—All normal events.

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **parse**—Trace configuration parsing.

- **policy**—Trace policy operations and actions.
- **route**—Trace routing table changes.
- **state**—Trace state transitions.
- **task**—Trace protocol task processing.
- **timer**—Trace protocol task timer processing.

no-world-readable—(Optional) Prevent any user from reading the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB) or megabytes (MB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When the ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Multicast Snooping on page 873
- Example: Configuring Multicast Snooping on page 874
- Enabling Bulk Updates for Multicast Snooping on page 879
- Example: Configuring Multicast Snooping on page 874

traceoptions (PIM Snooping)

Syntax

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit routing-instances *<instance-name>* protocols pim-snooping],
[edit logical-systems *<logical-system-name>* routing-instances *<instance-name>* protocols pim-snooping]

Release Information

Statement introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Statement introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Define tracing operations for PIM snooping.

Default

The **traceoptions** feature is disabled by default.

The default PIM trace options are those inherited from the routing protocol's **traceoptions** statement included at the **[edit routing-options]** hierarchy level.

Options

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**.

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

PIM Snooping Tracing Flags:

- **all**—All tracing operations.
- **general**—Trace general PIM snooping events.
- **hello**—Trace hello packets.
- **join**—Trace join messages.
- **normal**—Trace normal PIM snooping events. If you do not specify this flag, only unusual or abnormal operations are traced.
- **packets**—Trace all PIM packets.
- **policy**—Trace policy processing.
- **prune**—Trace prune messages.
- **route**—Trace routing information.
- **state**—Trace PIM state transitions.
- **task**—Trace PIM protocol task processing.
- **timer**—Trace PIM protocol timer processing.

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers per flag:

- **detail**—Provide detailed trace information.
- **disable**—Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.
- **receive**—Packets being received.
- **send**—Packets being transmitted.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- PIM Snooping for VPLS on page 886

traceoptions (Protocols AMT)

Syntax

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols amt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt],
[edit protocols amt],
[edit routing-instances *routing-instance-name* protocols amt]

Release Information

Statement introduced in Junos OS Release 10.2.

Description

Configure Automatic Multicast Tunneling (AMT) tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place tracing output in the file **igmp-log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you must also include the **size** statement to specify the maximum file size.

Range: 2 through 1000 files

Default: 2 files

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

AMT Tracing Flags

- **errors**—All error conditions
- **packets**—All AMT packets
- **tunnels**—All AMT tunnel-related information

Global Tracing Flags

- **all**—All tracing operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Do not allow users to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size ***size***—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*****.0**. When ***trace-file*** again reaches this size, ***trace-file*****.0** is renamed ***trace-file*****.1** and ***trace-file*** is renamed ***trace-file*****.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also include the **files** statement to specify the maximum number of trace files.

Syntax: ***x*****k** to specify KB, ***x*****m** to specify MB, or ***x*****g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring the AMT Protocol on page 423

traceoptions (Protocols DVMRP)

Syntax

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Statement introduced before Junos OS Release 7.4.

Description

Configure DVMRP tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

Default

The default DVMRP trace options are those inherited from the routing protocols **traceoptions** statement included at the **[edit routing-options]** hierarchy level.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place tracing output in the **dvmrp-log** file.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you must also include the **size** statement to specify the maximum file size.

Range: 2 through 1000 files

Default: 2 files

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

DVMRP Tracing Flags

- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **graft**—Graft messages
- **neighbor**—Neighbor probe messages
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **packets**—All DVMRP packets
- **poison**—Poison-route-reverse packets
- **probe**—Probe packets
- **prune**—Prune messages
- **report**—DVMRP route report packets
- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Do not allow users to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches this size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also include the **files** statement to specify the maximum number of trace files.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Tracing DVMRP Protocol Traffic on page 442

traceoptions (Protocols IGMP)

Syntax

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols igmp],
[edit protocols igmp]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure IGMP tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

To trace the paths of multicast packets, use the **mtrace** command.

Default

The default IGMP trace options are those inherited from the routing protocols **traceoptions** statement included at the **[edit routing-options]** hierarchy level.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place tracing output in the file **igmp-log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you must also include the **size** statement to specify the maximum file size.

Range: 2 through 1000 files

Default: 2 files

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

IGMP Tracing Flags

- **leave**—Leave group messages (for IGMP version 2 only).

- **mtrace**—Mtrace packets. Use the **mtrace** command to troubleshoot the software.
- **packets**—All IGMP packets.
- **query**—IGMP membership query messages, including general and group-specific queries.
- **report**—Membership report messages.

Global Tracing Flags

- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Do not allow users to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches this size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also include the **files** statement to specify the maximum number of trace files.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Tracing IGMP Protocol Traffic on page 47

traceoptions (Protocols IGMP Snooping)

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable> ;
    flag flag (detail | disable | receive | send);
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* bridge-domains *domain-name* protocols igmp-snooping],
[edit logical-systems *logical-system-name* routing-instances instance-name bridge-domains *domain-name* protocols igmp-snooping],
[edit logical-systems *logical-system-name* routing-instances *instance-name* protocols igmp-snooping],
[edit bridge-domains *domain-name* protocols igmp-snooping],
[edit routing-instances *instance-name* bridge-domains *domain-name* protocols igmp-snooping],
[edit routing-instances *instance-name* protocols igmp-snooping]
[edit protocols igmp-snooping vlan]

Release Information

Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description

Define tracing operations for IGMP snooping.

Default

The **traceoptions** feature is disabled by default.

Options

file *filename*—Name of the file to receive the output of the tracing operation. All files are placed in the directory **/var/log**.

files *number*—(Optional) Maximum number of trace files. When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached (***x***k to specify KB, ***x***m to specify MB, or ***x***g to specify gigabytes), at which point the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000

Default: 3 files

flag *flag* —Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements. You can include the following flags:

- **all**—All tracing operations.
- **client-notification**—Trace notifications.
- **general**—Trace general IGMP snooping protocol events.
- **group**—Trace group operations.
- **host-notification**—Trace host notifications.

- **leave**—Trace leave group messages (IGMPv2 only).
- **normal**—Trace normal IGMP snooping protocol events.
- **packets**—Trace all IGMP packets.
- **policy**—Trace policy processing.
- **query**—Trace IGMP membership query messages.
- **report**—Trace membership report messages.
- **route**—Trace routing information.
- **state**—Trace IGMP state transitions.
- **task**—Trace routing protocol task processing.
- **timer**—Trace routing protocol timer processing.

no-world-readable—(Optional) Restrict file access to the user who created the file.

size ***size***—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named **trace-file** reaches its maximum size, it is renamed **trace-file.0**, then **trace-file.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten. If you specify a maximum number of files, you also must specify a maximum file size with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify gigabytes

Range: 10 KB through 1 gigabytes

Default: 128 KB

world-readable—(Optional) Enable unrestricted file access.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring IGMP Snooping Trace Operations on page 113
- Configuring IGMP Snooping on page 104
- Example: Configuring IGMP Snooping on SRX Series Devices on page 114
- IGMP Snooping Overview on page 81
- **igmp-snooping on page 1077**

traceoptions (Protocols MSDP)

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols msdp],
[edit logical-systems *logical-system-name* protocols msdp group *group-name*],
[edit logical-systems *logical-system-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* protocols msdp peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp *group-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols msdp peer *address*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name*],
[edit routing-instances *routing-instance-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *routing-instance-name* protocols msdp peer *address*]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure MSDP tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

Default

The default MSDP trace options are those inherited from the routing protocol's **traceoptions** statement included at the **[edit routing-options]** hierarchy level.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place tracing output in the **msdp-log** file.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and

so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you must also include the **size** statement to specify the maximum file size.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

MSDP Tracing Flags

- **keepalive—**Keepalive messages
- **packets**—All MSDP packets
- **route**—MSDP changes to the routing table
- **source-active**—Source-active packets
- **source-active-request**—Source-active request packets
- **source-active-response**—Source-active response packets

Global Tracing Flags

- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Do not allow any user to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches this size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also include the **files** statement to specify the maximum number of trace files.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Tracing MSDP Protocol Traffic on page 410

traceoptions (Protocols MVPN)

Syntax

```
traceoptions {
  file filename <files number> <size size> <world-readable | no-world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols mvpn],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn],
[edit protocols mvpn],
[edit routing-instances *routing-instance-name* protocols mvpn]

Release Information

Statement introduced in Junos OS Release 8.4.
Support at the **[edit protocols mvpn]** hierarchy level introduced in Junos OS Release 13.3.

Description

Trace traffic flowing through a Multicast BGP (MBGP) MVPN.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name in quotation marks (" ").

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements. You can specify any of the following flags:

- **all**—All multicast VPN tracing options
- **cmcast-join**—Multicast VPN C-multicast join routes
- **error**—Error conditions
- **general**—General events
- **inter-as-ad**—Multicast VPN inter-AS automatic discovery routes
- **intra-as-ad**—Multicast VPN intra-AS automatic discovery routes
- **leaf-ad**—Multicast VPN leaf automatic discovery routes

- **mdt-safi-ad**—Multicast VPN MDT SAFI automatic discovery routes
- **nlri**—Multicast VPN advertisements received or sent by means of the BGP
- **normal**—Normal events
- **policy**—Policy processing
- **route**—Routing information
- **source-active**—Multicast VPN source active routes
- **spmsi-ad**—Multicast VPN SPMSI auto discovery active routes
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing
- **tunnel**—Provider tunnel events
- **umh**—Upstream multicast hop (UMH) events

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the following modifiers:

- **detail**—Provide detailed trace information
- **disable**—Disable the tracing flag
- **receive**—Trace received packets
- **send**—Trace sent packets

no-world-readable—Do not allow any user to read the log file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify kilobytes, ***x*m** to specify megabytes, or ***x*g** to specify gigabytes

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—Allow any user to read the log file.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Tracing MBGP MVPN Traffic and Operations*

traceoptions (Protocols PIM)

Syntax

```
traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* protocols pim],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information

Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Configure PIM tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

Default

The default PIM trace options are those inherited from the routing protocol's **traceoptions** statement included at the **[edit routing-options]** hierarchy level.

Options

disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as **all**.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks. All files are placed in the directory **/var/log**. We recommend that you place tracing output in the **pim-log** file.

files *number*—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**, then ***trace-file*.1**, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you must also include the **size** statement to specify the maximum file size.

Range: 2 through 1000 files

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

PIM Tracing Flags

- **assert**—Assert messages

- **bidirectional-df-election**—Bidirectional PIM designated-forwarder (DF) election events
- **bootstrap**—Bootstrap messages
- **cache**—Packets in the PIM sparse mode routing cache
- **graft**—Graft and graft acknowledgment messages
- **hello**—Hello packets
- **join**—Join messages
- **mt**—Multicast tunnel messages
- **nsr-synchronization**—Nonstop active routing (NSR) synchronization messages
- **packets**—All PIM packets
- **prune**—Prune messages
- **register**—Register and register stop messages
- **rp**—Candidate RP advertisements
- **all**—All tracing operations
- **general**—A combination of the **normal** and **route** trace operations
- **normal**—All normal operations

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **policy**—Policy operations and actions
- **route**—Routing table changes
- **state**—State transitions
- **task**—Interface transactions and processing
- **timer**—Timer usage

flag-modifier—(Optional) Modifier for the tracing flag. You can specify one or more of these modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Do not allow users to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches this size, it is renamed ***trace-file*.0**. When ***trace-file*** again reaches this size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also include the **files** statement to specify the maximum number of trace files.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 0 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Configuring PIM Trace Options on page 192
- Tracing DVMRP Protocol Traffic on page 442
- Tracing MSDP Protocol Traffic on page 410
- Configuring PIM Trace Options on page 192

transmit-interval (PIM BFD Liveness Detection)

Syntax

```
transmit-interval {
  minimum-interval milliseconds;
  threshold milliseconds;
}
```

Hierarchy Level

[edit protocols pim interface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information

Statement introduced in Junos OS Release 8.2.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Support for BFD authentication introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Specify the transmit interval for the **bfd-liveness-detection** statement. The negotiated transmit interval for a peer is the interval between the sending of BFD packets to peers. The receive interval for a peer is the minimum interval between receiving packets sent from its peer; the receive interval is not negotiated between peers. To determine the transmit interval, each peer compares its configured minimum transmit interval with its peer's minimum receive interval. The larger of the two numbers is accepted as the transmit interval for that peer.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194
- bfd-liveness-detection on page 984
- threshold on page 1324
- minimum-interval on page 1148
- minimum-receive-interval on page 1152

tunnel-devices (Protocols AMT)

Syntax tunnel-devices [ud-*fpc/pic/port*];

Hierarchy Level [edit logical-systems *logical-system-name* protocols amt relay],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay],
[edit protocols amt relay],
[edit routing-instances *routing-instance-name* protocols amt relay]

Release Information Statement introduced in Junos OS Release 13.2.

Description List one or more tunnel-capable Automatic Multicast Tunneling (AMT) PICs to be used for creating multicast tunnel (**ud**) interfaces. Creating an AMT PIC list enables you to control the load-balancing implementation.

Tunnel-capable PICs include DPC and MPC.

The physical position of the PIC in the routing device determines the multicast tunnel interface name.

Default Multicast tunnel interfaces are created on all available tunnel-capable AMT PICs, based on a round-robin algorithm.

Options **ud-*fpc/pic/port***—Interface that is automatically generated when a tunnel-capable PIC is installed in the routing device.

NOTE: Each **tunnel-devices** statement keyword is optional. By default, all configured tunnel devices are used. The keyword selects the subset of configured tunnel devices.

Tunnel devices must be configured on MX Series routers. They are not automatically available like M Series routers that have dedicated PICs. On MX Series routers, the tunnel device port is the next highest number after the physical ports – a PIC created with the **tunnel-services** statement at the **[edit chassis fpc *slot-number* pic *number*]** hierarchy level.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423
- Example: Configuring the AMT Protocol on page 428

tunnel-devices (Tunnel-Capable PICs)

Syntax tunnel-devices [mt-*fpc/pic/port*];

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *instance-name* protocols pim], [edit routing-instances *instance-name* protocols pim]

Release Information Statement introduced in Junos OS Release 10.2.
Statement introduced in Junos OS Release 10.2 for EX Series switches.

Description List one or more tunnel-capable PICs to be used for creating multicast tunnel (**mt**) interfaces. Creating a PIC list enables you to control the load-balancing implementation.

Tunnel-capable PICs include:

- Adaptive Services PIC
- Multiservices PIC or Multiservices DPC
- Tunnel Services PIC
- On MX Series routers, a PIC created with the **tunnel-services** statement at the **[edit chassis fpc *slot-number* pic *number*]** hierarchy level.

The physical position of the PIC in the routing device determines the multicast tunnel interface name. For example, if you have an Adaptive Services PIC installed in FPC slot 0 and PIC slot 0, the corresponding multicast tunnel interface name is **mt-0/0/0**. The same is true for Tunnel Services PICs, Multiservices PICs, and Multiservices DPCs.

Default Multicast tunnel interfaces are created on all available tunnel-capable PICs, based on a round-robin algorithm.

Options **mt-*fpc/pic/port***—Interface that is automatically generated when a tunnel-capable PIC is installed in the routing device.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Load Balancing Multicast Tunnel Interfaces Among Available PICs on page 459

tunnel-limit (Protocols AMT)

Syntax tunnel-limit *number*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols amt relay],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols amt relay],
[edit protocols amt relay],
[edit routing-instances *routing-instance-name* protocols amt relay]

Release Information Statement introduced in Junos OS Release 10.2.

Description Limit the number of Automatic Multicast Tunneling (AMT) data tunnels created. The system might reach a dynamic upper limit of tunnels of all types before the static AMT limit is reached.

Options *number*—Maximum number of data AMTs that can be created on the system.

Range: 0 through 4294967295

Default: 1 tunnel

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring the AMT Protocol on page 423

tunnel-limit (Routing Instances)

Syntax tunnel-limit *limit*;

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mdt],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt],
[edit routing-instances *routing-instance-name* protocols pim mdt],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6* mdt]

Release Information Statement introduced before Junos OS Release 7.4. In Junos OS Release 17.3R1, the **mdt** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default MDT in Rosen 7, and data MDT for Rosen 6 and Rosen 7. The **provider-tunnel mdt** hierarchy is now hidden for backward compatibility with existing scripts.

Description Limit the number of data MDTs created in this VRF instance. If the limit is 0, then no data MDTs are created for this VRF instance.

Options *limit*—Maximum number of data MDTs for this VRF instance.

Range: **0** through **1024**

Default: **0** (No data MDTs are created for this VRF instance.)

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

tunnel-limit (Routing Instances Provider Tunnel Selective)

Syntax tunnel-limit *number*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

Release Information Statement introduced in Junos OS Release 8.5.

Description Specify a limit on the number of selective tunnels that can be created for an LSP. This limit can be applied to the following types of selective tunnels:

- Ingress replication tunnels
- LDP-signaled LSP
- LDP point-to-multipoint LSP
- PIM-SSM provider tunnel
- RSVP-signaled LSP
- RSVP-signaled point-to-multipoint LSP

Options *number*—Specify the tunnel limit.

Range: 0 through 1024

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring Point-to-Multipoint LSPs for an MBGP MVPN*
- selective on page 1276
- wildcard-source on page 1385

tunnel-source

Syntax tunnel-source *address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel family *inet* | *inet6*pim-ssm],

Release Information Statement introduced in Junos OS Release 10.1.
In Junos OS Release 17.3R1, the **pim-ssm** hierarchy was moved from **provider-tunnel** to the **provider-tunnel family inet** and **provider-tunnel family inet6** hierarchies as part of an upgrade to add IPv6 support for default multicast distribution tree (MDT) in Rosen 7, and data MDT for Rosen 6 and Rosen 7.

Description Configure the source address for the provider space multipoint generic router encapsulation (mGRE) tunnel. This statement enables a VPN tunnel source for Rosen 6 or Rosen 7 multicast VPNs. .

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- group-address (Routing Instances) on page 1048

unicast (Route Target Community)

Syntax

```
unicast {
    receiver;
    sender;
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn route-target import-target],
[edit routing-instances *routing-instance-name* protocols mvpn route-target import-target]

Release Information

Statement introduced in Junos OS Release 8.4.

Description

Specify the same target community configured for unicast.

Options

receiver—Specify the unicast target community used when importing receiver site routes.

sender—Specify the unicast target community used when importing sender site routes.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Configuring VRF Route Targets for Routing Instances for an MBGP MVPN*

unicast (Virtual Tunnel in Routing Instances)

Syntax unicast;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* interface vt-*fpc/pic/port.unit-number*],
[edit routing-instances *routing-instance-name* interface vt-*fpc/pic/port.unit-number*]

Release Information Statement introduced in Junos OS Release 9.4.

Description In a multiprotocol BGP (MBGP) multicast VPN (MVPN), configure the virtual tunnel (VT) interface to be used for unicast traffic only.

Default If you omit this statement, the VT interface can be used for both multicast and unicast traffic.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 744
- Example: Configuring MBGP MVPN Extranets on page 669

unicast-umh-election

Syntax unicast-umh-election;

Hierarchy Level [edit routing-instances *routing-instance-name* protocols mvpn]

Release Information Statement introduced in Junos OS Release 9.6.

Description Configure a router to use the unicast route preference to determine the single forwarder election.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN on page 624
- mvpn (NG-MVPN) on page 1181

upstream-interface

Syntax upstream-interface [*interface-names*];

Hierarchy Level
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast pim-to-igmp-proxy],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* routing-options multicast pim-to-mld-proxy],
[edit logical-systems *logical-system-name* routing-options multicast pim-to-igmp-proxy],
[edit logical-systems *logical-system-name* routing-options multicast pim-to-mld-proxy],
[edit routing-instances *routing-instance-name* routing-options multicast pim-to-igmp-proxy],
[edit routing-instances *routing-instance-name* routing-options multicast pim-to-mld-proxy],
[edit routing-options multicast pim-to-igmp-proxy],
[edit routing-options multicast pim-to-mld-proxy]

Release Information
Statement introduced in Junos OS Release 9.6.
Statement introduced in Junos OS Release 9.6 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Configure at least one, but not more than two, upstream interfaces on the rendezvous point (RP) routing device that resides between a customer edge–facing Protocol Independent Multicast (PIM) domain and a core-facing PIM domain. The RP routing device translates PIM join or prune messages into corresponding IGMP report or leave messages (if you include the **pim-to-igmp-proxy** statement), or into corresponding MLD report or leave messages (if you include the **pim-to-mld-proxy** statement). The routing device then proxies the IGMP or MLD report or leave messages to one or both upstream interfaces to forward IPv4 multicast traffic (for IGMP) or IPv6 multicast traffic (for MLD) across the PIM domains.

Options
interface-names—Names of one or two upstream interfaces to which the RP routing device proxies IGMP or MLD report or leave messages for transmission of multicast traffic across PIM domains. You can specify a maximum of two upstream interfaces on the RP routing device. To configure a set of two upstream interfaces, specify the full interface names, including all physical and logical address components, within square brackets ([ ]).

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- Configuring PIM-to-IGMP Message Translation on page 386
- Configuring PIM-to-MLD Message Translation on page 387

use-p2mp-lsp

Syntax

```
igmp-snooping-options {
    use-p2mp-lsp;
  }
}
```

Hierarchy Level [edit routing-instances *instance name* igmp-snooping-options]

Release Information Statement introduced in Junos OS Release 13.3.

Description Point-to-multipoint LSP for IGMP snooping enables multicast data traffic in the core to take the point-to-multipoint path. The effect is a reduction in the amount of traffic generated on the PE router when sending multicast packets for multiple VPLS sessions because it avoids the need to send multiple parallel streams when forwarding multicast traffic to PE routers participating in the VPLS. Note that the options configured for IGMP snooping are applied on a per-routing-instance so all IGMP snooping routes in the same instance will use the same mode, point to multipoint or pseudowire.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Point-to-Multipoint LSP with IGMP Snooping on page 120
- show igmp snooping options on page 1488
- multicast-snooping-options on page 1174

version (Protocols BFD)

Syntax version (0 | 1 | automatic);

Hierarchy Level [edit protocols piminterface *interface-name* bfd-liveness-detection],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name* bfd-liveness-detection]

Release Information Statement introduced in Junos OS Release 8.1.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the bidirectional forwarding detection (BFD) protocol version that you want to detect.

Options Configure the BFD version to detect: **1** (BFD version 1) or **automatic** (autodetect the BFD version)

Default: automatic

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring BFD for PIM on page 194

version (Protocols PIM)

Syntax version *version*;

Hierarchy Level
[edit logical-systems *logical-system-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* protocols pim rp static address *address*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim rp static address *address*],
[edit protocols pim interface *interface-name*],
[edit protocols pim rp static address *address*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim rp static address *address*]

Release Information
Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Statement deprecated (hidden) in Junos OS Release 16.1 for later removal.

Description Starting in Junos OS Release 16.1, it is no longer necessary to specify a PIM version. PIMv1 is being obsoleted so the version choice is moot.

Options ***version***—PIM version number.

Range: See the Description, above.

Default: PIMv2 for both rendezvous point (RP) mode (at the **[edit protocols pim rp static address *address*]** hierarchy level). and interface mode (at the **[edit protocols pim interface *interface-name*]** hierarchy level).

Required Privilege Level
routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Enabling PIM Sparse Mode on page 217
- Configuring PIM Dense Mode Properties on page 205
- Configuring PIM Sparse-Dense Mode Properties on page 207

version (Protocols IGMP)

Syntax version *version*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]
[edit protocols igmp-snooping vlan (all | *vlan-name*)]

Release Information Statement introduced before Junos OS Release 7.4.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Specify the version of IGMP.

Options ***version***—IGMP version number.

Range: 1, 2, or 3

Default: IGMP version 2

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Changing the IGMP Version on page 36

version (Protocols IGMP AMT)

Syntax version *version*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols igmp amt relay defaults],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols igmp amt relay defaults],
[edit protocols igmp amt relay defaults],
[edit routing-instances *routing-instance-name* protocols igmp amt relay defaults]

Release Information Statement introduced in Junos OS Release 10.2.

Description Specify the version of IGMP used through an Automatic Multicast Tunneling (AMT) interface.

Options *version*—IGMP version number.

Range: 1, 2, or 3

Default: IGMP version 3

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring Default IGMP Parameters for AMT Interfaces on page 425

version (Protocols MLD)

Syntax version *version*;

Hierarchy Level [edit logical-systems *logical-system-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced before Junos OS Release 7.4.

Description Configure the MLD version explicitly. MLD version 2 (MLDv2) is used only to support source-specific multicast (SSM).

Options ***version***—MLD version to run on the interface.

Range: 1 or 2

Default: 1 (MLDv1)

Required Privilege Level routing and trace—To view this statement in the configuration.
routing-control and trace-control—To add this statement to the configuration.

Related Documentation

- Modifying the MLD Version on page 56

vrf-advertise-selective

Syntax

```
vrf-advertise-selective {
  family {
    inet-mvpn;
    inet6-mvpn;
  }
}
```

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name*], [edit routing-instances *routing-instance-name*]

Release Information Statement introduced in Junos OS Release 10.1.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

Description Explicitly enable IPv4 or IPv6 MVPN routes to be advertised from the VRF instance while preventing all other route types from being advertised.

If you configure the **vrf-advertise-selective** statement without any of its options, the router or switch has the same behavior as if you configured the **no-vrf-advertise** statement. All VPN routes are prevented from being advertised from a VRF routing instance to the remote PE routers. This behavior is useful for hub-and-spoke configurations, enabling you to configure a PE router to not advertise VPN routes from the primary (hub) instance. Instead, these routes are advertised from the secondary (downstream) instance.

The options are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *Limiting Routes to Be Advertised by an MVPN VRF Instance*
- *no-vrf-advertise*

vlan (Bridge Domains)

Syntax

```
vlan vlan-id {
  all
  immediate-leave;
  interface interface-name {
    group-limit limit;
    host-only-interface;
    multicast-router-interface;
    static {
      group multicast-group-address {
        source ip-address;
      }
    }
  }
  proxy {
    source-address ip-address;
  }
  query-interval seconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
}
```

Hierarchy Level

[edit bridge-domains *bridge-domain-name* protocols igmp-snooping],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name* protocols igmp-snooping]

Release Information

Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 13.2 for the QFX series.

Description

Configure IGMP snooping parameters for a particular VLAN.

Default

By default, IGMP snooping options apply to all VLANs.

Options

vlan-id—Apply the parameters to this VLAN.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring VLAN-Specific IGMP Snooping Parameters on page 105
- igmp-snooping on page 1077

vlan (IGMP Snooping)

List of Syntax

Syntax (EX Series and SRX-series:SRX 210)

```
vlan (all | vlan-name) {
  data-forwarding {
    source {
      groups group-prefix;
    }
    receiver {
      source-vlans vlan-list;
      install;
    }
  }
  disable;
  immediate-leave;
  interface (all | interface-name) {
    multicast-router-interface;
    static {
      group ip-address;
    }
  }
  proxy {
    source-address ip-address;
  }
  robust-count number;
  version number;
}
```

Syntax (QFX Series, QFabric and EX4600)

```
vlan vlan-name {
  immediate-leave;
  interface interface-name {
    multicast-router-interface;
    static {
      group multicast-ip-address;
    }
  }
  version number;
}
```

Syntax (QFX Series, EX4600 and NFX Series Devices)

```
vlan vlan-name {
  immediate-leave;
  interface interface-name {
    group-limit limit;
    host-only-interface;
    multicast-router-interface;
    static {
      group multicast-group-address {
        source ip-address;
      }
    }
```

```
    }
    l2-querier {
        source-address ip-address;
        }
    qualified-vlan;
    proxy {
        source-address ip-address;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
}
```

Hierarchy Level [edit protocols igmp-snooping]

Release Information Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.1 for EX Series switches.
Statement updated with enhanced **?** (CLI completion feature) functionality in Junos OS Release 9.5 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure IGMP snooping parameters for a VLAN.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type ? after vlan or vlans in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range. For IGMP snooping, secondary private VLANs are not listed.

When the **vlan** configuration statement is used without the **disable** statement, IGMP snooping is enabled on the specified VLAN or on all VLANs.

NOTE: You cannot configure IGMP snooping on a secondary VLAN.

TIP: To display a list of all configured VLANs on the system, including VLANs that are configured but not committed, type ? after vlan or vlans in your configuration mode command line. Note that only one VLAN is displayed for a VLAN range. For IGMP snooping, secondary private VLANs are not listed.

Default By default, IGMP snooping options apply to all VLANs.

IGMP snooping options apply to the specified VLAN.

If the **vlan** statement is not included in the configuration, IGMP snooping is disabled.

Options

- **all**—All VLANs on the switch
- ***vlan-name***—Name of a VLAN.

TIP: When you configure IGMP snooping parameters using the **vlan all** statement, any VLAN that is not individually configured for IGMP snooping inherits the **vlan all** configuration. Any VLAN that is individually configured for IGMP snooping, on the other hand, inherits none of its configuration from **vlan all**. Any parameters that are not explicitly defined for the individual VLAN assume their default values, not the values specified in the **vlan all** configuration.

For example, in the following configuration:

```
protocols {
    igmp-snooping {
        vlan all {
            robust-count 8;
        }
        vlan employee {
            interface ge-0/0/8.0 {
                static {
                    group 239.0.10.3
                }
            }
        }
    }
}
```

all VLANs, except **employee**, have a robust count of 8. Because **employee** has been individually configured, its robust count value is not determined by the value set under **vlan all**. Instead, its robust count is the default value of 2.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring IGMP Snooping on Switches on page 93
- Example: Configuring IGMP Snooping on EX Series Switches on page 90
- Configuring IGMP Snooping on Switches on page 88
- *Configuring IGMP Snooping (CLI Procedure)*
- *show igmp-snooping vlans*
- **show igmp-snooping vlans on page 1494**
- Configuring VLAN-Specific IGMP Snooping Parameters on page 105
- *igmp-snooping*

vlan (MLD Snooping)

Syntax

```
vlan (all | vlan-name) {
  disable;
  immediate-leave;
  interface ( all | interface-name) {
    group-limit limit;
    host-only-interface;
    immediate-leave;
    multicast-router-interface;
    static {
      group ip-address {
        source ip-address;
      }
    }
  }
  qualified-vlan;
  query-intervalseconds;
  query-last-member-interval seconds;
  query-response-interval seconds;
  robust-count number;
  traceoptions {
    file filename <files number> <size size> <world-readable | no-world-readable>;
    flag flag <flag-modifier>;
  }
  version version;
}
```

Hierarchy Level

[edit protocols mld-snooping]
[edit routing-instances *instance-name* protocols mld-snooping]

Release Information

Statement introduced in Junos OS Release 12.1 for EX Series switches.
Support at the **[edit routing-instances *instance-name* protocols mld-snooping]** hierarchy introduced in Junos OS Release 13.3 for EX Series switches.
Support for the **qualified-vlan**, **query-interval**, **query-last-member-interval**, **query-response-interval**, and **traceoptions** statements introduced in Junos OS Release 13.3 for EX Series switches.

Description

Configure MLD snooping parameters for a VLAN.

When the **vlan** configuration statement is used without the **disable** statement, MLD snooping is enabled on the specified VLAN or on all VLANs.

Default

If the **vlan** statement is not included in the configuration, MLD snooping is disabled.

Options

all—(All EX Series switches except EX9200) Configure MLD snooping parameters for all VLANs on the switch.

vlan-name—Configure MLD snooping parameters for the specified VLAN.

TIP: When you configure MLD snooping parameters using the vlan all statement, any VLAN that is not individually configured for MLD snooping inherits the vlan all configuration. Any VLAN that is individually configured for MLD snooping, on the other hand, inherits none of its configuration from vlan all. Any parameters that are not explicitly defined for the individual VLAN assume their default values, not the values specified in the vlan all configuration.

For example, in the following configuration:

```
protocols {
    mld-snooping {
        vlan all {
            robust-count 8;
        }
        vlan employee {
            interface ge-0/0/8.0 {
                static {
                    group ff1e::1;
                }
            }
        }
    }
}
```

all VLANs, except employee, have a robust count of 8. Because employee has been individually configured, its robust count value is not determined by the value set under vlan all. Instead, its robust count is the default value of 2.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

vlan (PIM Snooping)

Syntax

vlan <*vlan-id*>{
no-dr-flood;
}

Hierarchy Level

[edit routing-instances <*instance-name*> protocols pim-snooping],
[edit logical-systems <*logical-system-name*> routing-instances <*instance-name*> protocols pim-snooping]

Release Information

Statement introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Statement introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Configure PIM snooping parameters for a VLAN.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- PIM Overview on page 185
- *Configuring Basic PIM Settings*

vpn-group-address

Syntax vpn-group-address *address*;

Hierarchy Level [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced before Junos OS Release 7.4.
Use group-address in place of **vpn-group-address**.
Starting with Junos OS Release 11.4, to provide consistency with draft-rosen 7 and next-generation BGP-based multicast VPNs, configure the provider tunnels for draft-rosen 6 anysource multicast VPNs at the **[edit routing-instances *routing-instance-name* provider-tunnel]** hierarchy level. The **mdt**, **vpn-tunnel-source**, and **vpn-group-address** statements are deprecated at the **[edit routing-instances *routing-instance-name* protocols pim]** hierarchy level.

Description Configure the group address for the Layer 3 VPN in the service provider's network.

Options ***address***—Address for the Layer 3 VPN in the service provider's network.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation
- *Configuring Multicast Layer 3 VPNs*
- *Multicast Protocols Feature Guide*

wildcard-group-inet

Syntax

```
wildcard-group-inet {
  wildcard-source {
    inter-region-segmented{
      fan-out fan-out value;
    }
    ldp-p2mp;
    pim-ssm {
      group-range multicast-prefix;
    }
    rsvp-te {
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
      static-lsp lsp-name;
    }
    threshold-rate number;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

Release Information

Statement introduced in Junos OS Release 10.0.
The **inter-region-segmented** statement added in Junos OS Release 15.1.

Description

Configure a wildcard group matching any group IPv4 address.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- wildcard-group-inet6 on page 1383
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668
- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 662
- Configuring a Selective Provider Tunnel Using Wildcards on page 667

wildcard-group-inet6

Syntax

```
wildcard-group-inet6 {
  wildcard-source {
    inter-region-segmented{
      fan-out fan-out value;
    }
    ldp-p2mp;
    pim-ssm {
      group-range multicast-prefix;
    }
    rsvp-te {
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
      static-lsp lsp-name;
    }
    threshold-rate number;
  }
}
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

Release Information

Statement introduced in Junos OS Release 10.0.
The **inter-region-segmented** statement added in Junos OS Release 15.1.

Description

Configure a wildcard group matching any group IPv6 address.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- wildcard-group-inet on page 1382
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668
- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 662
- Configuring a Selective Provider Tunnel Using Wildcards on page 667

wildcard-source (PIM RPF Selection)

Syntax

```
wildcard-source {
  next-hop next-hop-address;
}
```

Hierarchy Level

[edit routing-instances *routing-instance-name* protocols pim rpf-selection group *group-address*],
[edit routing-instances *routing-instance-name* protocols pim rpf-selection prefix-list *prefix-list-addresses*]

Release Information

Statement introduced in Junos OS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.
Statement introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Use a wildcard for the multicast source instead of (or in addition to) a specific multicast source.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

view-level—To view this statement in the configuration.
control-level—To add this statement to the configuration.

Related Documentation

- Example: Configuring PIM RPF Selection on page 816

wildcard-source (Selective Provider Tunnels)

Syntax

```
wildcard-source {
  inter-region-segmented {
    fan-out fan-out value;
  }
ldp-p2mp;
  pim-ssm {
    group-range multicast-prefix;
  }
  rsvp-te {
    label-switched-path-template {
      (default-template | lsp-template-name);
    }
    static-lsp lsp-name;
  }
```

Hierarchy Level

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective group *group-prefix*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6],
[edit routing-instances *routing-instance-name* provider-tunnel selective group *group-prefix*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet],
[edit routing-instances *routing-instance-name* provider-tunnel selective wildcard-group-inet6]

Release Information

Statement introduced in Junos OS Release 10.0.
The **inter-region-segmented** statement added in Junos OS Release 15.1.

Description

Configure a selective provider tunnel for a shared tree using a wildcard source.

The remaining statements are explained separately. See CLI Explorer.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- wildcard-group-inet on page 1382
- wildcard-group-inet6 on page 1383
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 668
- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 662
- Configuring a Selective Provider Tunnel Using Wildcards on page 667

CHAPTER 28

Operational Commands

- show amt statistics
- show amt summary
- show amt tunnel
- show bgp group
- show configuration protocols igmp
- show dvmrp interfaces
- show dvmrp neighbors
- show dvmrp prefix
- show dvmrp prunes
- show igmp interface
- show igmp group
- show igmp snooping interface
- show igmp snooping membership
- show igmp snooping options
- show igmp snooping statistics
- show igmp-snooping vlans
- show ingress-replication mvpn
- show interfaces (Multicast Tunnel)
- show mld group
- show mld interface
- show mld statistics
- show mld snooping interface
- show mld-snooping membership
- show mld-snooping route
- show mld-snooping statistics
- show mld-snooping vlans
- show mpls lsp
- show msdp
- show msdp source
- show msdp source-active
- show msdp statistics
- show multicast backup-pe-groups
- show multicast flow-map
- show multicast forwarding-cache statistics
- show multicast interface
- show multicast mrinfo
- show multicast next-hops

- show multicast pim-to-igmp-proxy
- show multicast pim-to-mld-proxy
- show multicast route
- show multicast rpf
- show multicast scope
- show multicast sessions
- show multicast snooping next-hops
- show multicast snooping route
- show multicast statistics
- show multicast usage
- show mvpn c-multicast
- show mvpn instance
- show mvpn neighbor
- show mvpn suppressed
- show policy
- show pim bidirectional df-election
- show pim bidirectional df-election interface
- show pim bootstrap
- show pim interfaces
- show pim join
- show pim neighbors
- show pim snooping interfaces
- show pim snooping join
- show pim snooping neighbors
- show pim snooping statistics
- show pim rps
- show pim source
- show pim statistics
- show pim mdt
- show pim mdt data-mdt-joins
- show pim mdt data-mdt-limit
- show pim mvpn
- show route forwarding-table
- show route label
- show route snooping
- show route table

- show sap listen
- test msdp

clear amt statistics

Syntax clear amt statistics
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information Command introduced in JUNOS Release 10.2.

Description Clear Automatic Multicast Tunneling (AMT) statistics.

Options **none**—Clear the multicast statistics for all AMT tunnel interfaces.

instance *instance-name*—(Optional) Clear AMT multicast statistics for the specified instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level clear

Related Documentation
- show amt statistics on page 1441

List of Sample Output clear amt statistics on page 1391

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear amt statistics

```
user@host> clear amt statistics
```

clear amt tunnel

Syntax

clear amt tunnel
<gateway *gateway-ip-addr*> <port *port-number*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<statistics>
<tunnel-interface *interface-name*>

Release Information

Command introduced in JUNOS Release 10.2.

Description

Clear the Automatic Multicast Tunneling (AMT) multicast state. Optionally, clear AMT protocol statistics.

Options

none—Clear multicast state for all AMT tunnel interfaces.

gateway *gateway-ip-addr* port *port-number*—(Optional) Clear the AMT multicast state for the specified gateway address. If no port is specified, clear the AMT multicast state for all AMT gateways with the given IP address.

instance *instance-name*—(Optional) Clear the AMT multicast state for the specified instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

statistics—(Optional) Clear multicast statistics for all AMT tunnels or for specified tunnels.

tunnel-interface *interface-name*—(Optional) Clear the AMT multicast state for the specified AMT tunnel interface.

Required Privilege Level

clear

Related Documentation

- show amt tunnel on page 1446

List of Sample Output

clear amt tunnel on page 1392
clear amt tunnel statistics gateway-address on page 1393

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear amt tunnel

```
user@host> clear amt tunnel
```

clear amt tunnel statistics gateway-address

```
user@host> clear amt tunnel statistics gateway-address 100.31.1.21 port 4000
```

clear igmp membership

List of Syntax
Syntax on page 1394
Syntax (EX Series Switch and the QFX Series) on page 1394

Syntax
clear igmp membership
<all>
<group *address-range*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
clear igmp membership
<group *address-range*>
<interface *interface-name*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Clear Internet Group Management Protocol (IGMP) group members.

Options
all—Clear IGMP members for groups and interfaces in the master instance.

group *address-range*—(Optional) Clear all IGMP members that are in a particular address range. An example of a range is **233.252/16**. If you omit the destination prefix length, the default is **/32**.

interface *interface-name*—(Optional) Clear all IGMP group members on an interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
clear

Related Documentation
- show igmp group on page 1474
- show igmp interface on page 1470

List of Sample Output
clear igmp membership all on page 1395
clear igmp membership interface on page 1395
clear igmp membership group on page 1396

Output Fields
See show igmp group for an explanation of output fields.

Sample Output

clear igmp membership all

The following sample output displays IGMP group information before and after the **clear igmp membership** command is entered:

```
user@host> show igmp group
Interface        Group           Last Reported   Timeout
so-0/0/0         198.51.100.253   203.0.113.1          186
so-0/0/0         198.51.100.254   203.0.113.1          186
so-0/0/0         198.51.100.255   203.0.113.1          187
so-0/0/0         198.51.100.240   203.0.113.1          188
local            198.51.100.6     (null)                 0
local            198.51.100.5     (null)                 0
local            198.51.100.25    (null)                 0
local            198.51.100.22    (null)                 0
local            198.51.100.2     (null)                 0
local            198.51.100.13    (null)                 0

user@host> clear igmp membership all
Clearing Group Membership Info for so-0/0/0
Clearing Group Membership Info for so-1/0/0
Clearing Group Membership Info for so-2/0/0

user@host> show igmp group
Interface        Group           Last Reported   Timeout
local            198.51.100.6            (null)                 0
local            198.51.100.5            (null)                 0
local            198.51.100.254          (null)                 0
local            198.51.100.255          (null)                 0
local            198.51.100.2            (null)                 0
local            198.51.100.13           (null)                 0
```

clear igmp membership interface

The following sample output displays IGMP group information before and after the **clear igmp membership interface** command is issued:

```
user@host> show igmp group
Interface        Group           Last Reported   Timeout
so-0/0/0         198.51.100.253   203.0.113.1          210
so-0/0/0         198.51.100.200   203.0.113.1          210
so-0/0/0         198.51.100.255   203.0.113.1          215
so-0/0/0         198.51.100.254   203.0.113.1          216
local            198.51.100.6     (null)                 0
local            198.51.100.5     (null)                 0
local            198.51.100.254   (null)                 0
local            198.51.100.255   (null)                 0
local            198.51.100.2     (null)                 0
local            198.51.100.13    (null)                 0

user@host> clear igmp membership interface so-0/0/0
Clearing Group Membership Info for so-0/0/0

user@host> show igmp group
```

```
Interface        Group           Last Reported   Timeout
local            198.51.100.6       (null)               0
local            198.51.100.5       (null)               0
local            198.51.100.254     (null)               0
local            198.51.100.255     (null)               0
local            198.51.100.2       (null)               0
local            198.51.100.13      (null)               0
```

clear igmp membership group

The following sample output displays IGMP group information before and after the **clear igmp membership group** command is entered:

```
user@host> show igmp group
Interface        Group           Last Reported   Timeout
so-0/0/0         198.51.100.253   203.0.113.1          210
so-0/0/0         198.51.100.25    203.0.113.1          210
so-0/0/0         198.51.100.255   203.0.113.1          215
so-0/0/0         198.51.100.254   203.0.113.1          216
local            198.51.100.6     (null)                 0
local            198.51.100.5     (null)                 0
local            198.51.100.254   (null)                 0
local            198.51.100.25    (null)                 0
local            198.51.100.2     (null)                 0
local            198.51.100.13    (null)                 0

user@host> clear igmp membership group 233.252/16
Clearing Group Membership Range 198.51.100.0/16 on so-0/0/0
Clearing Group Membership Range 198.51.100.0/16 on so-1/0/0
Clearing Group Membership Range 198.51.100.0/16 on so-2/0/0

user@host> show igmp group
Interface        Group           Last Reported   Timeout
so-0/0/0         198.51.100.255   203.0.113.1          231
so-0/0/0         198.51.100.254   203.0.113.1          233
so-0/0/0         198.51.100.253   203.0.113.1          236
local            198.51.100.6     (null)                 0
local            198.51.100.5     (null)                 0
local            198.51.100.254   (null)                 0
local            198.51.100.255   (null)                 0
local            198.51.100.2     (null)                 0
local            198.51.100.13    (null)                 0
```

clear igmp snooping membership

Syntax

clear igmp snooping membership
<vlan *vlan-name*>
<group | source *address*>
<instance *instance-name*>
<interface *interface-name*>
<learning-domain *learning-domain-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-identifier*>

Release Information

Command introduced in Junos OS Release 8.5.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description

Clear IGMP snooping dynamic membership information from the multicast forwarding table.

Options

none—Clear IGMP snooping membership for all supported address families on all interfaces.

vlan *vlan-name* —(Optional) Clear dynamic membership information for the specified VLAN.

group | source *address*—(Optional) Clear IGMP snooping membership for the specified multicast group or source address.

instance *instance-name*—(Optional) Clear IGMP snooping membership for the specified instance.

interface *interface-name*—(Optional) Clear IGMP snooping membership on a specific interface.

learning-domain *learning-domain-name*—(Optional) Perform this operation on all learning domains or on a particular learning domain.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or for all logical systems.

vlan-id *vlan-identifier*—(Optional) Perform this operation on a particular VLAN.

Required Privilege Level

clear

Related Documentation

- show igmp snooping membership on page 1483

List of Sample Output

clear igmp snooping membership on page 1398

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear igmp snooping membership

```
user@host> clear igmp snooping membership
```

clear igmp snooping statistics

Syntax
clear igmp snooping statistics
<instance *instance-name*>
<interface *interface-name*>
<learning-domain (all | *learning-domain-name*)>
<logical-system *logical-system-name*>

Release Information
Command introduced in Junos OS Release 8.5.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description
Clear IP IGMP snooping statistics.

Options
none—Clear IGMP snooping statistics for all supported address families on all interfaces.

instance *instance-name*—(Optional) Clear IGMP snooping statistics for the specified instance.

interface *interface-name*—(Optional) Clear IGMP snooping statistics on a specific interface.

learning-domain (all | *learning-domain-name*)—(Optional) Perform this operation on all learning domains or on a particular learning domain.

logical-system *logical-system-name*—(Optional) Delete the IGMP snooping statistics for a given logical system or for all logical systems.

Required Privilege Level
clear

Related Documentation
- show igmp snooping statistics on page 1489

List of Sample Output
clear igmp snooping statistics on page 1399

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear igmp snooping statistics

```
user@host> clear igmp snooping statistics
```

clear igmp statistics

List of Syntax Syntax on page 1400
Syntax (EX Series Switches) on page 1400

Syntax

clear igmp statistics
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switches)

clear igmp statistics
<interface *interface-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear Internet Group Management Protocol (IGMP) statistics.

Options

none—Clear IGMP statistics on all interfaces.

interface *interface-name*—(Optional) Clear IGMP statistics for the specified interface only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

clear

Related Documentation

- *show igmp statistics*

List of Sample Output clear igmp statistics on page 1400

Output Fields

See *show igmp statistics* for an explanation of output fields.

Sample Output

clear igmp statistics

The following sample output displays IGMP statistics information before and after the **clear igmp statistics** command is entered:

```
user@host> show igmp statistics
IGMP packet statistics for all interfaces
IGMP Message type      Received       Sent  Rx errors
Membership Query           8883        459          0
V1 Membership Report          0          0          0
```

```
DVMRP                     19784      35476          0
PIM V1                    18310          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0
IGMP v3 unsupported type                             0
IGMP v3 source required for SSM                      0
IGMP v3 mode not applicable for SSM                  0

IGMP Global Statistics
Bad Length                    0
Bad Checksum                  0
Bad Receive If                0
Rx non-local               1227


user@host> clear igmp statistics
user@host> show igmp statistics
IGMP packet statistics for all interfaces
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          0          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0
IGMP v3 unsupported type                             0
IGMP v3 source required for SSM                      0
IGMP v3 mode not applicable for SSM                  0
IGMP Global Statistics
Bad Length                    0
Bad Checksum                  0
Bad Receive If                0
Rx non-local                  0
```

clear mld membership

Syntax clear mld membership
<all>
<group *group-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Release Information Command introduced before Junos OS Release 7.4.

Description Clear Multicast Listener Discovery (MLD) group membership.

Options

all—Clear MLD memberships for groups and interfaces in the master instance.

group *group-name*—(Optional) Clear MLD membership for the specified group.

interface *interface-name*—(Optional) Clear MLD group membership for the specified interface.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level view

Related Documentation

- show mld group on page 1503

List of Sample Output clear mld membership all on page 1402

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear mld membership all

```
user@host> clear mld membership all
```

clear mld-snooping membership

Syntax **clear mld-snooping membership**
<vlan *vlan-name*>

Release Information Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description Clear MLD snooping dynamic membership information from the multicast forwarding table.

Options

NOTE: If your EX Series switch CLI includes more options than what appears in this section, see the *clear mld snooping membership* command summary.

none—Clear dynamic membership information for all VLANs.

vlan *vlan-name*—(Optional) Clear dynamic membership information for the specified VLAN.

Required Privilege Level view

Related Documentation

- Understanding MLD Snooping on page 125
- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- mld-snooping on page 1155
- show mld-snooping membership on page 1517
- clear mld-snooping statistics on page 1404

List of Sample Output clear mld-snooping membership vlan employee-vlan on page 1403

Sample Output

clear mld-snooping membership vlan employee-vlan

```
user@host> clear mld-snooping membership vlan employee-vlan
```

clear mld-snooping statistics

Syntax **clear mld-snooping statistics**

Release Information Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description Clear MLD snooping statistics.

Required Privilege Level view

Related Documentation

- Understanding MLD Snooping on page 125
- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- mld-snooping on page 1155
- show mld-snooping statistics on page 1523
- clear mld-snooping membership on page 1403

List of Sample Output clear mld-snooping statistics on page 1404

Sample Output

clear mld-snooping statistics

```
user@host> clear mld-snooping statistics
```

clear mld statistics

Syntax clear mld statistics
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Release Information Command introduced before Junos OS Release 7.4.

Description Clear Multicast Listener Discovery (MLD) statistics.

Options **none**—(Same as **logical-system all**) Clear MLD statistics for all interfaces.

interface *interface-name*—(Optional) Clear MLD statistics for the specified interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level clear

Related Documentation
- show mld statistics on page 1511

List of Sample Output clear mld statistics on page 1405

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear mld statistics

```
user@host> clear mld statistics
```

clear msdp cache

Syntax

clear msdp cache
<all>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<peer *peer-address*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear the entries in the Multicast Source Discovery Protocol (MSDP) source-active cache.

Options

all— Clear all MSDP source-active cache entries in the master instance.

instance *instance-name*—(Optional) Clear entries for a specific MSDP instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

peer *peer-address*—(Optional) Clear the MSDP source-active cache entries learned from a specific peer.

Required Privilege Level

clear

Related Documentation

- show msdp source-active on page 1552

List of Sample Output

clear msdp cache all on page 1406

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear msdp cache all

```
user@host> clear msdp cache all
```

clear msdp statistics

Syntax
clear msdp statistics
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<peer *peer-address*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Clear Multicast Source Discovery Protocol (MSDP) peer statistics.

Options
none—Clear MSDP statistics for all peers.

instance *instance-name*—(Optional) Clear statistics for the specified instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

peer *peer-address*—(Optional) Clear the statistics for the specified peer.

Required Privilege Level
clear

Related Documentation
- show msdp statistics on page 1555

List of Sample Output
clear msdp statistics on page 1407

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear msdp statistics

```
user@host> clear msdp statistics
```

clear multicast bandwidth-admission

Syntax clear multicast bandwidth-admission
<group *group-address*>
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>
<source *source-address*>

Release Information Command introduced in Junos OS Release 8.3.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Reapply IP multicast bandwidth admissions.

Options **none**—Reapply multicast bandwidth admissions for all IPv4 forwarding entries in the master routing instance.

group *group-address*—(Optional) Reapply multicast bandwidth admissions for the specified group.

inet—(Optional) Reapply multicast bandwidth admission settings for IPv4 flows.

inet6—(Optional) Reapply multicast bandwidth admission settings for IPv6 flows.

instance *instance-name*—(Optional) Reapply multicast bandwidth admission settings for the specified instance. If you do not specify an instance, the command applies to the master routing instance.

interface *interface-name*—(Optional) Examines the corresponding outbound interface in the relevant entries and acts as follows:

- If the interface is congested, and it was admitted previously, it is removed.
- If the interface was rejected previously, the **clear multicast bandwidth-admission** command enables the interface to be admitted as long as enough bandwidth exists on the interface.
- If you do not specify an interface, issuing the **clear multicast bandwidth-admission** command readmits any previously rejected interface for the relevant entries as long as enough bandwidth exists on the interface.

To manually reject previously admitted outbound interfaces, you must specify the interface.

source *source-address*—(Optional) Use with the **group** option to reapply multicast bandwidth admission settings for the specified (source, group) entry.

Required Privilege Level clear

Related Documentation
- show multicast interface on page 1565

List of Sample Output clear multicast bandwidth-admission on page 1409

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear multicast bandwidth-admission

```
user@host> clear multicast bandwidth-admission
```

clear multicast forwarding-cache

Syntax

clear multicast forwarding-cache
<all>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced in Junos OS Release 12.2.

Description

Clear IP multicast forwarding cache entries.

This command is not supported for next-generation multiprotocol BGP multicast VPNs (MVPNs).

Options

all—Clear all multicast forwarding cache entries in the master instance.

inet—(Optional) Clear multicast forwarding cache entries for IPv4 family addresses.

inet6—(Optional) Clear multicast forwarding cache entries for IPv6 family addresses.

instance *instance-name*—(Optional) Clear multicast forwarding cache entries on a specific routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

clear

Related Documentation

- show multicast forwarding-cache statistics on page 1563

List of Sample Output

clear multicast forwarding-cache all on page 1410

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear multicast forwarding-cache all

```
user@host> clear multicast forwarding-cache all
```

clear multicast scope

List of Syntax
Syntax on page 1411
Syntax (EX Series Switch and the QFX Series) on page 1411

Syntax

clear multicast scope
<inet | inet6>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

clear multicast scope
<inet | inet6>
<interface *interface-name*>

Release Information

Command introduced in Junos OS Release 7.6.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 option introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear IP multicast scope statistics.

Options

none—(Same as **logical-system all**) Clear multicast scope statistics.

inet—(Optional) Clear multicast scope statistics for IPv4 family addresses.

inet6—(Optional) Clear multicast scope statistics for IPv6 family addresses.

interface *interface-name*—(Optional) Clear multicast scope statistics on a specific interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

clear

Related Documentation

- show multicast scope on page 1593

List of Sample Output

clear multicast scope on page 1412

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear multicast scope

```
user@host> clear multicast scope
```

clear multicast sessions

List of Syntax
Syntax on page 1413
Syntax (EX Series Switch and the QFX Series) on page 1413

Syntax
clear multicast sessions
<logical-system (all | *logical-system-name*)>
<*regular-expression*>

Syntax (EX Series Switch and the QFX Series)
clear multicast sessions
<*regular-expression*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Clear IP multicast sessions.

Options
none—(Same as **logical-system all**) Clear multicast sessions.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

regular-expression—(Optional) Clear only multicast sessions that contain the specified regular expression.

Required Privilege Level
clear

Related Documentation
- show multicast sessions on page 1595

List of Sample Output
clear multicast sessions on page 1413

Output Fields
When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear multicast sessions

```
user@host> clear multicast sessions
```

clear multicast statistics

List of Syntax
Syntax on page 1414
Syntax (EX Series Switch and the QFX Series) on page 1414

Syntax

clear multicast statistics
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

clear multicast statistics
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear IP multicast statistics.

Options

none—Clear multicast statistics for all supported address families on all interfaces.

inet—(Optional) Clear multicast statistics for IPv4 family addresses.

inet6—(Optional) Clear multicast statistics for IPv6 family addresses.

instance *instance-name*—(Optional) Clear multicast statistics for the specified instance.

interface *interface-name*—(Optional) Clear multicast statistics on a specific interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

clear

Related Documentation

- show multicast statistics on page 1606

List of Sample Output

clear multicast statistics on page 1415

Output Fields

When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear multicast statistics

```
user@host> clear multicast statistics
```

clear pim join

List of Syntax

Syntax on page 1416
Syntax (EX Series Switch and the QFX Series) on page 1416

Syntax

clear pim join
<all>
<*group-address*>
<bidirectional | dense | sparse>
<exact>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<rp *ip-address/prefix* | source *ip-address/prefix*>
<sg | star-g>

Syntax (EX Series Switch and the QFX Series)

clear pim join
<all>
<*group-address*>
<dense | sparse>
<exact>
<inet | inet6>
<instance *instance-name*>
<rp *ip-address/prefix* | source *ip-address/prefix*>
<sg | star-g>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Multiple new filter options introduced in Junos OS Release 13.2.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear the Protocol Independent Multicast (PIM) join and prune states.

Options

all—To clear PIM join and prune states for all groups and family addresses in the master instance, you must specify “all”..

group-address—(Optional) Clear the PIM join and prune states for a group address.

bidirectional | dense | sparse—(Optional) Clear PIM bidirectional mode, dense mode, or sparse and source-specific multicast (SSM) mode entries.

exact—(Optional) Clear only the group that exactly matches the specified group address.

inet | inet6—(Optional) Clear the PIM entries for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Clear the entries for a specific PIM-enabled routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

rp *ip-address/prefix* | source *ip-address/prefix*—(Optional) Clear the PIM entries with a specified rendezvous point (RP) address and prefix or with a specified source address and prefix. You can omit the prefix.

sg | star-g—(Optional) Clear PIM (S,G) or (*,G) entries.

Additional Information The **clear pim join** command cannot be used to clear the PIM join and prune state on a backup Routing Engine when nonstop active routing is enabled.

Required Privilege Level clear

Related Documentation
- show pim join on page 1642

List of Sample Output
clear pim join all on page 1417
clear pim join inet6 all on page 1417
clear pim join inet6 star-g all on page 1417

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear pim join all

```
user@host> clear pim join all
Cleared 8 Join/Prune states
```

clear pim join inet6 all

```
user@host> clear pim join inet6 all
Cleared 4 Join/Prune states
```

clear pim join inet6 star-g all

```
user@host> clear pim join inet6 star-g all
Cleared 1 Join/Prune states
```

clear pim join-distribution

Syntax

clear pim join-distribution
<all>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced in Junos OS Release 10.0.

Description

Clear the PIM **join-redistribute** states.

Use the **show pim source** command to find out if there are multiple paths available for a source (for example, an RP).

When you include the **join-load-balance** statement in the configuration, the PIM join states are distributed evenly on available equal-cost multipath links. When an upstream neighbor link fails, Junos OS redistributes the PIM join states to the remaining links. However, when new links are added or the failed link is restored, the existing PIM joins are not redistributed to the new link. New flows will be distributed to the new links. However, in a network without new joins and prunes, the new link is not used for multicast traffic. The **clear pim join-distribution** command redistributes the existing flows to the new upstream neighbors. Redistributing the existing flows causes traffic to be disrupted, so we recommend that you run the **clear pim join-distribution** command during a maintenance window.

Options

all— (Optional) Clear the PIM **join-redistribute** states for all groups and family addresses in the master instance.

none— Automatically clear all PIM join/prune states.

instance *instance-name*—(Optional) Redistribute the join states for a specific PIM-enabled routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Additional Information

The **clear pim join-distribution** command cannot be used to redistribute the PIM join states on a backup Routing Engine when nonstop active routing is enabled.

Required Privilege Level

clear

Related Documentation

- show pim neighbors on page 1657
- show pim join on page 1642
- join-load-balance on page 1116

List of Sample Output clear pim join-distribution all on page 1419

Output Fields When you enter this command, you are provided no feedback on the status of your request. You can enter the **show pim join** command before and after distributing the join state to verify the operation.

Sample Output

clear pim join-distribution all

```
user@host> clear pim join-distribution all
```

clear pim register

List of Syntax

Syntax on page 1420
Syntax (EX Series Switch and the QFX Series) on page 1420
Syntax (PTX Series) on page 1420

Syntax

clear pim register
<all>
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

clear pim register
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>

Syntax (PTX Series)

clear pim register
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced in Junos OS Release 7.6.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear Protocol Independent Multicast (PIM) register message counters.

Options

all—Required to clear the PIM register message counters for all groups and family addresses in the master instance.

inet | inet6—(Optional) Clear PIM register message counters for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Clear register message counters for a specific PIM-enabled routing instance.

interface *interface-name*—(Optional) Clear PIM register message counters for a specific interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Additional Information

The **clear pim register** command cannot be used to clear the PIM register state on a backup Routing Engine when nonstop active routing is enabled.

Required Privilege Level clear

Related Documentation
- show pim statistics on page 1689

List of Sample Output clear pim register all on page 1421

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

clear pim register all

```
user@host> clear pim register all
```

clear pim snooping join

Syntax

clear pim snooping join
<instance *instance-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-id*>

Release Information

Command introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Command introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Clear information about Protocol Independent Multicast (PIM) snooping joins.

Options

none—Display detailed information.

instance *instance-name*—(Optional) Clear PIM snooping join information for the specified routing instance.

logical-system *logical-system-name*—(Optional) Delete the IGMP snooping statistics for a given logical system or for all logical systems.

vlan-id *vlan-identifier*—(Optional) Clear PIM snooping join information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- PIM Snooping for VPLS on page 886

List of Sample Output

clear pim snooping join on page 1422

Output Fields

See show pim snooping join for an explanation of the output fields.

Sample Output

clear pim snooping join

The following sample output displays information about PIM snooping joins before and after the **clear pim snooping join** command is entered:

```
user@host> show pim snooping join extensive
Instance: vpls1
Learning-Domain: vlan-id 10
Learning-Domain: vlan-id 20

Group: 198.51.100.2
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 192.0.2.5, port: ge-1/3/7.20
Downstream port: ge-1/3/1.20
```

```
Downstream neighbors:
192.0.2.2 State: Join Flags: SRW Timeout: 185

Group: 198.51.100.3
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 192.0.2.4, port: ge-1/3/5.20
Downstream port: ge-1/3/3.20
Downstream neighbors:
192.0.2.3 State: Join Flags: SRW Timeout: 175
```

```
user@host> clear pim snooping join
Clearing the Join/Prune state for 203.0.113.0/24
Clearing the Join/Prune state for 203.0.113.0/24
```

```
user@host> show pim snooping join extensive
Instance: vpls1
Learning-Domain: vlan-id 10
Learning-Domain: vlan-id 20
```

clear pim snooping statistics

Syntax clear pim snooping statistics
<instance *instance-name*>
<interface *interface-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-id*>

Release Information Command introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Command introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description Clear Protocol Independent Multicast (PIM) snooping statistics.

Options **none**—Clear PIM snooping statistics for all family addresses, instances, and interfaces.

instance *instance-name*—(Optional) Clear statistics for a specific PIM-snooping-enabled routing instance.

interface *interface-name*—(Optional) Clear PIM snooping statistics for a specific interface.

logical-system *logical-system-name*—(Optional) Delete the IGMP snooping statistics for a given logical system or for all logical systems.

vlan-id *vlan-identifier*—(Optional) Clear PIM snooping statistics information for the specified VLAN.

Required Privilege Level clear

Related Documentation
- PIM Snooping for VPLS on page 886

List of Sample Output clear pim snooping statistics on page 1424

Output Fields See show pim snooping statistics for an explanation of the output fields.

Sample Output

clear pim snooping statistics

The following sample output displays PIM snooping statistics before and after the **clear pim snooping statistics** command is entered:

```
user@host> show pim snooping statistics
Instance: vpls1
Learning-Domain: vlan-id 10

Tx J/P messages 0
RX J/P messages 660
Rx J/P messages -- seen 0
```

```
Rx J/P messages -- received 660
Rx Hello messages 1396
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0

Learning-Domain: vlan-id 20
```

```
user@host> clear pim snooping statistics
user@host> show pim snooping statistics
Instance: vpls1
Learning-Domain: vlan-id 10

Tx J/P messages 0
RX J/P messages 0
Rx J/P messages -- seen 0
Rx J/P messages -- received 0
Rx Hello messages 0
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0

Learning-Domain: vlan-id 20
```

clear pim statistics

List of Syntax
Syntax on page 1426
Syntax (EX Series Switch and the QFX Series) on page 1426

Syntax

clear pim statistics
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

clear pim statistics
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Clear Protocol Independent Multicast (PIM) statistics.

Options

none—Clear PIM statistics for all family addresses, instances, and interfaces.

inet | inet6—(Optional) Clear PIM statistics for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Clear statistics for a specific PIM-enabled routing instance.

interface *interface-name*—(Optional) Clear PIM statistics for a specific interface.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Additional Information

The **clear pim statistics** command cannot be used to clear the PIM statistics on a backup Routing Engine when nonstop active routing is enabled.

Required Privilege Level

clear

Related Documentation

- show pim statistics on page 1689

List of Sample Output

clear pim statistics on page 1427

Output Fields

See show pim statistics for an explanation of output fields.

Sample Output

clear pim statistics

The following sample output displays PIM statistics before and after the **clear pim statistics** command is entered:

```
user@host> show pim statistics
PIM statistics on all interfaces:
PIM Message type       Received       Sent  Rx errors
Hello                         0          0          0
Register                      0          0          0
Register Stop                 0          0          0
Join Prune                    0          0          0
Bootstrap                     0          0          0
Assert                        0          0          0
Graft                         0          0          0
Graft Ack                     0          0          0
Candidate RP                  0          0          0
V1 Query                   2111       4222          0
V1 Register                   0          0          0
V1 Register Stop              0          0          0
V1 Join Prune             14200      13115          0
V1 RP Reachability            0          0          0
V1 Assert                     0          0          0
V1 Graft                      0          0          0
V1 Graft Ack                  0          0          0
PIM statistics summary for all interfaces:
Unknown type                              0
V1 Unknown type                           0
Unknown Version                           0
Neighbor unknown                          0
Bad Length                                0
Bad Checksum                              0
Bad Receive If                            0
Rx Intf disabled                       2007
Rx V1 Require V2                          0
Rx Register not RP                        0
RP Filtered Source                        0
Unknown Reg Stop                          0
Rx Join/Prune no state                 1040
Rx Graft/Graft Ack no state               0
...

user@host> clear pim statistics
user@host> show pim statistics
PIM statistics on all interfaces:
PIM Message type       Received       Sent  Rx errors
Hello                         0          0          0
Register                      0          0          0
Register Stop                 0          0          0
Join Prune                    0          0          0
Bootstrap                     0          0          0
Assert                        0          0          0
Graft                         0          0          0
Graft Ack                     0          0          0
Candidate RP                  0          0          0
V1 Query                      1          0          0
```

```
V1 Register                    0          0          0
...
```

mtrace

Syntax

mtrace *source*
<logical-system *logical-system-name*>
<routing-instance *routing-instance-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 9.5 for SRX1400, SRX3400, SRX3600, SRX5600, and SRX5800 devices.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 12.3 for the PTX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display trace information about an IP multicast path.

Options

source—Source hostname or address.

logical-system (*logical-system-name*)—(Optional) Perform this operation on a logical system.

routing-instance *routing-instance-name*—(Optional) Trace a particular routing instance.

Additional Information

The **mtrace** command for multicast traffic is similar to the **traceroute** command used for unicast traffic. Unlike **traceroute**, **mtrace** traces traffic backwards, from the receiver to the source.

Required Privilege Level

view

List of Sample Output

mtrace source on page 1431

Output Fields

Table 32 on page 1429 describes the output fields for the **mtrace** command. Output fields are listed in the approximate order in which they appear.

Table 32: mtrace Output Fields

Field Name	Field Description
Mtrace from	IP address of the receiver.
to	IP address of the source.
via group	IP address of the multicast group (if any).
Querying full reverse path	Indicates the full reverse path query has begun.
number-of-hops	Number of hops from the source to the named router or switch.

Table 32: mtrace Output Fields (continued)

Field Name	Field Description
router-name	Name of the router or switch for this hop.
address	Address of the router or switch for this hop.
protocol	Protocol used (for example, PIM).
Round trip time	Average round-trip time, in milliseconds (ms).
total ttl of	Time-to-live (TTL) threshold.

Sample Output

mtrace source

```
user@host> mtrace 192.168.4.2
Mtrace from 192.168.4.2 to 192.168.1.2 via group 0.0.0.0
Querying full reverse path... * *
  0  routerA.lab.mycompany.net (192.168.1.2)
 -1  routerB.lab.mycompany.net (192.168.2.2)  PIM  thresh^ 1
 -2  routerC.lab.mycompany.net (192.168.3.2)  PIM  thresh^ 1
 -3  hostA.lab.mycompany.net (192.168.4.2)
Round trip time 2 ms; total ttl of 2 required.
```

mtrace from-source

Syntax

mtrace from-source source *source*
<brief | detail>
<extra-hops *extra-hops*>
<group *group*>
<interval *interval*>
<loop>
<max-hops *max-hops*>
<max-queries *max-queries*>
<multicast-response | unicast-response>
<no-resolve>
<no-router-alert>
<response *response*>
<routing-instance *routing-instance-name*>
<ttl *ttl*>
<wait-time *wait-time*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display trace information about an IP multicast path from a source to this router or switch. If you specify a group address with this command, Junos OS returns additional information, such as packet rates and losses.

Options

brief | detail—(Optional) Display the specified level of output.

extra-hops *extra-hops*—(Optional) Number of hops to take after reaching a nonresponsive router. You can specify a number between **0** and **255**.

group *group*—(Optional) Group address for which to trace the path. The default group address is **0.0.0.0**.

interval *interval*—(Optional) Number of seconds to wait before gathering statistics again. The default value is **10** seconds.

loop—(Optional) Loop indefinitely, displaying rate and loss statistics.

max-hops *max-hops*—(Optional) Maximum hops to trace toward the source. The range of values is **0** through **255**. The default value is **32** hops.

max-queries *max-queries*—(Optional) Maximum number of query attempts for any hop. The range of values is **1** through **32**. The default is **3**.

multicast-response—(Optional) Always request the response using multicast.

no-resolve—(Optional) Do not attempt to display addresses symbolically.

no-router-alert—(Optional) Do not use the router-alert IP option.

response ***response***—(Optional) Send trace response to a host or multicast address.

routing-instance ***routing-instance-name***—(Optional) Trace a particular routing instance.

source ***source***—Source hostname or address.

ttl ***ttl***—(Optional) IP time-to-live (TTL) value. You can specify a number between **0** and **255**. Local queries to the multicast group use a value of **1**. Otherwise, the default value is **127**.

unicast-response—(Optional) Always request the response using unicast.

wait-time ***wait-time***—(Optional) Number of seconds to wait for a response. The default value is **3**.

Required Privilege Level view

List of Sample Output mtrace from-source on page 1434

Output Fields Table 33 on page 1433 describes the output fields for the **mtrace from-source** command. Output fields are listed in the approximate order in which they appear.

Table 33: mtrace from-source Output Fields

Field Name	Field Description
Mtrace from	IP address of the receiver.
to	IP address of the source.
via group	IP address of the multicast group (if any).
Querying full reverse path	Indicates the full reverse path query has begun.
number-of-hops	Number of hops from the source to the named router or switch.
router-name	Name of the router or switch for this hop.
address	Address of the router or switch for this hop.
protocol	Protocol used (for example, PIM).
Round trip time	Average round-trip time, in milliseconds (ms).
total ttl of	Time-to-live (TTL) threshold.
source	Source address.
Response Dest	Response destination address.

Table 33: mtrace from-source Output Fields (continued)

Field Name	Field Description
Overall	Average packet rate for all traffic at each hop.
Packet Statistics for Traffic From	Number of packets lost, number of packets sent, percentage of packets lost, and average packet rate at each hop.
Receiver	IP address receiving the multicast.
Query source	IP address sending the mtrace query.

Sample Output

mtrace from-source

```
user@host> mtrace from-source source 192.168.4.2 group 233.252.0.1
Mtrace from 192.168.4.2 to 192.168.1.2 via group 233.252.0.1
Querying full reverse path... * *
  0  routerA.lab.mycompany.net (192.168.1.2)
 -1  routerB.lab.mycompany.net (192.168.2.2)  PIM  thresh^ 1
 -2  routerC.lab.mycompany.net (192.168.3.2)  PIM  thresh^ 1
 -3  hostA.lab.mycompany.net (192.168.4.2)
Round trip time 2 ms; total ttl of 2 required.

Waiting to accumulate statistics...Results after 10 seconds:

Source        Response Dest    Overall     Packet Statistics For Traffic From
192.168.4.2 192.168.1.2    Packet      192.168.4.2 To 233.252.0.1
     v       __/  rtt    2 ms     Rate       Lost/Sent = Pct  Rate
192.168.2.1
192.168.3.2  routerC.lab.mycompany.net
     v     ^      ttl    2                     0/0    = --    0 pps
192.168.4.1
192.168.2.2   routerB.lab.mycompany.net
     v      \__   ttl    3                     ?/0            0 pps
192.168.1.2   192.168.1.2
  Receiver      Query Source
```

mtrace monitor

Syntax mtrace monitor

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.

Description Listen passively for IP multicast responses. To exit the **mtrace monitor** command, type Ctrl+c.

Options **none**—Trace the master instance.

Required Privilege Level view

List of Sample Output mtrace monitor on page 1436

Output Fields Table 34 on page 1435 describes the output fields for the **mtrace monitor** command. Output fields are listed in the approximate order in which they appear.

Table 34: mtrace monitor Output Fields

Field Name	Field Description
Mtrace query at	Date and time of the query.
by	Address of the host issuing the query.
resp to	Response destination.
qid	Query ID number.
packet from...to	IP address of the query source and default group destination.
from...to	IP address of the multicast source and the response address.
via group	IP address of the group to trace.
mxhop	Maximum hop setting.

Sample Output

mtrace monitor

```
user@host> mtrace monitor
Mtrace query at Oct 22 13:36:14 by 192.168.3.2, resp to 233.252.0.32, qid 74a5b8
packet from 192.168.3.2 to 233.252.0.2
from 192.168.3.2 to 192.168.3.38 via group 233.252.0.1 (mxhop=60)

Mtrace query at Oct 22 13:36:17 by 192.681.3.2, resp to 233.252.0.32, qid 1d07ba
packet from 192.168.3.2 to 233.252.0.2
from 192.168.3.2 to 192.168.3.38 via group 233.252.0.1 (mxhop=60)

Mtrace query at Oct 22 13:36:20 by 192.681.3.2, resp to same, qid 2fea1d
packet from 192.168.3.2 to 233.252.0.2
from 192.168.3.2 to 192.168.3.38 via group 233.252.0.1 (mxhop=60)

Mtrace query at Oct 22 13:36:30 by 192.168.3.2, resp to same, qid 7c88ad
packet from 192.168.3.2 to 233.252.0.2
from 192.168.3.2 to 192.168.3.38 via group 233.252.0.1 (mxhop=60)
```

mtrace to-gateway

Syntax

mtrace to-gateway gateway *gateway*
<brief | detail>
<extra-hops *extra-hops*>
<group *group*>
<interface *interface-name*>
<interval *interval*>
<loop>
<max-hops *max-hops*>
<max-queries *max-queries*>
<multicast-response | unicast-response>
<no-resolve>
<no-router-alert>
<response *response*>
<routing-instance *routing-instance-name*>
<ttl *ttl*>
<unicast-response>
<wait-time *wait-time*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display trace information about a multicast path from this router or switch to a gateway router or switch.

Options

gateway *gateway*—Send the trace query to a gateway multicast address.

brief | detail—(Optional) Display the specified level of output.

extra-hops *extra-hops*—(Optional) Number of hops to take after reaching a nonresponsive router or switch. You can specify a number between **0** and **255**.

group *group*—(Optional) Group address for which to trace the path. The default group address is **0.0.0.0**.

interface *interface-name*—(Optional) Source address for sending the trace query.

interval *interval*—(Optional) Number of seconds to wait before gathering statistics again. The default value is **10**.

loop—(Optional) Loop indefinitely, displaying rate and loss statistics.

max-hops *max-hops*—(Optional) Maximum hops to trace toward the source. You can specify a number between **0** and **255**.. The default value is **32**.

max-queries *max-queries*—(Optional) Maximum number of query attempts for any hop. You can specify a number between **0** and **255**. The default value is **3**.

multicast-response—(Optional) Always request the response using multicast.

no-resolve—(Optional) Do not attempt to display addresses symbolically.

no-router-alert—(Optional) Do not use the router-alert IP option.

response ***response***—(Optional) Send trace response to a host or multicast address.

routing-instance ***routing-instance-name***—(Optional) Trace a particular routing instance.

ttl ***ttl***—(Optional) IP time-to-live value. You can specify a number between **0** and **225**. Local queries to the multicast group use TTL **1**. Otherwise, the default value is **127**.

unicast-response—(Optional) Always request the response using unicast.

wait-time ***wait-time***—(Optional) Number of seconds to wait for a response. The default value is **3**.

Required Privilege Level view

List of Sample Output mtrace to-gateway on page 1439

Output Fields Table 35 on page 1438 describes the output fields for the **mtrace to-gateway** command. Output fields are listed in the approximate order in which they appear.

Table 35: mtrace to-gateway Output Fields

Field Name	Field Description
Mtrace from	IP address of the receiver.
to	IP address of the source.
via group	IP address of the multicast group (if any).
Querying full reverse path	Indicates the full reverse path query has begun.
number-of-hops	Number of hops from the source to the named router or switch.
router-name	Name of the router or switch for this hop.
address	Address of the router or switch for this hop.
protocol	Protocol used (for example, PIM).
Round trip time	Average round-trip time, in milliseconds (ms).
total ttl of	Time-to-live (TTL) threshold.

Sample Output

mtrace to-gateway

```
user@host> mtrace to-gateway gateway 192.168.3.2 group 233.252.0.1 interface 192.168.1.73
brief

Mtrace from 192.168.1.73 to 192.168.1.2 via group 233.252.0.1
Querying full reverse path... * *
  0  routerA.lab.mycompany.net (192.1.1.2)
 -1  routerA.lab.mycompany.net (192.1.1.2)  PIM  thresh^ 1
 -2  routerB.lab.mycompany.net (192.1.2.2)  PIM  thresh^ 1
 -3  routerC.lab.mycompany.net (192.1.3.2)  PIM  thresh^ 1
Round trip time 2 ms; total ttl of 3 required.
```

request pim multicast-tunnel rebalance

List of Syntax
Syntax on page 1440
Syntax (EX Series Switches) on page 1440

Syntax
request pim multicast-tunnel rebalance
<instance *instance-name*>
<logical-system (all | *logical-system-name*>

Syntax (EX Series Switches)
request pim multicast-tunnel rebalance
<instance *instance-name*>

Release Information
Command introduced in Junos OS Release 10.2.
Command introduced in Junos OS Release 10.2 for EX Series switches.

Description
Rebalance the assignment of multicast tunnel encapsulation interfaces across available tunnel-capable PICs or across a configured list of tunnel-capable PICs. You can determine whether a rebalance is necessary by running the **show pim interfaces instance** ***instance-name*** command.

Options
none—Re-create and rebalance all tunnel interfaces for all routing instances.

instance ***instance-name***—Re-create and rebalance all tunnel interfaces for a specific instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
maintenance

Related Documentation
- show pim interfaces on page 1639
- Load Balancing Multicast Tunnel Interfaces Among Available PICs on page 459

Output Fields
This command produces no output. To verify the operation of the command, run the **show pim interface instance** ***instance-name*** before and after running the **request pim multicast-tunnel rebalance** command.

show amt statistics

Syntax
show amt statistics
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information
Command introduced in JUNOS Release 10.2.

Description
Display information about the Automatic Multicast Tunneling (AMT) protocol tunnel statistics.

Options
none—Display summary information about all AMT Protocol tunnels.

instance *instance-name*—(Optional) Display information for the specified instance only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

Related Documentation
- clear amt statistics on page 1391
- show amt summary on page 1444
- show amt tunnel on page 1446

List of Sample Output
show amt statistics on page 1442

Output Fields
Table 36 on page 1441 describes the output fields for the **show amt statistics** command. Output fields are listed in the approximate order in which they appear.

Table 36: show amt statistics Output Fields

Field Name	Field Description
AMT receive message count	Summary of AMT statistics for messages received on all interfaces. • **AMT relay discovery**—Number of AMT relay discovery messages received. • **AMT membership request**—Number of AMT membership request messages received. • **AMT membership update**—Number of AMT membership update messages received.
AMT send message count	Summary of AMT statistics for messages sent on all interfaces. • **AMT relay advertisement**—Number of AMT relay advertisement messages sent. • **AMT membership query**—Number of AMT membership query messages sent.

Table 36: show amt statistics Output Fields (continued)

Field Name	Field Description
AMT error message count	Summary of AMT statistics for error messages received on all interfaces. • **AMT incomplete packet**—Number of messages received with length errors so severe that further classification could not occur. • **AMT invalid mac**—Number of messages received with an invalid message authentication code (MAC). • **AMT unexpected type**—Number of messages received with an unknown message type specified. • **AMT invalid relay discovery address**—Number of AMT relay discovery messages received with an address other than the configured anycast address. • **AMT invalid membership request address**—Number of AMT membership request messages received with an address other than the configured AMT local address. • **AMT invalid membership update address**—Number of AMT membership update messages received with an address other than the configured AMT local address. • **AMT incomplete relay discovery messages**—Number of AMT relay discovery messages received that are not fully formed. • **AMT incomplete membership request messages**—Number of AMT membership request messages received that are not fully formed. • **AMT incomplete membership update messages**—Number of AMT membership update messages received that are not fully formed. • **AMT no active gateway**—Number of AMT membership update messages received for a tunnel that does not exist for the gateway that sent the message. • **AMT invalid inner header checksum**—Number of AMT membership update messages received with an invalid IP checksum. • **AMT gateways timed out**—Number of gateways that timed out because of inactivity.

Sample Output

show amt statistics

```
user@host> show amt statistics

AMT receive message count
AMT relay advertisement                    :          2
AMT membership request                     :          5
AMT membership update                      :          5

AMT send message count
AMT relay advertisement                    :          2
AMT membership query                       :          5

AMT error message count
AMT incomplete packet                      :          0
AMT invalid mac                            :          0
AMT unexpected type                        :          0
AMT invalid relay discovery address        :          0
AMT invalid membership request address     :          0
AMT invalid membership update address      :          0
AMT incomplete relay discovery messages    :          0
AMT incomplete membership request messages :          0
AMT incomplete membership update messages  :          0
AMT no active gateway                      :          0
```

```
AMT invalid inner header checksum            :           0
AMT gateways timed out                       :           0
```

show amt summary

Syntax

show amt summary
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced in Junos OS Release 10.2.

Description

Display summary information about the Automatic Multicast Tunneling (AMT) protocol.

Options

none—Display summary information about all AMT protocol instances.

instance *instance-name*—(Optional) Display information for the specified instance only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- clear amt tunnel on page 1392
- show amt statistics on page 1441
- show amt tunnel on page 1446

List of Sample Output

show amt summary on page 1445

Output Fields

Table 37 on page 1444 describes the output fields for the **show amt summary** command. Output fields are listed in the approximate order in which they appear.

Table 37: show amt summary Output Fields

Field Name	Field Description	Level of Output
AMT anycast prefix	Prefix advertised by unicast routing protocols to route AMT discovery messages to the router from nearby AMT gateways.	All levels
AMT anycast address	Anycast address configured from which the anycast prefix is derived.	All levels
AMT local address	Local unique AMT relay IP address configured. Used to send AMT relay advertisement messages, it is the IP source address of AMT control messages and the source address of the data tunnel encapsulation.	All levels
AMT tunnel limit	Maximum number of AMT tunnels that can be created.	All levels
active tunnels	Number of active AMT tunnel interfaces.	All levels

Sample Output

show amt summary

```
user@host> show amt summary
 AMT anycast prefix : 20.0.0.4/32
 AMT anycast address : 20.0.0.4
 AMT local address : 20.0.0.4
 AMT tunnel limit : 1000, active tunnels : 2
```

show amt tunnel

Syntax

show amt tunnel
<brief | detail>
<gateway-address *gateway-ip-address*> <port *port-number*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<tunnel-interface *interface-name*>

Release Information

Command introduced in Junos OS Release 10.2.

Description

Display information about the Automatic Multicast Tunneling (AMT) dynamic tunnels.

Options

none—Display summary information about all AMT protocol instances.

brief | detail—(Optional) Display the specified level of detail.

gateway-address *gateway-ip-address* port *port-number*—(Optional) Display information for the specified AMT gateway only. If no port is specified, display information for all AMT gateways with the given IP address.

instance *instance-name*—(Optional) Display information for the specified instance only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

tunnel-interface *interface-name*—(Optional) Display information for the specified AMT tunnel interface only.

Required Privilege Level

view

Related Documentation

- clear amt tunnel on page 1392
- show amt statistics on page 1441
- show amt summary on page 1444

List of Sample Output

Output Fields

Table 38 on page 1447 describes the output fields for the **show amt tunnel** command. Output fields are listed in the approximate order in which they appear.

Table 38: show amt tunnel Output Fields

Field Name	Field Description	Level of Output
AMT gateway address	Address of the AMT gateway that is being connected by the AMT tunnel.	All levels
port	Client port used by the AMT tunnel.	All levels
AMT tunnel interface	Dynamically created AMT logical interfaces used by the AMT tunnel in the format **ud-FPC/PIC/Port.unit**.	All levels
AMT tunnel state	State of the AMT tunnel. The state is normally **Active**. • **Active**—The tunnel is active. • **Pending**—The tunnel creation is pending. This is a transient state. • **Down**—The tunnel is in the down state. • **Graceful restart pending**—Graceful restart is in progress. • **Reviving**—The routing protocol daemon or Routing Engine was restarted (not gracefully). The tunnel remains in the reviving state until the AMT gateway sends a control message. When the message is received the tunnel is moved to the **Active** state. If no message is received before the AMT tunnel inactivity timer expires, the tunnel is deleted.	All levels
AMT tunnel inactivity timeout	Number of seconds since the most recent control message was received from an AMT gateway. If no message is received before the AMT tunnel inactivity timer expires, the tunnel is deleted.	All levels
Number of groups	Number of multicast groups using the tunnel.	All levels
Group	Multicast group address or addresses using the tunnel.	**detail**
Include Source	Multicast source address for each IGMPv3 group using the tunnel.	**detail**
AMT message count	Statistics for AMT messages: • **AMT Request**—Number of AMT relay tunnel request messages received. • **AMT membership update**—Number of AMT membership update messages received.	All levels

Sample Output

show amt tunnel

```
user@host> show amt tunnel
AMT gateway address : 11.11.11.2, port : 2268
 AMT tunnel interface : ud-5/1/10.1120256
 AMT tunnel state : Active
 AMT tunnel inactivity timeout : 15
 Number of groups : 1

AMT message count:
AMT Request    AMT membership update
2              2
```

show amt tunnel detail

```
user@host> show amt tunnel detail
AMT gateway address : 11.11.11.2, port : 2268
 AMT tunnel interface : ud-5/3/10.1120512
 AMT tunnel state : Active
 AMT tunnel inactivity timeout : 62
 Number of groups : 1
  Group: 226.2.3.2

AMT message count:
AMT Request    AMT membership update
2              2

 AMT gateway address : 11.11.11.3, port : 2268
 AMT tunnel interface : ud-5/2/10.1120513
 AMT tunnel state : Active
 AMT tunnel inactivity timeout : 214
 Number of groups : 1
  Group: 226.2.3.3

AMT message count:
AMT Request    AMT membership update
2              2
```

show amt tunnel tunnel-interface

```
user@host> show amt tunnel tunnel-interface ud-5/3/10.1120512
 AMT gateway address : 11.11.11.2, port : 2268
 AMT tunnel interface : ud-5/3/10.1120512
 AMT tunnel state : Active
 AMT tunnel inactivity timeout : 145
 Number of groups : 1

AMT message count:
AMT Request    AMT membership update
2              2
```

show amt tunnel gateway-address

```
user@host> show amt tunnel gateway-address 11.11.11.3 port 2268
AMT gateway address : 11.11.11.3, port : 2268
 AMT tunnel interface : ud-5/2/10.1120513
 AMT tunnel state : Active
 AMT tunnel inactivity timeout : 214
 Number of groups : 1
  Group: 226.2.3.3

AMT message count:
AMT Request    AMT membership update
2              2
```

show amt tunnel gateway-address detail

```
user@host> show amt tunnel gateway-address 11.11.11.2 detail
AMT gateway address : 11.11.11.2, port : 2268
 AMT tunnel interface : ud-5/3/10.1120512
 AMT tunnel state : Active
 AMT tunnel inactivity timeout : 234
 Number of groups : 1
```

```
  Group: 226.2.3.2

AMT message count:
AMT Request    AMT membership update
2              2
```

show bgp group

List of Syntax

Syntax on page 1450
Syntax (EX Series Switch and QFX Series) on page 1450

Syntax

show bgp group
<brief | detail | summary>
<*group-name*>
<exact-instance *instance-name*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<rtf>

Syntax (EX Series Switch and QFX Series)

show bgp group
<brief | detail | summary>
<*group-name*>
<exact-instance *instance-name*>
<instance *instance-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
exact-instance option introduced in Junos OS Release 11.4.

Description

Display information about the configured BGP groups.

Options

none—Display group information about all BGP groups.

brief | detail | summary—(Optional) Display the specified level of output.

group-name—(Optional) Display group information for the specified group.

exact-instance *instance-name*—(Optional) Display information for the specified instance only.

instance *instance-name*—(Optional) Display information about BGP groups for all routing instances whose name begins with this string (for example, **cust1**, **cust11**, and **cust111** are all displayed when you run the **show bgp group instance cust1** command). The instance name can be master for the main instance, or any valid configured instance name or its prefix.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

rtf—(Optional) Display BGP group route targeting information.

Required Privilege Level

view

List of Sample Output

Output Fields

Table 39 on page 1451 describes the output fields for the **show bgp group** command. Output fields are listed in the approximate order in which they appear.

Table 39: show bgp group Output Fields

Field Name	Field Description	Level of Output
Group Type or **Group**	Type of BGP group: **Internal** or **External**.	All levels
group-index	Index number for the BGP peer group. The index number differentiates between groups when a single BGP group is split because of different configuration options at the group and peer levels.	**rtf detail**
AS	AS number of the peer. For internal BGP (IBGP), this number is the same as **Local AS**.	**brief detail** none
Local AS	AS number of the local routing device.	**brief detail** none
Name	Name of a specific BGP group.	**brief detail** none
Options	The Network Layer Reachability Information (NLRI) format used for BGP VPN multicast.	**none** none
Index	Unique index number of a BGP group.	**brief detail** none
Flags	Flags associated with the BGP group. This field is used by Juniper Networks customer support.	**brief detail** none
BGP-Static Advertisement Policy	Policies configured for the BGP group with the **advertise-bgp-static policy** statement.	**brief** none
Remove-private options	Options associated with the **remove-private** statement.	**brief detail** none
Holdtime	Maximum number of seconds allowed to elapse between successive keepalive or update messages that BGP receives from a peer in the BGP group, after which the connection to the peer is closed and routing devices through that peer become unavailable.	**brief detail** none
Export	Export policies configured for the BGP group with the **export** statement.	**brief detail** none

Table 39: show bgp group Output Fields (continued)

Field Name	Field Description	Level of Output
Optimal Route Reflection	Client nodes (primary and backup) configured in the BGP group.	**brief detail** none
MED tracks IGP metric update delay	Time, in seconds, that updates to multiple exit discriminator (MED) are delayed. Also displays the time remaining before the interval is set to expire	All levels
Traffic Statistics Interval	Time between sample periods for labeled-unicast traffic statistics, in seconds.	**brief detail** none
Total peers	Total number of peers in the group.	**brief detail** none
Established	Number of peers in the group that are in the established state.	All levels
Active/Received/Accepted/Damped	Multipurpose field that displays information about BGP peer sessions. The field's contents depend upon whether a session is established and whether it was established in the main routing device or in a routing instance. • If a peer is not established, the field shows the state of the peer session: **Active**, **Connect**, or **Idle**. • If a BGP session is established in the main routing device, the field shows the number of active, received, accepted, and damped routes that are received from a neighbor and appear in the **inet.0** (main) and **inet.2** (multicast) routing tables. For example, **8/10/10/2 and 2/4/4/0** indicate the following: ◦ 8 active routes, 10 received routes, 10 accepted routes, and 2 damped routes from a BGP peer appear in the **inet.0** routing table. ◦ 2 active routes, 4 received routes, 4 accepted routes, and no damped routes from a BGP peer appear in the **inet.2** routing table.	**summary**
ip-addresses	List of peers who are members of the group. The address is followed by the peer's port number.	All levels
Route Queue Timer	Number of seconds until queued routes are sent. If this time has already elapsed, this field displays the number of seconds by which the updates are delayed.	**detail**
Route Queue	Number of prefixes that are queued up for sending to the peers in the group.	**detail**
inet.*number*	Number of active, received, accepted, and damped routes in the routing table. For example, **inet.0: 7/10/9/0** indicates the following: • 7 active routes, 10 received routes, 9 accepted routes, and no damped routes from a BGP peer appear in the **inet.0** routing table.	none

Table 39: show bgp group Output Fields (continued)

Field Name	Field Description	Level of Output
Table inet.*number*	Information about the routing table. • **Received prefixes**—Total number of prefixes from the peer, both active and inactive, that are in the routing table. • **Active prefixes**—Number of prefixes received from the peer that are active in the routing table. • **Suppressed due to damping**—Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols. • **Advertised prefixes**—Number of prefixes advertised to a peer. • **Received external prefixes**—Total number of prefixes from the external BGP (EBGP) peers, both active and inactive, that are in the routing table. • **Active external prefixes**—Number of prefixes received from the EBGP peers that are active in the routing table. • **Externals suppressed**—Number of routes received from EBGP peers currently inactive because of damping or other reasons. • **Received internal prefixes**—Total number of prefixes from the IBGP peers, both active and inactive, that are in the routing table. • **Active internal prefixes**—Number of prefixes received from the IBGP peers that are active in the routing table. • **Internals suppressed**—Number of routes received from IBGP peers currently inactive because of damping or other reasons. • **RIB State**—Status of the graceful restart process for this routing table: **BGP restart is complete**, **BGP restart in progress**, **VPN restart in progress**, or **VPN restart is complete**.	**detail**
Groups	Total number of groups.	All levels
Peers	Total number of peers.	All levels
External	Total number of external peers.	All levels
Internal	Total number of internal peers.	All levels
Down peers	Total number of unavailable peers.	All levels
Flaps	Total number of flaps that occurred.	All levels
Table	Name of a routing table.	**brief**, none
Tot Paths	Total number of routes.	**brief**, none
Act Paths	Number of active routes.	**brief**, none
Suppressed	Number of routes currently inactive because of damping or other reasons. These routes do not appear in the forwarding table and are not exported by routing protocols.	**brief**, none

Table 39: show bgp group Output Fields (continued)

Field Name	Field Description	Level of Output
History	Number of withdrawn routes stored locally to keep track of damping history.	**brief**, none
Damp State	Number of active routes with a figure of merit greater than zero, but lower than the threshold at which suppression occurs.	**brief**, none
Pending	Routes being processed by the BGP import policy.	**brief**, none
Group	Group the peer belongs to in the BGP configuration.	**detail**
Receive mask	Mask of the received target included in the advertised route.	**detail**
Entries	Number of route entries received.	**detail**
Target	Route target that is to be passed by route-target filtering. If a route advertised from the provider edge (PE) routing device matches an entry in the route-target filter, the route is passed to the peer.	**detail**
Mask	Mask which specifies that the peer receive routes with the given route target.	**detail**

Sample Output

show bgp group

```
user@host> show bgp group
```

show bgp group

```
user@host> show bgp group
Group Type: Internal    AS: 1001                  Local AS: 1001
  Name: ibgp            Index: 2                  Flags: Export Eval
  Holdtime: 0
  Optimal Route Reflection: igp-primary 1.1.1.1, igp-backup 1.1.2.1
  Total peers: 1        Established: 1
  1.1.1.2+179
  Trace options:  all
  Trace file: /var/log/bgp-log size 10485760 files 10
  bgp.l3vpn.2: 0/0/0/0
  vpn-1.inet.2: 0/0/0/0

Group Type: Internal    AS: 1001                  Local AS: 1001
  Name: ibgp            Index: 3                  Flags: Export Eval
  Options: RFC6514CompliantSafi129
  Holdtime: 0
  Optimal Route Reflection: igp-primary 1.1.1.1, igp-backup 1.1.2.1
  Total peers: 1        Established: 1
  1.1.1.5+61698
  Trace options:  all
  Trace file: /var/log/bgp-log size 10485760 files 10
  bgp.l3vpn.2: 2/2/2/0
```

```
  vpn-1.inet.2: 2/2/2/0

Groups: 2  Peers: 2    External: 0    Internal: 2    Down peers: 0   Flaps: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.2
                       2          2          0          0          0          0
vpn-1.inet.0
                       0          0          0          0          0          0
vpn-1.inet.2
                       2          2          0          0          0          0
vpn-1.inet6.0
                       0          0          0          0          0          0
vpn-1.mdt.0
                       0          0          0          0          0          0
```

show bgp group brief

```
user@host> show bgp group brief
 Groups: 2  Peers: 2    External: 0    Internal: 2    Down peers: 1   Flaps: 0
 Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending

 inet.0
                        0          0          0          0          0          0

 bgp.l3vpn.0
                        0          0          0          0          0          0

 bgp.rtarget.0
                        2          0          0          0          0          0
```

show bgp group detail

```
user@host> show bgp group detail
Group Type: Internal    AS: 1                     Local AS: 1
  Name: ibgp            Index: 0                  Flags: <Export Eval>
  Holdtime: 0
  Optimal Route Reflection: igp-primary 1.1.1.1, igp-backup 1.1.2.1
  Total peers: 3        Established: 0
  22.0.0.2
  22.0.0.8
  22.0.0.5

Groups: 1  Peers: 3    External: 0    Internal: 3    Down peers: 3   Flaps: 3
  Table bgp.l3vpn.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
  Table bgp.mdt.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
```

```
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
  Table VPN-A.inet.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
  Table VPN-A.mdt.0
    Received prefixes:            0
    Accepted prefixes:            0
    Active prefixes:              0
    Suppressed due to damping:    0
    Received external prefixes:   0
    Active external prefixes:     0
    Externals suppressed:         0
    Received internal prefixes:   0
    Active internal prefixes:     0
    Internals suppressed:         0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
```

show bgp group rtf detail

```
user@host> show bgp group rtf detail
 Group: internal (group-index: 0)
     Receive mask: 00000002
     Table: bgp.rtarget.0                                    Entries: 2
         Target                          Mask
         100:100/64                      00000002
         200:201/64                      (Group)
 Group: internal (group-index: 1)
     Table: bgp.rtarget.0                                    Entries: 1
         Target                          Mask
         200:201/64                      (Group)
```

show bgp group summary

```
user@host> show bgp group summary
Group        Type       Peers     Established    Active/Received/Accepted/Damped
ibgp         Internal   3         0

Groups: 1  Peers: 3    External: 0    Internal: 3    Down peers: 3   Flaps: 3
  bgp.l3vpn.0      : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
  bgp.mdt.0        : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
```

```
VPN-A.inet.0      : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
VPN-A.mdt.0       : 0/0/0/0 External: 0/0/0/0 Internal: 0/0/0/0
```

show configuration protocols igmp

Syntax show configuration protocols igmp

Release Information Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Display Internet Group Management Protocol (IGMP) information.

Required Privilege Level view

Related Documentation
- IGMP Snooping Overview on page 81
- Configuring IGMP Snooping on Switches on page 88

List of Sample Output show configuration protocols igmp on page 1458

Output Fields Table 40 on page 1458 describes the output fields for the **show configuration protocols igmp** command that relate to IGMP querying.

Table 40: show igmp group Output Fields

Field Name	Field Description	Level of Output
accounting	Enables notification for join and leave events.	All levels
igmp-querier	Configured source address for the IGMP querier.	All levels
interface	Name of the interface that receives IGMP membership reports.	All levels
query-interval	Interval at which the IGMP querier sends general host-query messages to solicit membership information.	All levels
query-response-interval	How long the IGMP querier waits to receive a response from a query message before sending another query.	All levels
src-address	Source address of IGMP queries.	
version	IGMP version.	All levels

Sample Output

show configuration protocols igmp

```
user@switch> show configuration protocols igmp
query-interval 150;
query-response-interval 50;
accounting;
```

```
interface vlan.43 {
 version 2;
 }
igmp-querier {
  src-address 10.0.0.2;
}
```

show dvmrp interfaces

Syntax

show dvmrp interfaces
<logical-system (all | *logical-system-name*)>

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Command introduced before Junos OS Release 7.4.

Description

Display information about Distance Vector Multicast Routing Protocol (DVMRP)–enabled interfaces.

Options

none—(Same as **logical-system all**) Display information about DVMRP-enabled interfaces.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

List of Sample Output

show dvmrp interfaces on page 1462

Output Fields

Table 41 on page 1460 describes the output fields for the **show dvmrp interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 41: show dvmrp interfaces Output Fields

Field Name	Field Description
Interface	Name of the interface.
State	State of the interface: **up** or **down**.
Leaf	Whether the interface is a leaf (that is, whether it has no neighbors) or whether it has neighbors.
Metric	Interface metric: a value from 1 through 31.
Announce	Number of routes the interface is announcing.

Table 41: show dvmrp interfaces Output Fields (continued)

Field Name	Field Description
Mode	DVMRP mode: • **Forwarding**—DVMRP does both the routing and the multicast data forwarding. • **Unicast-routing**—DVMRP does only the routing. Forwarding of the multicast data packets can be done by enabling PIM on the interface.

Sample Output

show dvmrp interfaces

```
user@host> show dvmrp interfaces
Interface State Leaf Metric Announce Mode
fxp0.0   Up   N   1   4 Forwarding
fxp1.0   Up   N   1   4 Forwarding
fxp2.0   Up   N   1   3 Forwarding
lo0.0    Up   Y   1   0 Unicast-routing
```

show dvmrp neighbors

Syntax show dvmrp neighbors
<logical-system (all | *logical-system-name*)>

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Command introduced before Junos OS Release 7.4.

Description Display information about Distance Vector Multicast Routing Protocol (DVMRP) neighbors.

Options **none**—(Same as **logical-system all**) Display information about DVMRP neighbors.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level view

List of Sample Output show dvmrp neighbors on page 1464

Output Fields Table 42 on page 1463 describes the output fields for the **show dvmrp neighbors** command. Output fields are listed in the approximate order in which they appear.

Table 42: show dvmrp neighbors Output Fields

Field Name	Field Description
Neighbor	Address of the neighboring DVMRP router.
Interface	Interface through which the neighbor is reachable.
Version	Version of DVMRP that the neighbor is running, in the format ***majorminor***.
Flags	Information about the neighbor: • **1**—One way. The local router has seen the neighbor, but the neighbor has not seen the local router. • **G**—Neighbor supports generation ID. • **L**—Neighbor is a leaf router. • **M**—Neighbor supports mtrace. • **N**—Neighbor supports netmask in prune messages and graft messages. • **P**—Neighbor supports pruning. • **S**—Neighbor supports SNMP.

Table 42: show dvmrp neighbors Output Fields (continued)

Field Name	Field Description
Routes	Number of routes learned from the neighbor.
Timeout	How long until the DVMRP neighbor information times out, in seconds.
Transitions	Number of generation ID changes that have occurred since the local router learned about the neighbor.

Sample Output

show dvmrp neighbors

```
user@host> show dvmrp neighbors
Neighbor        Interface        Version Flags   Routes Timeout Transitions
192.168.1.1    ipip.0            3.255  PGM         3      28           1
```

show dvmrp prefix

Syntax

show dvmrp prefix
<brief | detail>
<logical-system (all | *logical-system-name*)>
<*prefix*>

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Command introduced before Junos OS Release 7.4.

Description

Display information about Distance Vector Multicast Routing Protocol (DVMRP) prefixes.

Options

none—Display standard information about all DVMRP prefixes.

brief | detail—(Optional) Display the specified level of output.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

prefix—(Optional) Display information about specific prefixes.

Required Privilege Level

view

List of Sample Output

show dvmrp prefix on page 1467
show dvmrp prefix brief on page 1467
show dvmrp prefix detail on page 1467

Output Fields

Table 43 on page 1465 describes the output fields for the **show dvmrp prefix** command. Output fields are listed in the approximate order in which they appear.

Table 43: show dvmrp prefix Output Fields

Field Name	Field Description	Level of Output
Prefix	DVMRP route.	All levels
Next hop	Next hop from which the route was learned.	All levels
Age	Last time that the route was refreshed.	All levels
multicast-group	Multicast group address.	**detail**

Table 43: show dvmrp prefix Output Fields (continued)

Field Name	Field Description	Level of Output
Prunes sent	Number of prune messages sent to the multicast group.	**detail**
Grafts sent	Number of grafts sent to the multicast group.	**detail**
Cache lifetime	Lifetime of the group in the multicast cache, in seconds.	**detail**
Prune lifetime	Lifetime remaining and total lifetime of prune messages, in seconds.	**detail**

Sample Output

show dvmrp prefix

```
user@host> show dvmrp prefix
Prefix              Next hop        Age
10.38.0.0      /30 10.38.0.1       00:06:17
10.38.0.4      /30 10.38.0.5       00:06:13
10.38.0.8      /30 10.38.0.2       00:00:04
10.38.0.12     /30 10.38.0.6       00:00:04
10.255.14.114  /32 10.255.14.114   00:06:17
10.255.14.142  /32 10.38.0.2       00:00:04
10.255.14.144  /32 10.38.0.2       00:00:04
10.255.70.15   /32 10.38.0.6       00:00:04
192.168.14.0   /24 192.168.14.114  00:06:17
192.168.195.40 /30 192.168.195.41  00:06:17
192.168.195.92 /30 10.38.0.2       00:00:04
```

show dvmrp prefix brief

The output for the **show dvmrp prefix brief** command is identical to that for the **show dvmrp prefix** command.

show dvmrp prefix detail

```
user@host> show dvmrp prefix detail
Prefix              Next hop        Age
10.38.0.0      /30 10.38.0.1       00:06:28
10.38.0.4      /30 10.38.0.5       00:06:24
10.38.0.8      /30 10.38.0.2       00:00:15
10.38.0.12     /30 10.38.0.6       00:00:15
10.255.14.114  /32 10.255.14.114   00:06:28
10.255.14.142  /32 10.38.0.2       00:00:15
10.255.14.144  /32 10.38.0.2       00:00:15
10.255.70.15   /32 10.38.0.6       00:00:15
192.168.14.0   /24 192.168.14.114  00:06:28
192.168.195.40 /30 192.168.195.41  00:06:28
192.168.195.92 /30 10.38.0.2       00:00:15
```

show dvmrp prunes

Syntax

show dvmrp prunes
<all | rx | tx>
<logical-system (all | *logical-system-name*)>

Release Information

NOTE: Distance Vector Multicast Routing Protocol (DVMRP) was deprecated in Junos OS Release 16.1. Although DVMRP commands continue to be available and configurable in the CLI, they are no longer visible and are scheduled for removal in a subsequent release.

Command introduced before Junos OS Release 7.4.

Description

Display information about active Distance Vector Multicast Routing Protocol (DVMRP) prune messages.

Options

none—Display received and transmitted DVMRP prune information.

all—(Optional) Display information about all received and transmitted prune messages.

rx—(Optional) Display information about received prune messages.

tx—(Optional) Display information about transmitted prune messages.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

List of Sample Output

show dvmrp prunes on page 1469

Output Fields

Table 44 on page 1468 describes the output fields for the **show dvmrp prunes** command. Output fields are listed in the approximate order in which they appear.

Table 44: show dvmrp prunes Output Fields

Field Name	Field Description
Group	Group address.
Source prefix	Prefix for the prune.
Timeout	How long until the prune message expires, in seconds.
Neighbor	Neighbor to which the prune was sent or from which the prune was received.

Sample Output

show dvmrp prunes

```
user@host> show dvmrp prunes
Group           Source prefix      Timeout Neighbor
224.0.1.1       128.112.0.0    /12    7077 192.168.1.1
224.0.1.32      160.0.0.0      /3     7087 192.168.1.1
224.2.123.4     136.0.0.0      /5     6955 192.168.1.1
224.2.127.1     129.0.0.0      /8     7046 192.168.1.1
224.2.135.86    128.102.128.0  /17    7071 192.168.1.1
224.2.135.86    129.0.0.0      /8     7074 192.168.1.1
224.2.135.86    130.0.0.0      /7     7071 192.168.1.1
...
```

show igmp interface

List of Syntax

Syntax on page 1470
Syntax (EX Series Switches and the QFX Series) on page 1470

Syntax

show igmp interface
<brief | detail>
<*interface-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switches and the QFX Series)

show igmp interface
<brief | detail>
<*interface-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display information about Internet Group Management Protocol (IGMP)-enabled interfaces.

Options

none—Display standard information about all IGMP-enabled interfaces.

brief | detail—(Optional) Display the specified level of output.

interface-name—(Optional) Display information about the specified IGMP-enabled interface only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- clear igmp membership on page 1394

List of Sample Output

show igmp interface on page 1472
show igmp interface brief on page 1473
show igmp interface detail on page 1473
show igmp interface <interface-name> on page 1473

Output Fields

Table 45 on page 1471 describes the output fields for the **show igmp interface** command. Output fields are listed in the approximate order in which they appear.

Table 45: show igmp interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface.	All levels
Querier	Address of the routing device that has been elected to send membership queries.	All levels
State	State of the interface: **Up** or **Down**.	All levels
SSM Map Policy	Name of the source-specific multicast (SSM) map policy that has been applied to the IGMP interface.	All levels
Timeout	How long until the IGMP querier is declared to be unreachable, in seconds.	All levels
Version	IGMP version being used on the interface: **1** , **2** , or **3**.	All levels
Groups	Number of groups on the interface.	All levels
Group limit	Maximum number of groups allowed on the interface. Any joins requested after the limit is reached are rejected.	All levels
Group threshold	Configured threshold at which a warning message is generated. This threshold is based on a percentage of groups received on the interface. If the number of groups received reaches the configured threshold, the device generates a warning message.	All levels
Group log-interval	Time (in seconds) between consecutive log messages.	All levels
Immediate Leave	State of the immediate leave option: • **On**—Indicates that the router removes a host from the multicast group as soon as the router receives a leave group message from a host associated with the interface. • **Off**—Indicates that after receiving a leave group message, instead of removing a host from the multicast group immediately, the router sends a group query to determine if another receiver responds.	All levels
Promiscuous Mode	State of the promiscuous mode option: • **On**—Indicates that the router can accept IGMP reports from subnetworks that are not associated with its interfaces. • **Off**—Indicates that the router can accept IGMP reports only from subnetworks that are associated with its interfaces.	All levels
Distributed	State of IGMP, which, by default, takes place on the Routing Engine for MX Series routers but can be distributed to the Packet Forwarding Engine to provide faster processing of join and leave events. • **On**—distributed IGMP is enabled.	All levels

Table 45: show igmp interface Output Fields (continued)

Field Name	Field Description	Level of Output
Passive	State of the passive mode option: • **On**—Indicates that the router can run IGMP on the interface but not send or receive control traffic such as IGMP reports, queries, and leaves. • **Off**—Indicates that the router can run IGMP on the interface and send or receive control traffic such as IGMP reports, queries, and leaves. The **passive** statement enables you to selectively activate up to two out of a possible three available query or control traffic options. When enabled, the following options appear after the **on** state declaration: • **send-general-query**—The interface sends general queries. • **send-group-query**—The interface sends group-specific and group-source-specific queries. • **allow-receive**—The interface receives control traffic.	All levels
OIF map	Name of the OIF map (if configured) associated with the interface.	All levels
SSM map	Name of the source-specific multicast (SSM) map (if configured) used on the interface.	All levels
Configured Parameters	Information configured by the user: • **IGMP Query Interval**—Interval (in seconds) at which this router sends membership queries when it is the querier. • **IGMP Query Response Interval**—Time (in seconds) that the router waits for a report in response to a general query. • **IGMP Last Member Query Interval**—Time (in seconds) that the router waits for a report in response to a group-specific query. • **IGMP Robustness Count**—Number of times the router retries a query.	All levels
Derived Parameters	Derived information: • **IGMP Membership Timeout**—Timeout period (in seconds) for group membership. If no report is received for these groups before the timeout expires, the group membership is removed. • **IGMP Other Querier Present Timeout**—Time (in seconds) that the router waits for the IGMP querier to send a query.	All levels

Sample Output

show igmp interface

```
user@host> show igmp interface
Interface: at-0/3/1.0
    Querier: 203.0.3.113.31
    State:         Up Timeout:    None Version:  2 Groups:      4
    SSM Map Policy: ssm-policy-A
Interface: so-1/0/0.0
    Querier: 203.0.113.11
    State:         Up Timeout:    None Version:  2 Groups:      2
    SSM Map Policy: ssm-policy-B
Interface: so-1/0/1.0
```

```
    Querier: 203.0.113.21
    State:         Up Timeout:    None Version:  2 Groups:      4
    SSM Map Policy: ssm-policy-C
Immediate Leave: On
Promiscuous Mode: Off
Passive: Off
Distributed: OnConfigured Parameters:

IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

show igmp interface brief

The output for the **show igmp interface brief** command is identical to that for the **show igmp interface** command. For sample output, see show igmp interface on page 1472.

show igmp interface detail

The output for the **show igmp interface detail** command is identical to that for the **show igmp interface** command. For sample output, see show igmp interface on page 1472.

show igmp interface <interface-name>

```
user@host# show igmp interface ge-3/2/0.0
Interface: ge-3/2/0.0
  Querier: 203.0.113.111
  State: Up Timeout:    None
  Version:  3
  Groups:      1
  Group limit: 8
  Group threshold: 60
  Group log-interval: 10
  Immediate leave:  Off
  Promiscuous mode:  Off
  Distributed: On
```

show igmp group

List of Syntax
Syntax on page 1474
Syntax (EX Series Switch and the QFX Series) on page 1474

Syntax
show igmp group
<brief | detail>
<*group-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show igmp group
<brief | detail>
<*group-name*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display Internet Group Management Protocol (IGMP) group membership information.

Options
none—Display standard information about membership for all IGMP groups.

brief | detail—(Optional) Display the specified level of output.

group-name—(Optional) Display group membership for the specified IP address only.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

Related Documentation
- clear igmp membership on page 1394

List of Sample Output
show igmp group (Include Mode) on page 1475
show igmp group (Exclude Mode) on page 1476
show igmp group brief on page 1476
show igmp group detail on page 1476

Output Fields
Table 40 on page 1458 describes the output fields for the **show igmp group** command. Output fields are listed in the approximate order in which they appear.

Table 46: show igmp group Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface that received the IGMP membership report. A name of **local** indicates that the local routing device joined the group itself.	All levels
Group	Group address.	All levels
Group Mode	Mode the SSM group is operating in: **Include** or **Exclude**.	All levels
Source	Source address.	All levels
Source timeout	Time remaining until the group traffic is no longer forwarded. The timer is refreshed when a listener in include mode sends a report. A group in exclude mode or configured as a static group displays a zero timer.	**detail**
Last reported by	Address of the host that last reported membership in this group.	All levels
Timeout	Time remaining until the group membership is removed.	**brief none**
Group timeout	Time remaining until a group in exclude mode moves to include mode. The timer is refreshed when a listener in exclude mode sends a report. A group in include mode or configured as a static group displays a zero timer.	**detail**
Type	Type of group membership: • **Dynamic**—Host reported the membership. • **Static**—Membership is configured.	All levels

Sample Output

show igmp group (Include Mode)

```
user@host> show igmp group
Interface: t1-0/1/0.0
    Group: 198.51.100.1
        Group mode: Include
        Source: 203.0.113.2
        Last reported by: 203.0.113.52
        Timeout:      24 Type: Dynamic
    Group: 198.51.100.1
        Group mode: Include
        Source: 203.0.113.3
        Last reported by: 203.0.113.52
        Timeout:      24 Type: Dynamic
    Group: 198.51.100.1
        Group mode: Include
        Source: 203.0.113.4
        Last reported by: 203.0.113.52
        Timeout:      24 Type: Dynamic
    Group: 198.51.100.2
        Group mode: Include
        Source: 203.0.113.4
        Last reported by: 203.0.113.52
```

```
        Timeout:      24 Type: Dynamic
Interface: t1-0/1/1.0
Interface: ge-0/2/2.0
Interface: ge-0/2/0.0
Interface: local
    Group: 198.51.100.12
        Source: 0.0.0.0
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: 198.51.100.22
        Source: 0.0.0.0
        Last reported by: Local
        Timeout:       0 Type: Dynamic
```

show igmp group (Exclude Mode)

```
user@host> show igmp group
Interface: t1-0/1/0.0
Interface: t1-0/1/1.0
Interface: ge-0/2/2.0
Interface: ge-0/2/0.0
Interface: local
    Group: 198.51.100.2
        Source: 0.0.0.0
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: 198.51.100.22
        Source: 0.0.0.0
        Last reported by: Local
        Timeout:       0 Type: Dynamic
```

show igmp group brief

The output for the **show igmp group brief** command is identical to that for the **show igmp group** command.

show igmp group detail

```
user@host> show igmp group detail
Interface: t1-0/1/0.0
    Group: 198.51.100.1
        Group mode: Include
        Source: 203.0.113.2
        Source timeout: 12
        Last reported by: 203.0.113.52
        Group timeout:       0 Type: Dynamic
    Group: 198.51.100.1
        Group mode: Include
        Source: 203.0.113.3
        Source timeout: 12
        Last reported by: 203.0.113.52
        Group timeout:       0 Type: Dynamic
    Group: 198.51.100.1
        Group mode: Include
        Source: 203.0.113.4
        Source timeout: 12
        Last reported by: 203.0.113.52
        Group timeout:       0 Type: Dynamic
    Group: 198.51.100.2
```

```
        Group mode: Include
        Source: 203.0.113.4
        Source timeout: 12
        Last reported by: 203.0.113.52
        Group timeout:       0 Type: Dynamic
Interface: t1-0/1/1.0
Interface: ge-0/2/2.0
Interface: ge-0/2/0.0
Interface: local
    Group: 198.51.100.12
        Group mode: Exclude
        Source: 0.0.0.0
        Source timeout: 0
        Last reported by: Local
        Group timeout:       0 Type: Dynamic
    Group: 198.51.100.22
        Group mode: Exclude
        Source: 0.0.0.0
        Source timeout: 0
        Last reported by: Local
        Group timeout:       0 Type: Dynamic
```

show igmp snooping interface

Syntax show igmp snooping interface *interface-name*
<brief | detail>
<bridge-domain *bridge-domain-name*>
<logical-system *logical-system-name*>
<virtual-switch *virtual-switch-name*>
<vlan-id *vlan-identifier*>

Release Information Command introduced in Junos OS Release 8.5.

Description Display IGMP snooping interface information.

Options **none** —Display detailed information.

brief | detail—(Optional) When applicable, this option lets you choose the how much detail to display.

bridge-domain *bridge-domain-name*—(Optional) Display information about a particular bridge domain.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

virtual-switch *virtual-switch-name*—(Optional) Display information about a particular virtual switch.

vlan-id *vlan-identifier*—(Optional) Display information about a particular VLAN.

Required Privilege Level view

Related Documentation
- show igmp snooping membership on page 1483
- show igmp snooping statistics on page 1489

List of Sample Output
show igmp snooping interface on page 1479
show igmp snooping interface (logical systems) on page 1480
show igmp snooping interface (Group Limit Configured) on page 1482

Output Fields Table 47 on page 1478 lists the output fields for the **show igmp snooping interface** command. Output fields are listed in the approximate order in which they appear.

Table 47: show igmp snooping interface Output Fields

Field Name	Field Description	Level of Output
Routing-instance	Routing instance for IGMP snooping.	All levels

Table 47: show igmp snooping interface Output Fields (continued)

Field Name	Field Description	Level of Output
Learning Domain	Learning domain for snooping.	All levels
IGMP Query Interval	Frequency (in seconds) with which this router sends membership queries when it is the querier.	All levels
IGMP Query Response Interval	Time (in seconds) that the router waits for a response to a general query.	All levels
IGMP Last Member Query Interval	Time (in seconds) that the router waits for a report in response to a group-specific query.	All levels
IGMP Robustness Count	Number of times the router retries a query.	All levels
immediate-leave	State of immediate leave: **On** or **Off**.	All levels
router-interface	Router interfaces that are part of this learning domain.	All levels
Group limit	Maximum number of (source,group) pairs allowed per interface. When a group limit is not configured, this field is not shown.	All levels
interface	Interfaces that are being snooped in this learning domain.	All levels
Groups	Number of groups on the interface.	All levels
State	State of the interface: **Up** or **Down**.	All levels
Up Groups	Number of active multicast groups attached to the logical interface.	All levels
IGMP Membeship Timeout	Timeout for group membership. If no report is received for these groups before the timeout expires, the group membership is removed.	All levels
IGMP Other Querier Present Timeout	Time that the router waits for the IGMP querier to send a query.	All levels

Sample Output

show igmp snooping interface

```
user@host> show igmp snooping interface ge-0/1/4
Instance: default-switch

Bridge-Domain: sample

Learning-Domain: default
Interface: ge-0/1/4.0
State: Up Groups: 0
Immediate leave: Off
Router interface: no
```

```
Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Derived Parameters:
IGMP Membership Timeout: 260.0
IGMP Other Querier Present Timeout: 255.0
```

show igmp snooping interface (logical systems)

```
user@host> show igmp snooping interface logical-system all
logical-system: default
Instance: VPLS-6
Learning-Domain: default
Interface: ge-0/2/2.601
    State:         Up Groups:     10
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Instance: VS-4
Bridge-Domain: VS-4-BD-1
Learning-Domain: vlan-id 1041
Interface: ae2.3
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no
Interface: ge-0/2/2.1041
    State:         Up Groups:     20
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Instance: default-switch
Bridge-Domain: bd-200
Learning-Domain: default
Interface: ge-0/2/2.100
    State:         Up Groups:     20
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Bridge-Domain: bd0
```

```
Learning-Domain: default
Interface: ae0.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: yes
Interface: ae1.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no
Interface: ge-0/2/2.0
    State:         Up Groups:     32
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Instance: VPLS-1
Learning-Domain: default
Interface: ge-0/2/2.502
    State:         Up Groups:     11
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Instance: VS-1
Bridge-Domain: VS-BD-1
Learning-Domain: default
Interface: ae2.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no
Interface: ge-0/2/2.1010
    State:         Up Groups:     20
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Bridge-Domain: VS-BD-2
Learning-Domain: default
Interface: ae2.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no
Interface: ge-0/2/2.1011
    State:         Up Groups:     20
    Immediate leave: Off
    Router interface: no
```

```
Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2

Instance: VPLS-p2mp
Learning-Domain: default
Interface: ge-0/2/2.3001
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2
```

show igmp snooping interface (Group Limit Configured)

```
user@host> show igmp snooping interface instance vpls1
Instance: vpls1

Learning-Domain: default
Interface: ge-1/3/9.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: yes
Interface: ge-1/3/8.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: yes
    Group limit:   1000

Configured Parameters:
IGMP Query Interval: 125.0
IGMP Query Response Interval: 10.0
IGMP Last Member Query Interval: 1.0
IGMP Robustness Count: 2
```

show igmp snooping membership

Syntax

show igmp snooping membership
<brief | detail>
<interface *interface-name*>
<vlan (*vlan-id* | *vlan-name*)>
<bridge-domain *bridge-domain-name*>
<group *group-name*>
<logical-system *logical-system-name*>
<virtual-switch *virtual-switch-name*>
<vlan-id *vlan-identifier*>

Release Information

Command introduced in Junos OS Release 8.5.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description

Display the multicast group membership information maintained by IGMP snooping.

Options

none—Display the multicast group membership information about all VLANs on which IGMP snooping is enabled.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

interface ***interface-name***—(Optional) Display the multicast group membership information about the specified interface.

vlan (*vlan-id* | *vlan-name*)—(Optional) Display the multicast group membership for the specified VLAN.

bridge-domain ***bridge-domain-name***—(Optional) Display information about a particular bridge domain.

group ***group-name*** —(Optional) Display information about this group address.

logical-system ***logical-system-name***—(Optional) Display information about a particular logical system, or type 'all'.

virtual-switch ***virtual-switch-name***—(Optional) Display information about a particular virtual switch.

vlan-id ***vlan-identifier***—(Optional) Display information about a particular VLAN.

Required Privilege Level

view

Related Documentation

- show igmp snooping interface on page 1478
- show igmp snooping statistics on page 1489
- clear igmp snooping membership on page 1397

List of Sample Output

Output Fields

Table 48 on page 1484 lists the output fields for the **show igmp snooping membership** command. Output fields are listed in the approximate order in which they appear.

Table 48: show igmp snooping membership Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All
Instance	Routing instance for IGMP snooping.	All levels
Learning Domain	Learning domain for snooping.	All levels
Interface	Interface on which this router is a proxy.	**detail**
Up Groups	Number of active multicast groups attached to the logical interface.	All levels
Group	Multicast group address in the membership database.	All levels
Group Mode	Mode the SSM group is operating in: **Include** or **Exclude**.	All levels
Source	Source address used on queries.	**detail**
Last reported by	Address of source last replying to the query.	**detail**
Group Timeout	Time remaining until a group in exclude mode moves to include mode. The timer is refreshed when a listener in exclude mode sends a report. A group in include mode or configured as a static group displays a zero timer.	All levels
Timeout	Length of time (in seconds) left until the entry is purged.	**detail**
Type	Way that the group membership information was learned: • **Dynamic**—Group membership was learned by the IGMP protocol. • **Static**—Group membership was learned by configuration.	**detail**
Include receiver	Source address of receiver included in membership with timeout (in seconds).	**detail**

Sample Output

show igmp snooping membership

```
user@host> show igmp snooping membership
Instance: vpls2

Learning-Domain: vlan-id 2
Interface: ge-3/0/0.2
Up Groups:      0
Interface: ge-3/1/0.2
Up Groups:      0
Interface: ge-3/1/5.2
Up Groups:      0

Instance: vpls1

Learning-Domain: vlan-id 1
Interface: ge-3/0/0.1
Up Groups:      0
Interface: ge-3/1/0.1
Up Groups:      0
Interface: ge-3/1/5.1
Up Groups:      1
    Group: 233.252.0.99
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 233.252.0.87
        Group timeout:     173 Type: Dynamic
```

show igmp-snooping membership (SRX1500)

```
user@host> show igmp-snooping membership
Instance: default-switch

Vlan: v1

Learning-Domain: default
Interface: ge-0/0/3.0, Groups: 1
Group: 233.252.0.100
Group mode: Exclude
Source: 0.0.0.0
Last reported by: Local
Group timeout: 0 Type: Static
```

show igmp-snooping membership detail (SRX1500)

```
user@host> show igmp-snooping membership detail

VLAN: vlan2 Tag: 2 (Index: 3)
    Router interfaces:
        ge-1/0/0.0 dynamic Uptime: 00:14:24 timeout: 253
    Group: 233.252.0.99
      ge-1/0/17.0 259 Last reporter: 10.0.0.90 Receiver count: 1
      Uptime: 00:00:19 timeout: 259 Flags: <V3-hosts>
        Include source: 10.2.11.5, 10.2.11.12
```

show igmp snooping membership (Exclude Mode)

```
user@host> show igmp snooping membership
Instance: vpls2

Learning-Domain: vlan-id 2
Interface: ge-3/0/0.2
Up Groups:      0
Interface: ge-3/1/0.2
Up Groups:      0
Interface: ge-3/1/5.2
Up Groups:      0

Instance: vpls1

Learning-Domain: vlan-id 1
Interface: ge-3/0/0.1
Up Groups:      0
Interface: ge-3/1/0.1
Up Groups:      0
Interface: ge-3/1/5.1
Up Groups:      1
    Group: 233.252.0.99
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 233.252.0.87
        Group timeout:     173 Type: Dynamic
```

show igmp-snooping membership detail (SRX1500)

```
user@host> show igmp-snooping membership detail

VLAN: vlan2 Tag: 2 (Index: 3)
    Router interfaces:
        ge-1/0/0.0 dynamic Uptime: 00:14:24 timeout: 253
    Group: 233.252.0.99
      ge-1/0/17.0 259 Last reporter: 233.252.0.82 Receiver count: 1
      Uptime: 00:00:19 timeout: 259 Flags: <V3-hosts>
        Include source: 233.252.0.84, 233.252.0.83
```

show igmp-snooping membership vlan detail (SRX1500)

```
user@host> show igmp-snooping membership vlan vlan700 detail
VLAN: vlan700 Tag: 700 (Index: 52)
    Router interfaces:
        ae2.0 dynamic Uptime: 16:53:13 timeout: 245
   Group: 233.252.0.1
50     ge-0/0/1.0 Last reporter: 233.252.0.87
     Uptime: 17:00:52 timeout: 237 Flags: <V2-hosts>
     ge-0/0/0.0 Last reporter: 10.2.188.202
     Uptime: 17:00:50 timeout: 243 Flags: <V2-hosts>
```

show igmp snooping membership interface ge-0/1/2.200

```
user@host> show igmp snooping membership interface ge-0/1/2.200
Instance: bridge-domain bar

Learning-Domain: default
Interface: ge-0/1/2.200
```

```
    Group: 233.252.0.1
        Source: 0.0.0.0
        Timeout:     391 Type: Static
    Group: 232.1.1.1
        Source: 192.128.1.1
        Timeout:       0 Type: Static
```

show igmp snooping membership vlan-id 1

```
user@host> show igmp snooping membership vlan-id 1
Instance: vpls2

Instance: vpls1

Learning-Domain: vlan-id 1
Interface: ge-3/0/0.1
Up Groups:      0
Interface: ge-3/1/0.1
Up Groups:      0
Interface: ge-3/1/5.1
Up Groups:      1
    Group: 233.252.0.1
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 233.252.0.82
        Group timeout:     209 Type: Dynamic
```

show igmp snooping options

Syntax

show igmp snooping options
<brief | detail>
instance <*instance-name*>
<logical-system *logical-system-name*>

Release Information

Command introduced in Junos OS Release 13.3 for MX Series routers.

Description

Show the operational status of point-to-multipoint LSP for IGMP snooping routes.

Options

brief | detail—Display the specified level of output per routing instance. The default is brief.

instance-name—(Optional) Output for the specified routing instance only.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

Required Privilege Level

view

Related Documentation

- Configuring Point-to-Multipoint LSP with IGMP Snooping on page 120
- use-p2mp-lsp on page 1366
- multicast-snooping-options on page 1174

List of Sample Output

show igmp snooping options on page 1488

Sample Output

show igmp snooping options

```
user@host> show igmp snooping options

Instance: master
    P2MP LSP in use: no
Instance: default-switch
    P2MP LSP in use: no
Instance: name
    P2MP LSP in use: yes
```

show igmp snooping statistics

Syntax

show igmp snooping statistics
<brief | detail>
<bridge-domain *bridge-domain-name*>
<logical-system *logical-system-name*>
<virtual-switch *virtual-switch-name*>
<vlan-id *vlan-identifier*>

Release Information

Command introduced in Junos OS Release 8.5.
Command introduced in Junos OS Release 18.1R1 for the SR1500 devices.

Description

Display IGMP snooping statistics.

Options

none—(Optional) Display detailed information.

brief | detail—(Optional) Display the specified level of output.

bridge-domain *bridge-domain-name*—(Optional) Display information about a particular bridge domain.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

virtual-switch *virtual-switch-name*—(Optional) Display information about a particular virtual switch.

vlan-id *vlan-identifier*—(Optional) Display information about a particular VLAN.

Required Privilege Level

view

Related Documentation

- show igmp snooping interface on page 1478
- show igmp snooping membership on page 1483
- clear igmp snooping statistics on page 1399

List of Sample Output

show igmp snooping statistics on page 1491
show igmp-snooping statistics (SRX1500) on page 1491
show igmp snooping statistics logical-systems all on page 1492
show igmp snooping statistics interface (Bridge Domains Configured) on page 1493

Output Fields

Table 49 on page 1490 lists the output fields for the **show igmp snooping statistics** command. Output fields are listed in the approximate order in which they appear.

Table 49: show igmp snooping statistics Output Fields

Field Name	Field Description	Level of Output
Routing-instance	Routing instance for IGMP snooping.	All levels
IGMP packet statistics	Heading for IGMP snooping statistics for all interfaces or for the specified interface.	All levels
learning-domain	Appears at end of "IGMP packets statistics" line.	All levels
IGMP Message type	Summary of IGMP statistics: • **Membership Query**—Number of membership queries sent and received. • **V1 Membership Report**—Number of version 1 membership reports sent and received. • **DVMRP**—Number of DVMRP messages sent or received. • **PIM V1**—Number of PIM version 1 messages sent or received. • **Cisco Trace**—Number of Cisco trace messages sent or received. • **V2 Membership Report**—Number of version 2 membership reports sent or received. • **Group Leave**—Number of group leave messages sent or received. • **Domain Wide Report**—Number of domain-wide reports sent or received. • **V3 Membership Report**—Number of version 3 membership reports sent or received. • **Other Unknown types**—Number of unknown message types received. • **IGMP v3 unsupported type**—Number of messages received with unknown and unsupported IGMP version 3 message types. • **IGMP v3 source required for SSM**—Number of IGMP version 3 messages received that contained no source. • **IGMP v3 mode not applicable for SSM**—Number of IGMP version 3 messages received that did not contain a mode applicable for source-specific multicast (SSM).	All levels
Received	Number of messages received.	All levels
Sent	Number of messages sent.	All levels
Rx errors	Number of received packets that contained errors.	All levels
IGMP Global Statistics	Summary of IGMP snooping statistics for all interfaces. • **Bad Length**—Number of messages received with length errors so severe that further classification could not occur. • **Bad Checksum**—Number of messages received with a bad IP checksum. No further classification was performed. • **Rx non-local**—Number of messages received from senders that are not local.	All levels

Sample Output

show igmp snooping statistics

```
user@host> show igmp snooping statistics
Routing-instance foo

 IGMP packet statistics for all interfaces in learning-domain vlan-100

 IGMP Message type      Received       Sent  Rx errors
 Membership Query             89         51          0
 V1 Membership Report          0          0          0
 DVMRP                         0          0          0
 PIM V1                        0          0          0
 Cisco Trace                   0          0          0
 V2 Membership Report        139          0          0
 Group Leave                   0          0          0
 Domain Wide Report            0          0          0
 V3 Membership Report        136          0          0
 Other Unknown types                                  0
 IGMP v3 unsupported type                             0
 IGMP v3 source required for SSM                     23
 IGMP v3 mode not applicable for SSM                  0

 IGMP Global Statistics
 Bad Length                    0
 Bad Checksum                  0
 Rx non-local                  0

Routing-instance bar

 IGMP packet statistics for all interfaces in learning-domain vlan-100

 IGMP Message type      Received       Sent  Rx errors
 Membership Query             89         51          0
 V1 Membership Report          0          0          0
 DVMRP                         0          0          0
 PIM V1                        0          0          0
 Cisco Trace                   0          0          0
 V2 Membership Report        139          0          0
 Group Leave                   0          0          0
 Domain Wide Report            0          0          0
 V3 Membership Report        136          0          0
 Other Unknown types                                  0
 IGMP v3 unsupported type                             0
 IGMP v3 source required for SSM                     23
 IGMP v3 mode not applicable for SSM                  0

 IGMP Global Statistics
 Bad Length                    0
 Bad Checksum                  0
 Rx non-local                  0
```

show igmp-snooping statistics (SRX1500)

```
user@host> show igmp-snooping statistics
Vlan: v1
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          0          0
V1 Membership Report          0          0          0
```

```
DVMRP                          0          0          0
PIM V1                         0          0          0
Cisco Trace                    0          0          0
V2 Membership Report           0          0          0
Group Leave                    0          0          0
Mtrace Response                0          0          0
Mtrace Request                 0          0          0
Domain Wide Report             0          0          0
V3 Membership Report           0          0          0
Other Unknown types                                  0
```

show igmp snooping statistics logical-systems all

```
user@host> show igmp snooping statistics logical-systems all

logical-system: default
Bridge: VPLS-6
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          4          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0

Learning-Domain: vlan-id 1041 bridge-domain VS-4-BD-1
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          4          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0

Bridge: VPLS-p2mp
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          2          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0
```

```
Bridge: VS-BD-1
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          6          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0
```

show igmp snooping statistics interface (Bridge Domains Configured)

```
user@host> show igmp snooping statistics interface

Bridge: bridge-domain1
IGMP interface packet statistics for ge-2/0/8.0
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          2          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0

Bridge: bridge-domain2
IGMP interface packet statistics for ge-2/0/8.0
IGMP Message type      Received       Sent  Rx errors
Membership Query              0          2          0
V1 Membership Report          0          0          0
DVMRP                         0          0          0
PIM V1                        0          0          0
Cisco Trace                   0          0          0
V2 Membership Report          0          0          0
Group Leave                   0          0          0
Mtrace Response               0          0          0
Mtrace Request                0          0          0
Domain Wide Report            0          0          0
V3 Membership Report          0          0          0
Other Unknown types                                  0
```

show igmp-snooping vlans

Syntax
show igmp-snooping vlans
<**brief** | **detail**>
<**vlan** ***vlan-id*** | *vlan-name*>

Release Information
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description
Display IGMP snooping VLAN information.

Options
none—Display general parameters.

brief | detail—(Optional) Display the specified level of output.

vlan *vlan-id* | vlan *vlan-number*—(Optional) Display VLAN information for the specified VLAN.

Required Privilege Level
view

Related Documentation
- Monitoring IGMP Snooping on page 96
- Configuring IGMP Snooping on Switches on page 88
- *show igmp-snooping route*
- *show igmp-snooping statistics*

List of Sample Output
show igmp-snooping vlans on page 1495
show igmp-snooping vlans vlan on page 1495
show igmp-snooping vlans vlan detail on page 1495

Output Fields
Table 50 on page 1494 lists the output fields for the **show igmp-snooping vlans** command. Output fields are listed in the approximate order in which they appear.

Table 50: show igmp-snooping vlans Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All levels
IGMP-L2-Querier	Source address for IGMP snooping queries (if switch is an IGMP querier)	All levels
Interfaces	Number of interfaces in the VLAN.	All levels
Groups	Number of groups in the VLAN.	All levels
MRouters	Number of multicast routers associated with the VLAN.	All levels

Table 50: show igmp-snooping vlans Output Fields (continued)

Field Name	Field Description	Level of Output
Receivers	Number of host receivers in the VLAN.	All levels
Tag	Numerical identifier of the VLAN.	**detail**
tagged | untagged	Interface participates in a tagged (802.1Q) or untagged (native) VLAN.	**detail**
vlan-interface	Internal VLAN interface identifier.	**detail**
Membership timeout	Membership timeout value.	**detail**
Querier timeout	Timeout value for interfaces dynamically marked as router or switch interfaces (interfaces that receive queries). When the querier timeout is reached, the switch marks the interface as a host interface.	**detail**
Interface	Name of the interface.	**detail**
Reporters	Number of dynamic groups on an interface.	**detail**

Sample Output

show igmp-snooping vlans

```
user@switch> show igmp-snooping vlans
VLAN     Interfaces Groups MRouters Receivers
default     0          0      0         0
v1          11         50     0         0
v10         1          0      0         0
v11         1          0      0         0
v180        3          0      1         0
v181        3          0      0         0
v182        3          0      0         0
```

show igmp-snooping vlans vlan

```
user@switch> show igmp-snooping vlans vlan v10
user@switch> show igmp-snooping vlans vlan v10
VLAN       Interfaces Groups MRouters Receivers
v10        1          0      0        0
```

show igmp-snooping vlans vlan detail

```
user@switch> show igmp-snooping vlans vlan v10 detail
VLAN: v10, Tag: 10, vlan-interface: vlan.10
        Interface: ge-0/0/10.0, tagged, Groups: 0
IGMP-L2-Querier: Stopped, SourceAddress: 10.10.1.2
```

show ingress-replication mvpn

Syntax show ingress-replication mvpn

Release Information Command introduced in Junos OS Release 10.4.

Description Display the state and configuration of the ingress replication tunnels created for the MVPN application when using the **mpls-internet-multicast** routing instance type.

Required Privilege Level View

List of Sample Output show ingress-replication mvpn on page 1496

Output Fields Table 51 on page 1496 lists the output fields for the **show ingress-replication mvpn** command. Output fields are listed in the approximate order in which they appear.

Table 51: show ingress-replication mvpn Output Fields

Field Name	Field Description
Ingress tunnel	Identifies the MVPN ingress replication tunnel.
Application	Identifies the application (MVPN).
Unicast tunnels	List of unicast tunnels in use.
Leaf address	Address of the tunnel.
Tunnel type	Identifies the unicast tunnel type.
Mode	Indicates whether the tunnel was created as a new tunnel for the ingress replication, or if an existing tunnel was used.
State	Indicates whether the tunnel is Up or Down.

Sample Output

show ingress-replication mvpn

```
user@host> show ingress-replication mvpn
Ingress Tunnel: mvpn:1
  Application: MVPN
  Unicast tunnels
    Leaf Address       Tunnel-type       Mode       State
    10.255.245.2       P2P LSP           New        Up
    10.255.245.4       P2P LSP           New        Up
Ingress Tunnel: mvpn:2
  Application: MVPN
  Unicast tunnels
```

```
Leaf Address       Tunnel-type       Mode       State
10.255.245.2       P2P LSP           Existing   Up
```

show interfaces (Multicast Tunnel)

Syntax

show interfaces *interface-type*
<brief | detail | extensive | terse>
<descriptions>
<media>
<snmp-index *snmp-index*>
<statistics>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display status information about the specified multicast tunnel interface and its logical encapsulation and de-encapsulation interfaces.

Options

interface-type—On M Series and T Series routers, the interface type is **mt-*fpc/pic/port***.

brief | detail | extensive | terse—(Optional) Display the specified level of output.

descriptions—(Optional) Display interface description strings.

media—(Optional) Display media-specific information about network interfaces.

snmp-index *snmp-index*—(Optional) Display information for the specified SNMP index of the interface.

statistics—(Optional) Display static interface statistics.

Additional Information

The multicast tunnel interface has two logical interfaces: encapsulation and de-encapsulation. These interfaces are automatically created by the Junos OS for every multicast-enabled VPN routing and forwarding (VRF) instance. The encapsulation interface carries multicast traffic traveling from the edge interface to the core interface. The de-encapsulation interface carries traffic coming from the core interface to the edge interface.

Required Privilege Level

view

List of Sample Output

show interfaces (Multicast Tunnel) on page 1500
show interfaces brief (Multicast Tunnel) on page 1500
show interfaces detail (Multicast Tunnel) on page 1500
show interfaces extensive (Multicast Tunnel) on page 1500
show interfaces (Multicast Tunnel Encapsulation) on page 1502
show interfaces (Multicast Tunnel De-Encapsulation) on page 1502

Output Fields

Table 52 on page 1499 lists the output fields for the **show interfaces** (Multicast Tunnel) command. Output fields are listed in the approximate order in which they appear.

Table 52: Multicast Tunnel show interfaces Output Fields

Field Name	Field Description	Level of Output
Physical Interface		
Physical interface	Name of the physical interface.	All levels
Enabled	State of the interface. Possible values are described in the "Enabled Field" section under *Common Output Fields Description*.	All levels
Interface index	Physical interface's index number, which reflects its initialization sequence.	**detail extensive** none
SNMP ifIndex	SNMP index number for the physical interface.	**detail extensive** none
Generation	Unique number for use by Juniper Networks technical support only.	**detail extensive**
Type	Type of interface.	All levels
Link-level type	Encapsulation used on the physical interface.	All levels
MTU	MTU size on the physical interface.	All levels
Speed	Speed at which the interface is running.	All levels
Hold-times	Current interface hold-time up and hold-time down, in milliseconds.	**detail extensive**
Device flags	Information about the physical device. Possible values are described in the "Device Flags" section under *Common Output Fields Description*.	All levels
Interface flags	Information about the interface. Possible values are described in the "Interface Flags" section under *Common Output Fields Description*.	All levels
Input Rate	Input rate in bits per second (bps) and packets per second (pps).	None specified
Output Rate	Output rate in bps and pps.	None specified
Statistics last cleared	Time when the statistics for the interface were last set to zero.	**detail extensive**

Table 52: Multicast Tunnel show interfaces Output Fields (continued)

Field Name	Field Description	Level of Output
Traffic statistics	Number and rate of bytes and packets received and transmitted on the physical interface. • **Input bytes**—Number of bytes received on the interface. • **Output bytes**—Number of bytes transmitted on the interface. • **Input packets**—Number of packets received on the interface. • **Output packets**—Number of packets transmitted on the interface.	All levels

Sample Output

show interfaces (Multicast Tunnel)

```
user@host> show interfaces mt-1/2/0
Physical interface: mt-1/2/0, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 41
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Input rate     : 0 bps (0 pps)
  Output rate    : 0 bps (0 pps)
```

show interfaces brief (Multicast Tunnel)

```
user@host> show interfaces mt-1/2/0 brief
Physical interface: mt-1/2/0, Enabled, Physical link is Up
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Device flags   : Present Running
  Interface flags: SNMP-Traps
```

show interfaces detail (Multicast Tunnel)

```
user@host> show interfaces mt-1/2/0 detail
Physical interface: mt-1/2/0, Enabled, Physical link is Up
  Interface index: 145, SNMP ifIndex: 41, Generation: 28
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
   Traffic statistics:
    Input  bytes  :           170664562              560000 bps
    Output bytes  :           112345376              368176 bps
    Input  packets:             2439107                1000 pps
    Output packets:             2439120                1000 pps
```

show interfaces extensive (Multicast Tunnel)

```
user@host> show interfaces mt-1/2/0 extensive
Physical interface: mt-1/2/0, Enabled, Physical link is Up
   Interface index: 141, SNMP ifIndex: 529, Generation: 144
   Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
   Hold-times     : Up 0 ms, Down 0 ms
   Device flags   : Present Running
```

```
   Interface flags: SNMP-Traps
   Statistics last cleared: Never
   Traffic statistics:
    Input  bytes  :            170664562               560000 bps
    Output bytes  :            112345376               368176 bps
    Input  packets:              2439107                 1000 pps
    Output packets:              2439120                 1000 pps
    IPv6 transit statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0

   Logical interface mt-1/2/0.32768 (Index 83) (SNMP ifIndex 556) (Generation
148)
     Flags: Point-To-Point SNMP-Traps 0x4000 IP-Header
192.0.2.1:10.0.0.6:47:df:64:0000000800000000 Encapsulation: GRE-NULL
     Traffic statistics:
      Input  bytes  :            170418430
      Output bytes  :            112070294
      Input  packets:              2434549
      Output packets:              2435593
      IPv6 transit statistics:
       Input  bytes  :                   0
       Output bytes  :                   0
       Input  packets:                   0
       Output packets:                   0
     Local statistics:
      Input  bytes  :                   0
      Output bytes  :                80442
      Input  packets:                   0
      Output packets:                1031
     Transit statistics:
      Input  bytes  :            170418430               560000 bps
      Output bytes  :            111989852               368176 bps
      Input  packets:              2434549                 1000 pps
      Output packets:              2434562                 1000 pps
      IPv6 transit statistics:
       Input  bytes  :                   0
       Output bytes  :                   0
       Input  packets:                   0
       Output packets:                   0
     Protocol inet, MTU: 1572, Generation: 182, Route table: 4
       Flags: None
     Protocol inet6, MTU: 1572, Generation: 183, Route table: 4
       Flags: None

   Logical interface mt-1/2/0.1081344 (Index 84) (SNMP ifIndex 560) (Generation
149)
     Flags: Point-To-Point SNMP-Traps 0x6000 Encapsulation: GRE-NULL
     Traffic statistics:
      Input  bytes  :               246132
      Output bytes  :               355524
      Input  packets:                 4558
      Output packets:                 4558
      IPv6 transit statistics:
       Input  bytes  :                   0
       Output bytes  :                   0
       Input  packets:                   0
       Output packets:                   0
     Local statistics:
```

```
    Input  bytes  :               246132
    Output bytes  :                    0
    Input  packets:                 4558
    Output packets:                    0
   Transit statistics:
    Input  bytes  :                    0                    0 bps
    Output bytes  :               355524                    0 bps
    Input  packets:                    0                    0 pps
    Output packets:                 4558                    0 pps
    IPv6 transit statistics:
     Input  bytes  :                   0
     Output bytes  :                   0
     Input  packets:                   0
     Output packets:                   0
   Protocol inet, MTU: Unlimited, Generation: 184, Route table: 4
     Flags: None
   Protocol inet6, MTU: Unlimited, Generation: 185, Route table: 4
     Flags: None
```

show interfaces (Multicast Tunnel Encapsulation)

```
user@host> show interfaces mt-3/1/0.32768
Logical interface mt-3/1/0.32768 (Index 67) (SNMP ifIndex 0)
    Flags: Point-To-Point SNMP-Traps 0x4000
    IP-Header 198.51.100.1:10.255.70.15:47:df:64:0000000800000000
    Encapsulation: GRE-NULL
  Input packets : 0
  Output packets: 2
    Protocol inet, MTU: Unlimited
      Flags: None
```

show interfaces (Multicast Tunnel De-Encapsulation)

```
user@host> show interfaces mt-3/1/0.49152
Logical interface mt-3/1/0.49152 (Index 74) (SNMP ifIndex 0)
    Flags: Point-To-Point SNMP-Traps 0x6000 Encapsulation: GRE-NULL
  Input packets : 0
  Output packets: 2
    Protocol inet, MTU: Unlimited
      Flags: None
```

show mld group

Syntax

show mld group
<brief | detail>
<*group-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display information about Multicast Listener Discovery (MLD) group membership.

Options

none—Display standard information about all MLD groups.

brief | detail—(Optional) Display the specified level of output.

group-name—(Optional) Display MLD information about the specified group.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- clear mld membership on page 1402

List of Sample Output

show mld group (Include Mode) on page 1504
show mld group (Exclude Mode) on page 1505
show mld group brief on page 1505
show mld group detail (Include Mode) on page 1505
show mld group detail (Exclude Mode) on page 1506

Output Fields

Table 53 on page 1503 describes the output fields for the **show mld group** command. Output fields are listed in the approximate order in which they appear.

Table 53: show mld group Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface that received the MLD membership report; **local** means that the local router joined the group itself.	All levels
Group	Group address.	All levels
Source	Source address.	All levels
Group Mode	Mode the SSM group is operating in: **Include** or **Exclude**.	All levels
Last reported by	Address of the host that last reported membership in this group.	All levels

Table 53: show mld group Output Fields (continued)

Field Name	Field Description	Level of Output
Source timeout	Time remaining until the group traffic is no longer forwarded. The timer is refreshed when a listener in include mode sends a report. A group in exclude mode or configured as a static group displays a zero timer.	**detail**
Timeout	Time remaining until the group membership is removed.	**brief none**
Group timeout	Time remaining until a group in exclude mode moves to include mode. The timer is refreshed when a listener in exclude mode sends a report. A group in include mode or configured as a static group displays a zero timer.	**detail**
Type	Type of group membership: • **Dynamic**—Host reported the membership. • **Static**—Membership is configured.	All levels

Sample Output

show mld group (Include Mode)

```
user@host> show mld group
Interface: fe-0/1/2.0
    Group: ff02::1:ff05:1a67
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     245 Type: Dynamic
    Group: ff02::1:ffa8:c35e
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     241 Type: Dynamic
    Group: ff02::2:43e:d7f6
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     244 Type: Dynamic
    Group: ff05::2
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     244 Type: Dynamic
Interface: local
    Group: ff02::2
        Source: ::
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: ff02::16
        Source: ::
        Last reported by: Local
        Timeout:       0 Type: Dynamic
```

show mld group (Exclude Mode)

```
user@host> show mld group
Interface: ge-0/2/2.0
Interface: ge-0/2/0.0
    Group: ff02::6
        Source: ::
        Last reported by: fe80::21f:12ff:feb6:4b3a
        Timeout:     245 Type: Dynamic
    Group: ff02::16
        Source: ::
        Last reported by: fe80::21f:12ff:feb6:4b3a
        Timeout:      28 Type: Dynamic
Interface: local
    Group: ff02::2
        Source: ::
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: ff02::16
        Source: ::
        Last reported by: Local
        Timeout:       0 Type: Dynamic
```

show mld group brief

The output for the **show mld group brief** command is identical to that for the **show mld group** command. For sample output, see show mld group (Include Mode) on page 1504
show mld group (Exclude Mode) on page 1505.

show mld group detail (Include Mode)

```
user@host> show mld group detail
Interface: fe-0/1/2.0
    Group: ff02::1:ff05:1a67
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     224 Type: Dynamic
    Group: ff02::1:ffa8:c35e
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     220 Type: Dynamic
    Group: ff02::2:43e:d7f6
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     223 Type: Dynamic
    Group: ff05::2
        Group mode: Include
        Source: ::
        Last reported by: fe80::2e0:81ff:fe05:1a67
        Timeout:     223 Type: Dynamic
Interface: so-1/0/1.0
    Group: ff02::2
        Group mode: Include
        Source: ::
        Last reported by: fe80::280:42ff:fe15:f445
        Timeout:     258 Type: Dynamic
```

```
Interface: local
    Group: ff02::2
        Group mode: Include
        Source: ::
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: ff02::16
        Source: ::
        Last reported by: Local
        Timeout:       0 Type: Dynamic
```

show mld group detail (Exclude Mode)

```
user@host> show mld group detail
Interface: ge-0/2/2.0
Interface: ge-0/2/0.0
    Group: ff02::6
        Group mode: Exclude
        Source: ::
        Source timeout: 0
        Last reported by: fe80::21f:12ff:feb6:4b3a
        Group timeout:     226 Type: Dynamic
    Group: ff02::16
        Group mode: Exclude
        Source: ::
        Source timeout: 0
        Last reported by: fe80::21f:12ff:feb6:4b3a
        Group timeout:     246 Type: Dynamic
Interface: local
    Group: ff02::2
        Group mode: Exclude
        Source: ::
        Source timeout: 0
        Last reported by: Local
        Group timeout:       0 Type: Dynamic
    Group: ff02::16
        Group mode: Exclude
        Source: ::
        Source timeout: 0
        Last reported by: Local
        Group timeout:       0 Type: Dynamic
```

show mld interface

Syntax

show mld interface
<brief | detail>
<*interface-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display information about multipoint Listener Discovery (MLD)-enabled interfaces.

Options

none—Display standard information about all MLD-enabled interfaces.

brief | detail—(Optional) Display the specified level of output.

interface-name—(Optional) Display information about the specified interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- clear mld membership on page 1402

List of Sample Output

show mld interface on page 1509
show mld interface brief on page 1510
show mld interface detail on page 1510
show mld interface <interface-name> on page 1510

Output Fields

Table 54 on page 1507 describes the output fields for the **show mld interface** command. Output fields are listed in the approximate order in which they appear.

Table 54: show mld interface Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the interface.	All levels
Querier	Address of the router that has been elected to send membership queries.	All levels
State	State of the interface: **Up** or **Down**.	All levels
SSM Map Policy	Name of the source-specific multicast (SSM) map policy that has been applied to the interface.	All levels
SSM Map Policy	Name of the source-specific multicast (SSM) map policy at the MLD interface.	All levels

Table 54: show mld interface Output Fields (continued)

Field Name	Field Description	Level of Output
Timeout	How long until the MLD querier is declared to be unreachable, in seconds.	All levels
Version	MLD version being used on the interface: **1** or **2**.	All levels
Groups	Number of groups on the interface.	All levels
Passive	State of the passive mode option: • **On**—Indicates that the router can run IGMP or MLD on the interface but not send or receive control traffic such as IGMP or MLD reports, queries, and leaves. • **Off**—Indicates that the router can run IGMP or MLD on the interface and send or receive control traffic such as IGMP or MLD reports, queries, and leaves. The **passive** statement enables you to selectively activate up to two out of a possible three available query or control traffic options. When enabled, the following options appear after the **on** state declaration: • **send-general-query**—The interface sends general queries. • **send-group-query**—The interface sends group-specific and group-source-specific queries. • **allow-receive**—The interface receives control traffic	All levels
OIF map	Name of the OIF map associated to the interface.	All levels
SSM map	Name of the source-specific multicast (SSM) map used on the interface, if configured.	All levels
Group limit	Maximum number of groups allowed on the interface. Any memberships requested after the limit is reached are rejected.	All levels
Group threshold	Configured threshold at which a warning message is generated. This threshold is based on a percentage of groups received on the interface. If the number of groups received reaches the configured threshold, the device generates a warning message.	All levels
Group log-interval	Time (in seconds) between consecutive log messages.	All levels
Immediate Leave	State of the immediate leave option: • **On**—Indicates that the router removes a host from the multicast group as soon as the router receives a multicast listener done message from a host associated with the interface. • **Off**—Indicates that after receiving a multicast listener done message, instead of removing a host from the multicast group immediately, the router sends a group query to determine if another receiver responds.	All levels

Table 54: show mld interface Output Fields (continued)

Field Name	Field Description	Level of Output
Distributed	State of MLD, which, by default, takes place on the Routing Engine for MX Series routers but can be distributed to the Packet Forwarding Engine to provide faster processing of join and leave events. • **On**—distributed MLD is enabled.	All levels
Configured Parameters	Information configured by the user. • **MLD Query Interval (.1 secs)**—Interval at which this router sends membership queries when it is the querier. • **MLD Query Response Interval (.1 secs)**—Time that the router waits for a report in response to a general query. • **MLD Last Member Query Interval (.1 secs)**—Time that the router waits for a report in response to a group-specific query. • **MLD Robustness Count**—Number of times the router retries a query.	All levels
Derived Parameters	Derived information. • **MLD Membership Timeout (.1 secs)**—Timeout period for group membership. If no report is received for these groups before the timeout expires, the group membership will be removed. • **MLD Other Querier Present Timeout (.1 secs)**—Time that the router waits for the IGMP querier to send a query.	All levels

Sample Output

show mld interface

```
user@host> show mld interface
Interface: fe-0/0/0
    Querier: None
    State: Up         Timeout:       0    Version:  1    Groups:      0
    SSM Map Policy: ssm-policy-A
Interface: at-0/3/1.0
    Querier:  8038::c0a8:c345
    State: Up         Timeout:    None    Version:  1    Groups:      0
    SSM Map Policy: ssm-policy-B
Interface: fe-1/0/1.0
    Querier:  ::192.168.195.73
    State: Up         Timeout:    None    Version:  1    Groups:      3
    SSM Map Policy: ssm-policy-C
    SSM map: ipv6map1
Immediate Leave: On

Promiscuous Mode: Off
Passive: Off
Distributed: OnConfigured Parameters:

Configured Parameters:
MLD Query Interval (.1 secs): 1250
MLD Query Response Interval (.1 secs): 100
MLD Last Member Query Interval (.1 secs): 10
MLD Robustness Count: 2
```

```
Derived Parameters:
MLD Membership Timeout (.1secs): 2600
MLD Other Querier Present Timeout (.1 secs): 2550
```

show mld interface brief

The output for the **show mld interface brief** command is identical to that for the **show mld interface** command. For sample output, see show mld interface on page 1509.

show mld interface detail

The output for the **show mld interface detail** command is identical to that for the **show mld interface** command. For sample output, see show mld interface on page 1509.

show mld interface <interface-name>

```
user@host# show mld interface ge-3/2/0.0
Interface: ge-3/2/0.0
  Querier: 203.0.113.111
  State: Up Timeout:    None Version:  3 Groups:      1
  Group limit: 8
  Group threshold: 60
  Group log-interval: 10
  Immediate leave:  Off
  Promiscuous mode:  Off   Distributed: On
```

show mld statistics

Syntax

show mld statistics
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced before Junos OS Release 7.4.

Description

Display information about Multicast Listener Discovery (MLD) statistics.

Options

none—Display MLD statistics for all interfaces.

interface *interface-name*—(Optional) Display statistics about the specified interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- clear mld statistics on page 1405

List of Sample Output

show mld statistics on page 1512
show mld statistics interface on page 1513

Output Fields

Table 55 on page 1511 describes the output fields for the **show mld statistics** command. Output fields are listed in the approximate order in which they appear.

Table 55: show mld statistics Output Fields

Field Name	Field Description
Received	Number of received packets.
Sent	Number of transmitted packets.
Rx errors	Number of received packets that contained errors.

Table 55: show mld statistics Output Fields (continued)

Field Name	Field Description
MLD Message type	Summary of MLD statistics. • **Listener Query (v1/v2)**—Number of membership queries sent and received. • **Listener Report (v1)**—Number of version 1 membership reports sent and received. • **Listener Done (v1/v2)**—Number of Listener Done messages sent and received. • **Listener Report (v2)**—Number of version 2 membership reports sent and received. • **Other Unknown types**—Number of unknown message types received. • **MLD v2 source required for SSM**—Number of MLD version 2 messages received that contained no source. • **MLD v2 mode not applicable for SSM**—Number of MLD version 2 messages received that did not contain a mode applicable for source-specific multicast (SSM).
MLD Global Statistics	Summary of MLD statistics for all interfaces. • **Bad Length**—Number of messages received with length errors so severe that further classification could not occur. • **Bad Checksum**—Number of messages received with an invalid IP checksum. No further classification was performed. • **Bad Receive If**—Number of messages received on an interface not enabled for MLD. • **Rx non-local**—Number of messages received from nonlocal senders. • **Timed out**—Number of groups that timed out as a result of not receiving an explicit leave message. • **Rejected Report**—Number of reports dropped because of the MLD group policy. • **Total Interfaces**—Number of interfaces configured to support IGMP.

Sample Output

show mld statistics

```
user@host> show mld statistics
MLD packet statistics for all interfaces
MLD Message type            Received       Sent  Rx errors
Listener Query (v1/v2)             0          2          0
Listener Report (v1)               0          0          0
Listener Done (v1/v2)              0          0          0
Listener Report (v2)               0          0          0
Other Unknown types                                      0
MLD v2 source required for SSM     2
MLD v2 mode not applicable for SSM 0

MLD Global Statistics
Bad Length                         0
Bad Checksum                       0
Bad Receive If                     0
Rx non-local                       0
```

```
Timed out                          0
Rejected Report                    0
Total Interfaces                   2
```

show mld statistics interface

```
user@host> show mld statistics interface fe-1/0/1.0
MLD interface packet statistics for fe-1/0/1.0
MLD Message type             Received       Sent  Rx errors
Listener Query (v1/v2)              0          2          0
Listener Report (v1)                0          0          0
Listener Done (v1/v2)               0          0          0
Listener Report (v2)                0          0          0
Other Unknown types                                       0
MLD v2 source required for SSM     2
MLD v2 mode not applicable for SSM 0

MLD Global Statistics
Bad Length                         0
Bad Checksum                       0
Bad Receive If                     0
Rx non-local                       0
Timed out                          0
Rejected Report                    0
Total Interfaces                   2
```

show mld snooping interface

Syntax

show mld snooping interface
<brief | detail>
<instance *routing-instance*>
<*interface-name*>
<qualified-vlan *vlan-name*>
<vlan *vlan-name*>

Release Information

Command introduced in Junos OS Release 13.3 for EX Series switches.
Command introduced in Junos OS Release 14.2 for MX Series routers with MPC.

Description

Display MLD snooping information for an interface.

Options

none—Display MLD snooping information for all interfaces on which MLD snooping is enabled.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

instance *routing-instance*—(Optional) Display MLD snooping information for the specified routing instance.

interface-name—(Optional) Display MLD snooping information for the specified interface.

qualified-vlan *vlan-name*—(Optional) Display MLD snooping information for the specified qualified VLAN.

vlan *vlan-name*—(Optional) Display MLD snooping information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- *clear mld snooping membership*
- *clear mld snooping statistics*
- Verifying MLD Snooping on Switches on page 170
- Configuring MLD Snooping on a Switch VLAN with ELS Support (CLI Procedure) on page 142

List of Sample Output

show mld snooping interface on page 1515
show mld snooping interface ge-0/0/2.0 on page 1516
show mld snooping interface brief on page 1516
show mld snooping interface detail on page 1516

Output Fields

Table 56 on page 1515 lists the output fields for the **show mld snooping interface** command. Output fields are listed in the approximate order in which they appear. Details may differ for EX switches and MX routers.

Table 56: show mld snooping interface Output Fields

Field Name	Field Description	Level of Output
Instance	Routing instance for MLD snooping.	All levels
Learning Domain	Learning domain for MLD snooping.	All levels
Vlan	Name of the VLAN for which MLD snooping is enabled.	All levels
Interface	Name of the interface.	All levels
State	State of the interface: **Up** or **Down**.	**detail**, none
Groups	Number of multicast groups on the interface.	**detail**, none
Immediate leave	State of the immediate leave option: • **On**—Indicates that the MLD querier removes a host from the multicast group as soon as it receives a leave report from a host associated with the interface. • **Off**—Indicates that after receiving a leave report, instead of removing a host from the multicast group immediately, the MLD querier sends a group query to determine if there are any other hosts on that interface still interested in the multicast group.	**detail**, none
Router interface	Indicates whether the interface is a multicast router interface: **Yes** or **No**.	**detail**
Configured Parameters	Information configured by the user. • **MLD Query Interval**—Interval (in seconds) at which the MLD querier sends membership queries. • **MLD Query Response Interval**—Time (in seconds) that the MLD querier waits for a report in response to a general query. • **MLD Last Member Query Interval**—Time (in seconds) that the MLD querier waits for a report in response to a group-specific query. • **MLD Robustness Count**—Number of times the MLD querier retries a query.	All levels

Sample Output

show mld snooping interface

```
user@switch> show mld snooping interface
Instance: default-switch

Vlan: v100

Learning-Domain: default
Interface: ge-0/0/1.0
    State:         Up Groups:      1
    Immediate leave: Off
    Router interface: no
Interface: ge-0/0/2.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no
```

```
Configured Parameters:
MLD Query Interval: 125.0
MLD Query Response Interval: 10.0
MLD Last Member Query Interval: 1.0
MLD Robustness Count: 2
```

show mld snooping interface ge-0/0/2.0

```
user@switch> show mld snooping interface ge-0/0/2.0
Instance: default-switch

Vlan: v100

Learning-Domain: default
Interface: ge-0/0/2.0
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: no

Configured Parameters:
MLD Query Interval: 125.0
MLD Query Response Interval: 10.0
MLD Last Member Query Interval: 1.0
MLD Robustness Count: 2
```

show mld snooping interface brief

```
user@switch> show mld snooping interface brief
Instance: default-switch

Vlan: v1

Learning-Domain: default
Interface: ge-0/0/1.0
Interface: ge-0/0/2.0

Configured Parameters:
MLD Query Interval: 125.0
MLD Query Response Interval: 10.0
MLD Last Member Query Interval: 1.0
MLD Robustness Count: 2
```

show mld snooping interface detail

The output for the **show mld snooping interface detail** command is identical to that for the **show mld snooping interface** command. For sample output, see show mld snooping interface on page 1515.

show mld-snooping membership

Syntax

show mld-snooping membership
<**brief | detail**>
<interface *logical-interface-name*>
<vlan (*vlan-id* | *vlan-name*) >

Release Information

Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description

Display the multicast group membership information maintained by MLD snooping.

Options

none—Display the multicast group membership information for all VLANs on which MLD snooping is enabled.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

interface *interface-name*—(Optional) Display the multicast group membership information for the specified interface.

vlan (*vlan-id* | *vlan-name*)—(Optional) Display the multicast group membership for the specified VLAN.

Required Privilege Level

view

Related Documentation

- Understanding MLD Snooping on page 125
- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- **mld-snooping on page 1155**
- **clear mld-snooping membership on page 1403**
- **show mld-snooping statistics on page 1523**
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

List of Sample Output

show mld-snooping membership on page 1518
show mld-snooping membership detail on page 1519

Output Fields

Table 57 on page 1517 lists the output fields for the **show mld-snooping membership** command. Output fields are listed in the approximate order in which they appear.

Table 57: show mld-snooping membership Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All

Table 57: show mld-snooping membership Output Fields (continued)

Field Name	Field Description	Level of Output
Interfaces	Interfaces that are members of the listed multicast group.	**brief**
Tag	Numerical identifier of the VLAN.	**detail**
Router interfaces	List of information about multicast-router interfaces: • Name of the multicast-router interface. • **static** or **dynamic**—Whether the multicast-router interface has been statically configured or dynamically learned. • **Uptime**—For static interfaces, amount of time since the interface was configured as a multicast-router interface or since the interface last flapped. For dynamic interfaces, amount of time since the first query was received on the interface or since the interface last flapped. • **timeout**—Seconds remaining before a dynamic multicast-router interface times out.	**detail**
Group	IP multicast address of the multicast group. The following information is provided for the multicast group: • Name of the interface belonging to the multicast group. • **Timeout**—Time (in seconds) left until a dynamically learned interface is removed from the multicast group if no MLD membership reports are received on the interface. This counter is reset to its maximum value when a membership report is received. • **Flags**—The lowest MLD version in use by a host that is a member of the group on the interface. If the flag **static** is included, the interface has been configured as static member of the multicast group. • **Receiver count**—Number of hosts on the interface that are members of the multicast group. This field appears only if immediate-leave is configured on the VLAN. • **Last reporter**—Last host to report membership for the multicast group. • **Include source**—Multicast source addresses from all MLDv2 membership reports received for the group on the interface.	**detail**

Sample Output

show mld-snooping membership

```
user@host> show mld-snooping membership
VLAN: mld_vlan
    2001:db8:ff1e::2010
        Interfaces: ge-1/0/30.0
    2001:db8:ff1e::2011
```

```
        Interfaces: ge-1/0/30.0
    2001:db8:ff1e::2012
        Interfaces: ge-1/0/30.0
    2001:db8:ff1e::2013
        Interfaces: ge-1/0/30.0
    2001:db8:ff1e::2014
        Interfaces: ge-1/0/30.0
```

show mld-snooping membership detail

```
user@host> show mld-snooping membership detail
VLAN: mld-vlan Tag: 100 (Index: 3)
    Router interfaces:
        ge-1/0/0.0 static Uptime: 00:57:13
   Group: 2001:db8:ff1e::2010
     ge-1/0/30.0 Timeout: 180 Flags: <V2-hosts>
     Last reporter: 2001:db8:2020:1:1:3
       Include source: 2001:db8:1:1::2
VLAN: mld-vlan1 Tag: 200 (Index: 4)
    Router interfaces:
        ae200.0 dynamic Uptime: 00:14:24 timeout: 244
   Group: 2001:db8:ff1e::2010
     ge-12/0/31.0 Timeout: 224 Flags: <V1-hosts>
     Last reporter: 2001:db8:2020:1:1:4
```

show mld-snooping route

Syntax

show mld-snooping route
<**brief** | **detail**>
<**ethernet-switching** | **inet6**>
<**vlan** (*vlan-id* | *vlan-name*)>

Release Information

Command introduced in Junos OS Release 12.1 for EX Series switches.

Description

Display multicast route information maintained by MLD snooping.

Options

none—Display route information for all VLANs on which MLD snooping is enabled.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

ethernet-switching—(Optional) Display information on Layer 2 IPv6 multicast routes. This is the default.

inet6—(Optional) Display information on Layer 3 IPv6 multicast routes.

vlan (*vlan-id* | *vlan-name*) —(Optional) Display route information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- show mld-snooping membership on page 1517
- show mld-snooping statistics on page 1523
- show mld-snooping vlans on page 1525
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

List of Sample Output

show mld-snooping route on page 1521
show mld-snooping route detail on page 1521
show mld-snooping route inet6 detail on page 1522

Output Fields

Table 58 on page 1520 lists the output fields for the **show mld-snooping route** command. Output fields are listed in the approximate order in which they appear.

Table 58: show mld-snooping route Output Fields

Field Name	Field Description
Table	Routing table ID for virtual routing instances.
Routing Table	Routing table ID for virtual routing instances.

Table 58: show mld-snooping route Output Fields (continued)

Field Name	Field Description
VLAN	Name of the VLAN on which MLD snooping is enabled.
Group	Multicast IPv6 group address. Only the last 32 bits of the address are shown. The switch uses only these bits in determining multicast routes.
Next-hop	ID associated with the next-hop device.
Routing next-hop	ID associated with the Layer 3 next-hop device.
Interface or **Interfaces**	Name of the interface or interfaces in the VLAN associated with the multicast group.
Layer 2 next-hop	ID associated with the Layer 2 next-hop device.

Sample Output

show mld-snooping route

```
user@switch> show mld-snooping route

VLAN            Group          Next-hop
vlan1           ::0000:0001        1464
vlan1           ff00::
vlan10          ::0000:0002        1599
vlan10          ff00::
vlan11          ::0000:0002        1513
vlan11          ff00::
vlan12          ff00::
vlan13          ff00::
vlan14          ff00::
vlan15          ff00::
vlan16          ff00::
vlan17          ff00::
vlan18          ff00::
vlan19          ff00::
vlan2           ff00::
vlan20          ::0000:0002        1602
vlan20          ff00::
vlan3           ff00::
vlan4           ff00::
vlan5           ff00::
vlan6           ff00::
vlan7           ff00::
vlan8           ff00::
vlan9           ff00::
default         ff00::
```

show mld-snooping route detail

```
user@switch> show mld-snooping route detail
```

```
VLAN             Group          Next-hop
mld-vlan         ::0000:2010        1323
        Interfaces: ge-1/0/30.0
VLAN             Group          Next-hop
mld-vlan         ff00::             1317
        Interfaces: ge-1/0/0.0
VLAN             Group          Next-hop
mld-vlan         ::0000:0000        1317
        Interfaces: ge-1/0/0.0
VLAN             Group          Next-hop
mld-vlan1        ::0000:2010        1324
        Interfaces: ge-12/0/31.0
VLAN             Group          Next-hop
mld-vlan1        ff00::             1318
        Interfaces: ae200.0
VLAN             Group          Next-hop
mld-vlan1        ::0000:0000        1318
        Interfaces: ae200.0
```

show mld-snooping route inet6 detail

```
user@switch> show mld-snooping route inet6 detail
Routing table: 0
Group: ff05::1, 4001::11
    Routing next-hop: 1352
        vlan.2
    Interface: vlan.2, VLAN: vlan2, Layer 2 next-hop: 1387
```

show mld-snooping statistics

Syntax **show mld-snooping statistics**

Release Information Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devies.

Description Display MLD snooping statistics.

Required Privilege Level view

Related Documentation

- Understanding MLD Snooping on page 125
- Example: Configuring MLD Snooping on SRX Series Devices on page 151
- **mld-snooping on page 1155**
- **clear mld-snooping statistics on page 1404**
- **show mld-snooping membership on page 1517**
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167

List of Sample Output **show mld-snooping statistics on page 1524**

Output Fields Table 59 on page 1523 lists the output fields for the **show mld-snooping statistics** command. Output fields are listed in the approximate order in which they appear.

Table 59: show mld-snooping statistics Output Fields

Field Name	Field Description
Bad length	MLD packet has illegal or bad length.
Bad checksum	MLD or IP checksum is incorrect.
Invalid interface	Packet was received through an invalid interface.
Not Local	Not used—always 0.
Receive unknown	Unknown MLD message type.
Timed out	Not used—always 0.
MLD Type	Type of MLD message (Query, Report, Leaves, or Other).
Received	Number of MLD packets received.
Transmitted	Number of MLD packets transmitted.

Table 59: show mld-snooping statistics Output Fields (continued)

Field Name	Field Description
Recv Errors	Number of packets received that did not conform to the MLD version 1 (MLDv1) or MLDv2 standards.

Sample Output

show mld-snooping statistics

```
user@host> show mld-snooping statistics
Bad length: 0 Bad checksum: 0 Invalid interface: 0
Not local: 0 Receive unknown: 0 Timed out: 0

    MLD Type           Received      Transmitted    Recv Errors
    Queries:              74295                0              0
    Reports:           18148423                0       16333523
    Leaves:                   0                0              0
    Other:                    0                0              0
```

show mld-snooping vlans

Syntax

show mld-snooping vlans
<**brief** | **detail**>
<vlan *vlan-name*>

Release Information

Command introduced in Junos OS Release 12.1 for EX Series switches.
Command introduced in Junos OS Release 18.1R1 for the SRX1500 devices.

Description

Display MLD snooping information for a VLAN or for all VLANs.

Options

none—Display MLD snooping information for all VLANs on which MLD snooping is enabled.

brief | detail—(Optional) Display the specified level of output. The default is **brief**.

vlan *vlan-name* —(Optional) Display MLD snooping information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- mld-snooping on page 1155
- show mld-snooping membership on page 1517
- show mld-snooping route on page 1520
- show mld-snooping statistics on page 1523
- Verifying MLD Snooping on EX Series Switches (CLI Procedure) on page 167
- Configuring MLD Snooping on an EX Series Switch VLAN (CLI Procedure) on page 134

List of Sample Output

show mld-snooping vlans on page 1526
show mld-snooping vlans vlan v10 on page 1526
show mld-snooping vlans vlan vlan2 detail on page 1526
show mld-snooping vlans detail on page 1527

Output Fields

Table 56 on page 1515 lists the output fields for the **show mld-snooping vlans** command. Output fields are listed in the approximate order in which they appear.

Table 60: show mld-snooping vlans Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All levels
Interfaces	Number of interfaces in the VLAN.	**brief**
Groups	Number of groups in the VLAN.	**brief**

Table 60: show mld-snooping vlans Output Fields (continued)

Field Name	Field Description	Level of Output
MRouters	Number of multicast-router interfaces in the VLAN.	**brief**
Receivers	Number of interfaces in the VLAN with a receiver for any group. Indicates how many interfaces might receive data because of MLD group membership.	**brief**
Tag	VLAN tag.	**detail**
vlan-interface	The Layer 3 interface, if any, associated with the VLAN.	**detail**
Interface	Name of the interface. The following information is provided for each interface: • **tagged** or **untagged**—Whether the interface accepts tagged packets (trunk mode and tagged-access mode ports) or untagged packets (access mode ports) • **Groups**—The number of multicast groups the interface belongs to • **Reporters**—The number of hosts on the interface that are current members of multicast groups. This field appears only when immediate-leave is configured on the VLAN. • **Router**—Indicates the interface is a multicast-router interface	**detail**

Sample Output

show mld-snooping vlans

```
user@host> show mld-snooping vlans
VLAN           Interfaces Groups MRouters Receivers
default                 0      0        0         0
v1                     11     50        0         0
v10                     1      0        0         0
v11                     1      0        0         0
v180                    3      0        1         0
v181                    3      0        0         0
v182                    3      0        0         0
```

show mld-snooping vlans vlan v10

```
user@host> show mld-snooping vlans vlan v10
 VLAN             Interfaces Groups MRouters Receivers
v10                       3      1        1         0       0
```

show mld-snooping vlans vlan vlan2 detail

```
user@host> show mld-snooping vlans vlan vlan2 detail

VLAN: vlan2, Tag: 2, vlan-interface: vlan.2
    Interface: ge-0/0/2.0, untagged, Groups: 5
    Interface: ge-0/0/4.0, tagged, Groups: 3, Router
```

show mld-snooping vlans detail

```
user@host> show mld-snooping vlans detail
VLAN: mld-vlan, Tag: 100
    Interface: ge-1/0/0.0, untagged, Groups: 0, Router
    Interface: ge-1/0/30.0, untagged, Groups: 1
    Interface: ge-1/0/33.0, untagged, Groups: 0
    Interface: ge-12/0/30.0, untagged, Groups: 0
VLAN: mld-vlan1, Tag: 200
    Interface: ge-1/0/31.0, untagged, Groups: 0
    Interface: ge-12/0/31.0, untagged, Groups: 1
    Interface: ae200.0, untagged, Groups: 0, Router
```

show mpls lsp

List of Syntax

Syntax on page 1528
Syntax (EX Series Switches) on page 1528

Syntax

show mpls lsp
<brief | detail | extensive | terse>
<autobandwidth>
<bidirectional | unidirectional>
<bypass>
<count-active-routes>
<defaults>
<descriptions>
<down | up>
<externally-controlled>
<externally-provisioned>
<logical-system (all | *logical-system-name*)>
<*lsp-type*>
<name *name*>
<p2mp>
<statistics>
<transit>

Syntax (EX Series Switches)

show mpls lsp
<brief | detail | extensive | terse>
<bidirectional | unidirectional>
<bypass>
<descriptions>
<down | up>
<externally-controlled>
<externally-provisioned>
<*lsp-type*>
<name *name*>
<p2mp>
<statistics>
<transit>

Release Information

Command introduced before Junos OS Release 7.4.
defaults option added in Junos OS Release 8.5.
Command introduced in Junos OS Release 9.5 for EX Series switches.
autobandwidth option added in Junos OS Release 11.4.
externally-controlled option added in Junos OS Release 12.3.
externally-provisioned option added in Junos OS Release 13.3.
Command introduced in Junos OS Release 13.2X51-D15 for QFX Series.
instance *instance-name* option added in Junos OS Release 15.1.

Description

Display information about configured and active dynamic Multiprotocol Label Switching (MPLS) label-switched paths (LSPs).

Options

none—Display standard information about all configured and active dynamic MPLS LSPs.

brief | detail | extensive | terse—(Optional) Display the specified level of output. The extensive option displays the same information as the detail option, but covers the most recent 50 events.

autobandwidth—(Optional) Display automatic bandwidth information. This option is explained separately (see **show mpls lsp autobandwidth**).

bidirectional | unidirectional—(Optional) Display bidirectional or unidirectional LSP information, respectively.

bypass—(Optional) Display LSPs used for protecting other LSPs.

count-active-routes—(Optional) Display active routes for LSPs.

defaults—(Optional) Display the MPLS LSP default settings.

descriptions—(Optional) Display the MPLS label-switched path (LSP) descriptions. To view this information, you must configure the description statement at the **[edit protocol mpls lsp]** hierarchy level. Only LSPs with a description are displayed. This command is only valid for the ingress routing device, because the description is not propagated in RSVP messages.

down | up—(Optional) Display only LSPs that are inactive or active, respectively.

externally-controlled—(Optional) Display the LSPs that are under the control of an external Path Computation Element (PCE).

externally-provisioned—(Optional) Display the LSPs that are generated dynamically and provisioned by an external Path Computation Element (PCE).

instance ***instance-name***—(Optional) Display MPLS LSP information for the specified instance. If ***instance-name*** is omitted, MPLS LSP information is displayed for the master instance.

logical-system (all | ***logical-system-name*****)**—(Optional) Perform this operation on all logical systems or on a particular logical system.

lsp-type—(Optional) Display information about a particular LSP type:

- **bypass**—Sessions for bypass LSPs.
- **egress**—Sessions that terminate on this routing device.
- **ingress**—Sessions that originate from this routing device.
- **pop-and-forward**—Sessions that originate from RSVP-TE pop-and-forward LSP tunnels.
- **transit**—Sessions that pass through this routing device.

name ***name***—(Optional) Display information about the specified LSP or group of LSPs.

p2mp—(Optional) Display information about point-to-multipoint LSPs.

statistics—(Optional) (Ingress and transit routers only) Display accounting information about LSPs. Statistics are not available for LSPs on the egress routing device, because the penultimate routing device in the LSP sets the label to 0. Also, as the packet arrives at the egress routing device, the hardware removes its MPLS header and the packet reverts to being an IPv4 packet. Therefore, it is counted as an IPv4 packet, not an MPLS packet.

NOTE: If a bypass LSP is configured for the primary static LSP, display cumulative statistics of packets traversing through the protected LSP and bypass LSP when traffic is re-optimized when the protected LSP link is restored. (Bypass LSPs are not supported on QFX Series switches.)

When used with the **bypass** option (**show mpls lsp bypass statistics**), display statistics for the traffic that flows only through the bypass LSP.

transit—(Optional) Display LSPs transiting this routing device.

Required Privilege Level

view

Related Documentation

- *clear mpls lsp*
- *show mpls lsp autobandwidth*

List of Sample Output

show mpls lsp defaults on page 1538
show mpls lsp descriptions on page 1538
show mpls lsp detail on page 1538
show mpls lsp detail (When Egress Protection Is in Standby Mode) on page 1539
show mpls lsp detail (When Egress Protection Is in Effect During a Local Repair) on page 1540
show mpls lsp extensive on page 1541
show mpls lsp ingress extensive on page 1542
show mpls lsp extensive (automatic bandwidth adjustment enabled) on page 1543
show mpls lsp bypass extensive on page 1543
show mpls lsp p2mp on page 1544
show mpls lsp p2mp detail on page 1544
show mpls lsp detail count-active-routes on page 1545
show mpls lsp statistics extensive on page 1545

Output Fields

Table 61 on page 1531 describes the output fields for the **show mpls lsp** command. Output fields are listed in the approximate order in which they appear.

Table 61: show mpls lsp Output Fields

Field Name	Field Description	Level of Output
Ingress LSP	Information about LSPs on the ingress routing device. Each session has one line of output.	All levels
Egress LSP	Information about the LSPs on the egress routing device. MPLS learns this information by querying RSVP, which holds all the transit and egress session information. Each session has one line of output.	All levels
Transit LSP	Number of LSPs on the transit routing devices and the state of these paths. MPLS learns this information by querying RSVP, which holds all the transit and egress session information.	All levels
P2MP name	Name of the point-to-multipoint LSP. Dynamically generated P2MP LSPs used for VPLS flooding use dynamically generated P2MP LSP names. The name uses the format ***identifier*:vpls:*router-id*:*routing-instance-name***. The ***identifier*** is automatically generated by Junos OS.	All levels
P2MP branch count	Number of destination LSPs the point-to-multipoint LSP is transmitting to.	All levels
P	An asterisk (*) under this heading indicates that the LSP is a primary path.	All levels
address	(**detail** and **extensive**) Destination (egress routing device) of the LSP.	**detail extensive**
To	Destination (egress routing device) of the session.	**brief**
From	Source (ingress routing device) of the session.	**brief detail**
State	State of the LSP handled by this RSVP session: **Up**, **Dn** (down), or **Restart**.	**brief detail**
Active Route	Number of active routes (prefixes) installed in the forwarding table. For ingress LSPs, the forwarding table is the primary IPv4 table (**inet.0**). For transit and egress RSVP sessions, the forwarding table is the primary MPLS table (**mpls.0**).	**detail extensive**
Rt	Number of active routes (prefixes) installed in the routing table. For ingress RSVP sessions, the routing table is the primary IPv4 table (**inet.0**). For transit and egress RSVP sessions, the routing table is the primary MPLS table (mpls.0).	**brief**
P	Path. An asterisk (*) underneath this column indicates that the LSP is a primary path.	**brief**
ActivePath	(Ingress LSP) Name of the active path: **Primary** or **Secondary**.	**detail extensive**
LSPname	Name of the LSP.	**brief detail**
Statistics	Displays the number of packets and the number of bytes transmitted over the LSP. These counters are reset to zero whenever the LSP path is optimized (for example, during an automatic bandwidth allocation).	**extensive**
Aggregate statistics	Displays the number of packets and the number of bytes transmitted over the LSP. These counters continue to iterate even if the LSP path is optimized. You can reset these counters to zero using the **clear mpls lsp statistics** command.	**extensive**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
Packets	Displays the number of packets transmitted over the LSP.	**brief extensive**
Bytes	Displays the number of bytes transmitted over the LSP.	**brief extensive**
DiffServeInfo	Type of LSP: multiclass LSP **(multiclass diffServ-TE LSP)** or Differentiated-Services-aware traffic engineering LSP **(diffServ-TE LSP)**.	**detail**
LSPtype	Type of LSP: • **Static configured**—Static • **Dynamic configured**—Dynamic • **Externally controlled**—External path computing entity Also indicates if the LSP is a **Penultimate hop popping** LSP or an **Ultimate hop popping** LSP.	**detail extensive**
Bypass	(Bypass LSP) Destination address (egress routing device) for the bypass LSP.	All levels
LSPpath	Indicates whether the RSVP session is for the primary or secondary LSP path. **LSPpath** can be either **primary** or **secondary** and can be displayed on the ingress, egress, and transit routing devices.	**detail**
Bidir	(GMPLS) The LSP allows data to travel in both directions between GMPLS devices.	All levels
Bidirectional	(GMPLS) The LSP allows data to travel both ways between GMPLS devices.	All levels
FastReroute desired	Fast reroute has been requested by the ingress routing device.	**detail**
Link protection desired	Link protection has been requested by the ingress routing device.	**detail**
Node/Link protection desired	Link protection has been requested by the ingress routing device.	**detail**
LSP Control Status	(Ingress LSP) LSP control mode: • External—By default, all PCE-controlled LSPs are under external control. When an LSP is under external control, the PCC uses the PCE-provided parameters to set up the LSP. • Local—A PCE-controlled LSP can come under local control. When the LSP switches from external control to local control, path computation is done using the CLI-configured parameters and constraint-based routing. Such a switchover happens only when there is a trigger to re-signal the LSP. Until then, the PCC uses the PCE-provided parameters to signal the PCE-controlled LSP, although the LSP remains under local control. A PCE-controlled LSP switches to local control from its default external control mode in cases such as no connectivity to a PCE or when a PCE returns delegation of LSPs back to the PCC.	**extensive**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
External Path CSPF status	(PCE-controlled LSPs) Status of the PCE-controlled LSP with per path attributes: • Local • External	**extensive**
Externally Computed ERO	(PCE-controlled LSPs) Externally computed explicit route when the route object is not null or empty. A series of hops, each with an address followed by a hop indicator. The value of the hop indicator can be strict (**S**) or loose (**L**).	**extensive**
EXTCTRL_LSP	(PCE-controlled LSPs) Display path history including the bandwidth, priority, and metric values received from the external controller.	**extensive**
flap counter	Counts the number of times a LSP flaps down or up.	extensive
LoadBalance	(Ingress LSP) CSPF load-balancing rule that was configured to select the LSP's path among equal-cost paths: **Most-fill**, **Least-fill**, or **Random**.	**detail extensive**
Signal type	Signal type for GMPLS LSPs. The signal type determines the peak data rate for the LSP: **DS0**, **DS3**, **STS-1**, **STM-1**, or **STM-4**.	All levels
Encoding type	LSP encoding type: **Packet**, **Ethernet**, **PDH**, **SDH/SONET**, **Lambda**, or **Fiber**.	All levels
Switching type	Type of switching on the links needed for the LSP: **Fiber**, **Lamda**, **Packet**, **TDM**, or **PSC-1**.	All levels
GPID	Generalized Payload Identifier (identifier of the payload carried by an LSP): **HDLC**, **Ethernet**, **IPv4**, **PPP**, or **Unknown**.	All levels
Protection	Configured protection capability desired for the LSP: **Extra**, **Enhanced**, **none**, **One plus one**, **One to one**, or **Shared**.	All levels
Upstream label in	(Bidirectional LSPs) Incoming label for reverse direction traffic for this LSP.	All levels
Upstream label out	(Bidirectional LSPs) Outgoing label for reverse direction traffic for this LSP.	All levels
Suggested label received	(Bidirectional LSPs) Label the upstream interface suggests to use in the Resv message that is sent.	All levels
Suggested label sent	(Bidirectional LSPs) Label the downstream node suggests to use in the Resv message that is returned.	All levels
Autobandwidth	(Ingress LSP) The LSP is performing autobandwidth allocation.	**detail extensive**
Mbb counter	Counts the number of times a LSP incurs MBB.	extensive
MinBW	(Ingress LSP) Configured minimum value of the LSP, in bps.	**detail extensive**
MaxBW	(Ingress LSP) Configured maximum value of the LSP, in bps.	**detail extensive**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
Dynamic MinBW	(Ingress LSP) Displays the current dynamically specified minimum bandwidth allocation for the LSP, in bps.	**detail extensive**
Dynamic MinBW	(Ingress LSP) Displays the current dynamically specified minimum bandwidth allocation for the LSP, in bps.	**detail extensive**
AdjustTimer	(Ingress LSP) Configured value for the **adjust-timer** statement, indicating the total amount of time allowed before bandwidth adjustment will take place, in seconds.	**detail extensive**
Adjustment Threshold	(Ingress LSP) Configured value for the **adjust-threshold** statement. Specifies how sensitive the automatic bandwidth adjustment for an LSP is to changes in bandwidth utilization.	**detail extensive**
Time for Next Adjustment	(Ingress LSP) Time in seconds until the next automatic bandwidth adjustment sample is taken.	**detail extensive**
Time of Last Adjustment	(Ingress LSP) Date and time since the last automatic bandwidth adjustment was completed.	**detail extensive**
MaxAvgBW util	(Ingress LSP) Current value of the actual maximum average bandwidth utilization, in bps.	**detail extensive**
Overflow limit	(Ingress LSP) Configured value of the threshold overflow limit.	**detail extensive**
Overflow sample count	(Ingress LSP) Current value for the overflow sample count.	**detail extensive**
Bandwidth Adjustment in *nnn* second(s)	(Ingress LSP) Current value of the bandwidth adjustment timer, indicating the amount of time remaining until the bandwidth adjustment will take place, in seconds.	**detail extensive**
Underflow limit	(Ingress LSP) Configured value of the threshold underflow limit.	**detail extensive**
Underflow sample count	(Ingress LSP) Current value for the underflow sample count.	**detail extensive**
Underflow Max AvgBW	(Ingress LSP) The highest sample bandwidth among the underflow samples recorded currently. This is the signaling bandwidth if an adjustment occurs because of an underflow.	**detail extensive**
Active path indicator	(Ingress LSP) A value of * indicates that the path is active. The absence of * indicates that the path is not active. In the following example, “long” is the active path. ***Primary long** **Standby short**	**detail extensive**
Primary	(Ingress LSP) Name of the primary path.	**detail extensive**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
Secondary	(Ingress LSP) Name of the secondary path.	**detail extensive**
Standby	(Ingress LSP) Name of the path in standby mode.	**detail extensive**
State	(Ingress LSP) State of the path: **Up** or **Dn** (down).	**detail extensive**
COS	(Ingress LSP) Class-of-service value.	**detail extensive**
Bandwidth per class	(Ingress LSP) Active bandwidth for the LSP path for each MPLS class type, in bps.	**detail extensive**
Priorities	(Ingress LSP) Configured value of the setup priority and the hold priority respecitively (the setup priority is displayed first), where 0 is the highest priority and 7 is the lowest priority. If you have not explicitly configured these values, the default values are displayed (7 for the setup priority and 0 for the hold priority).	**detail extensive**
OptimizeTimer	(Ingress LSP) Configured value of the optimize timer, indicating the total amount of time allowed before path reoptimization, in seconds.	**detail extensive**
SmartOptimizeTimer	(Ingress LSP) Configured value of the smart optimize timer, indicating the total amount of time allowed before path reoptimization, in seconds.	**detail extensive**
Reoptimization in xxx seconds	(Ingress LSP) Current value of the optimize timer, indicating the amount of time remaining until the path will be reoptimized, in seconds.	**detail extensive**
Computed ERO (S [L] denotes strict [loose] hops)	(Ingress LSP) Computed explicit route. A series of hops, each with an address followed by a hop indicator. The value of the hop indicator can be strict (**S**) or loose (**L**).	**detail extensive**
CSPF metric	(Ingress LSP) Constrained Shortest Path First metric for this path.	**detail extensive**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
Received RRO	(Ingress LSP) Received record route. A series of hops, each with an address followed by a flag. (In most cases, the received record route is the same as the computed explicit route. If **Received RRO** is different from **Computed ERO**, there is a topology change in the network, and the route is taking a detour.) The following flags identify the protection capability and status of the downstream node: • **0x01**—Local protection available. The link downstream from this node is protected by a local repair mechanism. This flag can be set only if the Local protection flag was set in the **SESSION_ATTRIBUTE** object of the corresponding Path message. • **0x02**—Local protection in use. A local repair mechanism is in use to maintain this tunnel (usually because of an outage of the link it was routed over previously). • **0x03**—Combination of **0x01** and **0x02**. • **0x04**—Bandwidth protection. The downstream routing device has a backup path providing the same bandwidth guarantee as the protected LSP for the protected section. • **0x08**—Node protection. The downstream routing device has a backup path providing protection against link and node failure on the corresponding path section. If the downstream routing device can set up only a link-protection backup path, the **Local protection available** bit is set but the **Node protection** bit is cleared. • **0x09**—Detour is established. Combination of **0x01** and **0x08**. • **0x10**—Preemption pending. The preempting node sets this flag if a pending preemption is in progress for the traffic engine LSP. This flag indicates to the ingress legacy edge router (LER) of this LSP that it should be rerouted. • **0x20**—Node ID. Indicates that the address specified in the RRO's IPv4 or IPv6 sub-object is a node ID address, which refers to the router address or router ID. Nodes must use the same address consistently. • **0xb**—Detour is in use. Combination of **0x01**, **0x02**, and **0x08**.	**detail extensive**
Labels	Labels of pop-and-forward LSP tunnel: • P—Pop labels. • D—Delegation labels.	**extensive**
Index number	(Ingress LSP) Log entry number of each LSP path event. The numbers are in chronological descending order, with a maximum of 50 index numbers displayed.	**extensive**
Date	(Ingress LSP) Date of the LSP event.	**extensive**
Time	(Ingress LSP) Time of the LSP event.	**extensive**
Event	(Ingress LSP) Description of the LSP event.	**extensive**
Created	(Ingress LSP) Date and time the LSP was created.	**extensive**
Resv style	(Bypass) RSVP reservation style. This field consists of two parts. The first is the number of active reservations. The second is the reservation style, which can be **FF** (fixed filter), **SE** (shared explicit), or **WF** (wildcard filter).	**brief detail extensive**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
Labelin	Incoming label for this LSP.	**brief detail**
Labelout	Outgoing label for this LSP.	**brief detail**
LSPname	Name of the LSP.	**brief detail**
Time left	Number of seconds remaining in the lifetime of the reservation.	**detail**
Since	Date and time when the RSVP session was initiated.	**detail**
Tspec	Sender's traffic specification, which describes the sender's traffic parameters.	**detail**
Port number	Protocol ID and sender or receiver port used in this RSVP session.	**detail**
PATH rcvfrom	Address of the previous-hop (upstream) routing device or client, interface the neighbor used to reach this router, and number of packets received from the upstream neighbor.	**detail**
PATH sentto	Address of the next-hop (downstream) routing device or client, interface used to reach this neighbor, and number of packets sent to the downstream routing device.	**detail**
RESV rcvfrom	Address of the previous-hop (upstream) routing device or client, interface the neighbor used to reach this routing device, and number of packets received from the upstream neighbor. The output in this field, which is consistent with that in the **PATH rcvfrom** field, indicates that the RSVP negotiation is complete.	**detail**
Record route	Recorded route for the session, taken from the record route object.	**detail**
Pop-and-forward	Attributes of the pop-and-forward LSP tunnel.	**extensive**
ETLD In	Number of transport labels that the LSP-Hop can potentially receive from its upstream hop. It is recorded as Effective Transport Label Depth (ETLD) at the transit and egress devices.	**extensive**
ETLD Out	Number of transport labels the LSP-Hop can potentially send to its downstream hop. It is recorded as ETLD at the transit and ingress devices.	**extensive**
Delegation hop	Specifies if the transit hop is selected as a delegation label: • Yes • No	**extensive**
Soft preempt	Number of soft preemptions that occurred on a path and when the last soft preemption occurred. Only successful soft preemptions are counted (those that actually resulted in a new path being used).	**detail**
Soft preemption pending	Path is in the process of being soft preempted. This display is removed once the ingress router has calculated a new path.	**detail**

Table 61: show mpls lsp Output Fields (continued)

Field Name	Field Description	Level of Output
MPLS-TE LSP Defaults	Default settings for MPLS traffic engineered LSPs: • **LSP Holding Priority**—Determines the degree to which an LSP holds on to its session reservation after the LSP has been set up successfully. • **LSP Setup Priority**—Determines whether a new LSP that preempts an existing LSP can be established. • **Hop Limit**—Specifies the maximum number of routers the LSP can traverse (including the ingress and egress). • **Bandwidth**—Specifies the bandwidth in bits per second for the LSP. • **LSP Retry Timer**—Length of time in seconds that the ingress router waits between attempts to establish the primary path.	**defaults**

The XML tag name of the **bandwidth** tag under the **auto-bandwidth** tag has been updated to **maximum-average-bandwidth** . You can see the new tag when you issue the **show mpls lsp extensive** command with the **| display xml** pipe option. If you have any scripts that use the **bandwidth** tag, ensure that they are updated to **maximum-average-bandwidth**.

Sample Output

show mpls lsp defaults

```
user@host> show mpls lsp defaults
   MPLS-TE LSP Defaults
      LSP Holding Priority      0
      LSP Setup Priority        7
      Hop Limit                 255
      Bandwidth                 0
      LSP Retry Timer           30 seconds
```

show mpls lsp descriptions

```
user@host> show mpls lsp descriptions
Ingress LSP: 3 sessions
To              LSP name                                Description
10.0.0.195      to-sanjose                              to-sanjose-desc
10.0.0.195      to-sanjose-other-desc                   other-desc
Total 2 displayed, Up 2, Down 0
```

show mpls lsp detail

```
user@host> show mpls lsp detail
Ingress LSP: 1 sessions

192.168.0.4
  From: 192.168.0.5, State: Up, ActiveRoute: 0, LSPname: E-D
  ActivePath:  (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
```

```
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
 10.0.0.18 S 10.0.0.22 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          10.0.0.18 10.0.0.22
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

192.168.0.5
  From: 192.168.0.4, LSPstate: Up, ActiveRoute: 0
  LSPname: E-D, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  157, Since: Wed Jul 18 17:55:12 2012
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 46128 protocol 0
  PATH rcvfrom: 10.0.0.18 (lt-1/2/0.17) 3 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.0.22 10.0.0.18 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

show mpls lsp detail (When Egress Protection Is in Standby Mode)

```
user@host> show mpls lsp detail
Ingress LSP: 1 sessions

192.168.0.4
  From: 192.168.0.5, State: Up, ActiveRoute: 0, LSPname: E-D
  ActivePath:  (primary)
  LSPtype: Static Configured, Ultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
 10.0.0.18 S 10.0.0.22 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          10.0.0.18 10.0.0.22
   11 Sep 20 15:54:35.032 Make-before-break: Switched to new instance
   10 Sep 20 15:54:34.029 Record Route:  10.0.0.18 10.0.0.22
    9 Sep 20 15:54:34.029 Up
    8 Sep 20 15:54:20.271 Originate make-before-break call
    7 Sep 20 15:54:20.271 CSPF: computation result accepted  10.0.0.18 10.0.0.22

    6 Sep 20 15:52:10.247 Selected as active path
    5 Sep 20 15:52:10.246 Record Route:  10.0.0.18 10.0.0.22
    4 Sep 20 15:52:10.243 Up
    3 Sep 20 15:52:09.745 Originate Call
    2 Sep 20 15:52:09.745 CSPF: computation result accepted  10.0.0.18 10.0.0.22

    1 Sep 20 15:51:39.903 CSPF failed: no route toward 192.168.0.4
  Created: Thu Sep 20 15:51:08 2012
```

```
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

192.168.0.5
  From: 192.168.0.4, LSPstate: Up, ActiveRoute: 0
  LSPname: E-D, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  148, Since: Thu Sep 20 15:52:10 2012
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 49601 protocol 0
  PATH rcvfrom: 10.0.0.18 (lt-1/2/0.17) 27 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.0.22 10.0.0.18 <self>
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

show mpls lsp detail (When Egress Protection Is in Effect During a Local Repair)

```
user@host> show mpls lsp detail
Ingress LSP: 1 sessions

192.168.0.4
  From: 192.168.0.5, State: Up, ActiveRoute: 0, LSPname: E-D
  ActivePath:  (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
 10.0.0.18 S 10.0.0.22 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          10.0.0.18 10.0.0.22
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

192.168.0.5
  From: 192.168.0.4, LSPstate: Down, ActiveRoute: 0
  LSPname: E-D, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  157, Since: Wed Jul 18 17:55:12 2012
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 46128 protocol 0
Egress protection PLR as protector: In Use
PATH rcvfrom: 10.0.0.18 (lt-1/2/0.17) 3 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.0.22 10.0.0.18 <self>
```

```
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

show mpls lsp extensive

```
user@host> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.0.4
  From: 192.168.0.5, State: Up, ActiveRoute: 0, LSPname: E-D
  ActivePath:  (primary)
  LSPtype: Static Configured, Ultimate hop popping
  LSP Control Status: Externally controlled
  LoadBalance: Random
  Metric: 10
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    External Path CSPF status: local
    Bandwidth: 98.76kbps
    SmartOptimizeTimer: 180
    Include All: green
    Externally Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric:
0) 1.2.3.2 S 2.3.3.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          10.0.0.18 10.0.0.22
    9 May 17 16:55:06.574 EXTCTRL LSP: Sent Path computation request and LSP
status
    8 May 17 16:55:06.574 EXTCTRL_LSP: Computation request/lsp status contains:
signalled bw 98760 req BW 0 admin group(exclude 0 include any 0 include all 16)
priority setup 5 hold 4 hops: 1.2.3.2 2.3.3.2
    7 May 17 16:55:06.574 Selected as active path
    6 May 17 16:55:06.558 EXTCTRL LSP: Sent Path computation request and LSP
status
    8 May 17 16:55:06.574 EXTCTRL_LSP: Computation request/lsp status contains:
signalled bw 98760 req BW 0 admin group(exclude 0 include any 0 include all 16)
priority setup 5 hold 4 hops: 1.2.3.2 2.3.3.2
    7 May 17 16:55:06.574 Selected as active path
    6 May 17 16:55:06.558 EXTCTRL LSP: Sent Path computation request and LSP
status
    5 May 17 16:55:06.558 EXTCTRL_LSP: Computation request/lsp status contains:
signalled bw 98760 req BW 0 admin group(exclude 0 include any 0 include all 16)
priority setup 5 hold 4 hops: 1.2.3.2 2.3.3.2
    4 May 17 16:55:06.557 Record Route: 1.2.3.2 2.3.3.2
    3 May 17 16:55:06.557 Up
    2 May 17 16:55:06.382 Originate Call
    1 May 17 16:55:06.382 EXTCTRL_LSP: Received setup parameters :: local_cspf,
1.2.3.2 2.3.3.2
  Created: Tue May 17 16:55:07 2016
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions

192.168.0.5
  From: 192.168.0.4, LSPstate: Up, ActiveRoute: 0
  LSPname: E-D, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: -
```

```
  Resv style: 1 FF, Label in: 3, Label out: -
  Time left:  148, Since: Thu Sep 20 15:52:10 2012
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 49601 protocol 0
  PATH rcvfrom: 10.0.0.18 (lt-1/2/0.17) 27 pkts
  Adspec: received MTU 1500
  PATH sentto: localclient
  RESV rcvfrom: localclient
  Record route: 10.0.0.22 10.0.0.18 <self>
```

show mpls lsp ingress extensive

```
user@host> show mpls lsp ingress extensive
Ingress LSP: 1 sessions

50.0.0.1
  From: 10.0.0.1, State: Up, ActiveRoute: 0, LSPname: test
  ActivePath:  (primary)
  LSPtype: Static Pop-and-forward Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    OptimizeTimer: 300
    SmartOptimizeTimer: 180
    Reoptimization in 240 second(s).
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 3)
 1.1.1.2 S 4.4.4.1 S 5.5.5.2 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
                             (Labels: P=Pop D=Delegation)
          80.1.1.2(Label=18 P) 50.1.1.2(Label=17 P) 70.1.1.2(Label=16 P)
          92.1.1.1(Label=16 D) 93.1.1.2(Label=16 P) 99.1.1.1(Label=16 P)
          99.2.1.1(Label=16 P) 99.3.1.2(Label=3)
   17 Aug  3 13:17:33.601 CSPF: computation result ignored, new path less avail
bw[3 times]
   16 Aug  3 13:02:51.283 CSPF: computation result ignored, new path no benefit[2
 times]
   15 Aug  3 12:54:36.678 Selected as active path
   14 Aug  3 12:54:36.676 Record Route:  1.1.1.2 4.4.4.1 5.5.5.2
   13 Aug  3 12:54:36.676 Up
   12 Aug  3 12:54:33.924 Deselected as active
   11 Aug  3 12:54:33.924 Originate Call
   10 Aug  3 12:54:33.923 Clear Call
    9 Aug  3 12:54:33.923 CSPF: computation result accepted  1.1.1.2 4.4.4.1
5.5.5.2
    8 Aug  3 12:54:33.922 2.2.2.2: No Route toward dest
    7 Aug  3 12:54:28.177 CSPF: computation result ignored, new path no benefit[4
 times]
    6 Aug  3 12:35:03.830 Selected as active path
    5 Aug  3 12:35:03.828 Record Route:  2.2.2.2 3.3.3.2
    4 Aug  3 12:35:03.827 Up
    3 Aug  3 12:35:03.814 Originate Call
    2 Aug  3 12:35:03.814 CSPF: computation result accepted  2.2.2.2 3.3.3.2
    1 Aug  3 12:34:34.921 CSPF failed: no route toward 50.0.0.1
  Created: Tue Aug  3 12:34:35 2010
Total 1 displayed, Up 1, Down 0
```

show mpls lsp extensive (automatic bandwidth adjustment enabled)

```
user@host> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.0.4
  From: 192.168.0.5, State: Up, ActiveRoute: 0, LSPname: E-D
  ActivePath:  (primary)
  Node/Link protection desired
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Autobandwidth
  MinBW: 300bps, MaxBW: 1000bps, Dynamic MinBW: 1000bps
  Adjustment Timer: 300 secs AdjustThreshold: 25%
  Max AvgBW util: 963.739bps, Bandwidth Adjustment in 0 second(s).
  Min BW Adjust Interval: 1000, MinBW Adjust Threshold (in %): 50
  Overflow limit: 0, Overflow sample count: 0
  Underflow limit: 0, Underflow sample count: 9, Underflow Max AvgBW: 614.421bps

  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    Bandwidth: 1000bps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
 10.0.0.18 S 10.0.0.22 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          192.168.0.6(flag=0x20) 10.0.0.18(Label=299792) 192.168.0.4(flag=0x20)
10.0.0.22(Label=3)
   12 Apr 30 10:25:17.024 Make-before-break: Switched to new instance
   11 Apr 30 10:25:16.023 Record Route:  192.168.0.6(flag=0x20)
10.0.0.18(Label=299792) 192.168.0.4(flag=0x20) 10.0.0.22(Label=3)
   10 Apr 30 10:25:16.023 Up
    9 Apr 30 10:25:16.023 Automatic Autobw adjustment succeeded: BW changes from
 300 bps to 1000 bps
    8 Apr 30 10:25:15.946 Originate make-before-break call
    7 Apr 30 10:25:15.946 CSPF: computation result accepted  10.0.0.18 10.0.0.22

    6 Apr 30 10:16:42.891 Selected as active path
    5 Apr 30 10:16:42.891 Record Route:  192.168.0.6(flag=0x20)
10.0.0.18(Label=299776) 192.168.0.4(flag=0x20) 10.0.0.22(Label=3)
    4 Apr 30 10:16:42.890 Up
    3 Apr 30 10:16:42.828 Originate Call
    2 Apr 30 10:16:42.828 CSPF: computation result accepted  10.0.0.18 10.0.0.22

    1 Apr 30 10:16:14.064 CSPF: could not determine self[2 times]
  Created: Tue Apr 30 10:15:16 2013
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

show mpls lsp bypass extensive

```
user@host # show mpls lsp bypass extensive
```

```
Ingress LSP: 1 sessions

2.2.2.2
  From: 1.1.1.1, LSPstate: Up, ActiveRoute: 0
  LSPname: Bypass->1.1.2.2
  LSPtype: Static Configured
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 300032
  Resv style: 1 SE, Label in: -, Label out: 300032
  Time left:    -, Since: Tue Dec  3 15:19:49 2013
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 1 receiver 55750 protocol 0
  Type: Bypass LSP
    Number of data route tunnel through: 1
    Number of RSVP session tunnel through: 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 1500
  Path MTU: received 1500
  PATH sentto: 1.1.5.2 (lt-1/2/0.15) 1221 pkts
  RESV rcvfrom: 1.1.5.2 (lt-1/2/0.15) 1221 pkts, Entropy label: No
  Explct route: 1.1.5.2 1.2.5.1
  Record route: <self>  1.1.5.2 1.2.5.1
+     4 Dec  3 15:19:49 Record Route:  1.1.5.2 1.2.5.1
+    3 Dec  3 15:19:49 Up
+     2 Dec  3 15:19:49 CSPF: computation result accepted
+    1 Dec  3 15:19:47 Originate Call
Total 1 displayed, Up 1, Down 0
Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
Transit LSP: 0 sessions
```

show mpls lsp p2mp

```
user@host> show mpls lsp p2mp
Ingress LSP: 2 sessions
P2MP name: p2mp-lsp1, P2MP branch count: 1
To              From            State Rt P ActivePath       LSPname
10.255.245.51   10.255.245.50   Up     0 * path1            p2mp-branch-1
P2MP name: p2mp-lsp2, P2MP branch count: 1
To              From            State Rt P ActivePath       LSPname
10.255.245.51   10.255.245.50   Up     0 * path1            p2mp-st-br1
Total 2 displayed, Up 2, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

show mpls lsp p2mp detail

```
user@host> show mpls lsp p2mp detail
Ingress LSP: 2 sessions
P2MP name: p2mp-lsp1, P2MP branch count: 1

10.255.245.51
  From: 10.255.245.50, State: Up, ActiveRoute: 0, LSPname: p2mp-branch-1
  ActivePath: path1 (primary)
  P2MP name: p2mp-lsp1
```

```
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   path1            State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 25)
 192.168.208.17 S
   Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):

          192.168.208.17
P2MP name: p2mp-lsp2, P2MP branch count: 1

10.255.245.51
  From: 10.255.245.50, State: Up, ActiveRoute: 0, LSPname: p2mp-st-br1
  ActivePath: path1 (primary)
  P2MP name: p2mp-lsp2
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   path1            State: Up
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 25)
 192.168.208.17 S
   Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt):

          192.168.208.17
Total 2 displayed, Up 2, Down 0
```

show mpls lsp detail count-active-routes

```
user@host> show mpls lsp detail count-active-routes
Ingress LSP: 1 sessions

213.119.192.2
  From: 156.154.162.128, State: Up, ActiveRoute: 1, LSPname: to-lahore
  ActivePath:  (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Autobandwidth
  MinBW: 5Mbps MaxBW: 250Mbps
  AdjustTimer: 300 secs
  Max AvgBW util: 0bps, Bandwidth Adjustment in 102 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    Bandwidth: 5Mbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 4)
 10.252.0.177 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          10.252.0.177
Total 1 displayed, Up 1, Down 0

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

show mpls lsp statistics extensive

```
user@host> show mpls lsp statistics extensive
```

```
Ingress LSP: 1 sessions

192.168.0.4
  From: 192.168.0.5, State: Up, ActiveRoute: 0, LSPname: E-D
  Statistics: Packets 302, Bytes 28992
  Aggregate statistics: Packets 302, Bytes 28992
  ActivePath:  (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 30)
 10.0.0.18 S 10.0.0.22 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
 20=Node-ID):
          10.0.0.18 10.0.0.22
    6 Oct  3 11:18:28.281 Selected as active path
    5 Oct  3 11:18:28.281 Record Route:  10.0.0.18 10.0.0.22
    4 Oct  3 11:18:28.280 Up
    3 Oct  3 11:18:27.995 Originate Call
    2 Oct  3 11:18:27.995 CSPF: computation result accepted  10.0.0.18 10.0.0.22

    1 Oct  3 11:17:59.118 CSPF failed: no route toward 192.168.0.4[2 times]
  Created: Wed Oct  3 11:17:01 2012
Total 1 displayed, Up 1, Down 0
```

show msdp

Syntax

show msdp
<brief | detail>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<peer *peer-address*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display Multicast Source Discovery Protocol (MSDP) information.

Options

none—Display standard MSDP information for all routing instances.

brief | detail—(Optional) Display the specified level of output.

instance *instance-name*—(Optional) Display information for the specified instance only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

peer *peer-address*—(Optional) Display information about the specified peer only,

Required Privilege Level

view

Related Documentation

- show msdp source on page 1550
- show msdp source-active on page 1552
- show msdp statistics on page 1555

List of Sample Output

show msdp on page 1548
show msdp brief on page 1548
show msdp detail on page 1548

Output Fields

Table 62 on page 1547 describes the output fields for the **show msdp** command. Output fields are listed in the approximate order in which they appear.

Table 62: show msdp Output Fields

Field Name	Field Description	Level of Output
Peer address	IP address of the peer.	All levels
Local address	Local address of the peer.	All levels

Table 62: show msdp Output Fields (continued)

Field Name	Field Description	Level of Output
State	Status of the MSDP connection: **Listen**, **Established**, or **Inactive**.	All levels
Last up/down	Time at which the most recent peer-state change occurred.	All levels
Peer-Group	Peer group name.	All levels
SA Count	Number of source-active cache entries advertised by each peer that were accepted, compared to the number that were received, in the format ***number-accepted/number-received***.	All levels
Peer Connect Retries	Number of peer connection retries.	**detail**
State timer expires	Number of seconds before another message is sent to a peer.	**detail**
Peer Times out	Number of seconds to wait for a response from the peer before the peer is declared unavailable.	**detail**
SA accepted	Number of entries in the source-active cache accepted from the peer.	**detail**
SA received	Number of entries in the source-active cache received by the peer.	**detail**

Sample Output

show msdp

```
user@host> show msdp
Peer address     Local address   State       Last up/down Peer-Group SA Count
198.32.8.193     198.32.8.195    Established  5d 19:25:44 North23    120/150
198.32.8.194     198.32.8.195    Established  3d 19:27:27 North23    300/345
198.32.8.196     198.32.8.195    Established  5d 19:39:36 North23    10/13
198.32.8.197     198.32.8.195    Established  5d 19:32:27 North23    5/6
198.32.8.198     198.32.8.195    Established  3d 19:33:04 North23    2305/3000
```

show msdp brief

The output for the **show msdp brief** command is identical to that for the **show msdp** command. For sample output, see show msdp on page 1548.

show msdp detail

```
user@host> show msdp detail
Peer: 10.255.70.15
Local address: 10.255.70.19
State: Established
Peer Connect Retries: 0
State timer expires: 22
Peer Times out: 49
SA accepted: 0
SA received: 0
```

show msdp source

Syntax

show msdp source
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<*source-address*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display multicast sources learned from Multicast Source Discovery Protocol (MSDP).

Options

none—Display standard MSDP source information for all routing instances.

instance *instance-name*—(Optional) Display information for the specified instance only.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

source-address—(Optional) IP address and optional prefix length. Display information for the specified source address only.

Required Privilege Level

view

Related Documentation

- show msdp on page 1547
- show msdp source-active on page 1552
- show msdp statistics on page 1555

List of Sample Output

show msdp source on page 1551

Output Fields Table 63 on page 1551 describes the output fields for the **show msdp source** command. Output fields are listed in the approximate order in which they appear.

Table 63: show msdp source Output Fields

Field Name	Field Description
Source address	IP address of the source.
/Len	Length of the prefix for this IP address.
Type	Discovery method for this multicast source: • **Configured**—Source-active limit explicitly configured for this source. • **Dynamic**—Source-active limit established when this source was discovered.
Maximum	Source-active limit applied to this source.
Threshold	Source-active threshold applied to this source.
Exceeded	Number of source-active messages received from this source exceeding the established maximum.

Sample Output

show msdp source

```
user@host> show msdp source
Source address /Len   Type       Maximum   Threshold    Exceeded
0.0.0.0        /0     Configured       5        none           0
10.1.0.0       /16    Configured     500        none           0
10.1.1.1       /32    Configured   10000        none           0
10.1.1.2       /32    Dynamic       6936        none           0
10.1.5.5       /32    Dynamic        500        none         123
10.2.1.1       /32    Dynamic          2        none           0
```

show msdp source-active

Syntax

show msdp source-active
<brief | detail>
<group *group*>
<instance *instance-name*>
<local>
<logical-system (all | *logical-system-name*)>
<originator *originator*>
<peer *peer-address*>
<source *source-address*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display the Multicast Source Discovery Protocol (MSDP) source-active cache.

Options

none—Display standard MSDP source-active cache information for all routing instances.

brief | detail—(Optional) Display the specified level of output.

group *group*—(Optional) Display source-active cache information for the specified group.

instance *instance-name*—(Optional) Display information for the specified instance.

local—(Optional) Display all source-active caches originated by this router.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

originator *originator*—(Optional) Display information about the peer that originated the source-active cache entries.

peer *peer-address*—(Optional) Display the source-active cache of the specified peer.

source *source-address*—(Optional) Display the source-active cache of the specified source.

Required Privilege Level

view

Related Documentation

- show msdp on page 1547
- show msdp source on page 1550
- show msdp statistics on page 1555

List of Sample Output

show msdp source-active on page 1553
show msdp source-active brief on page 1554

show msdp source-active detail on page 1554
show msdp source-active source on page 1554

Output Fields Table 64 on page 1553 describes the output fields for the **show msdp source-active** command. Output fields are listed in the approximate order in which they appear.

Table 64: show msdp source-active Output Fields

Field Name	Field Description
Global active source limit exceeded	Number of times all peers have exceeded configured active source limits.
Global active source limit maximum	Configured number of active source messages accepted by the device.
Global active source limit threshold	Configured threshold for applying random early discard (RED) to drop some but not all MSDP active source messages.
Global active source limit log-warning	Threshold at which a warning message is logged (percentage of the number of active source messages accepted by the device).
Global active source limit log interval	Time (in seconds) between consecutive log messages.
Group address	Multicast address of the group.
Source address	IP address of the source.
Peer address	IP address of the peer.
Originator	Router ID configured on the source of the rendezvous point (RP) that originated the message, or the loopback address when the router ID is not configured.
Flags	Flags: **Accept**, **Reject**, or **Filtered**.

Sample Output

show msdp source-active

```
user@host> show msdp source-active
Group address   Source address  Peer address    Originator      Flags
230.0.0.0       192.168.195.46  local           10.255.14.30    Accept
230.0.0.1       192.168.195.46  local           10.255.14.30    Accept
230.0.0.2       192.168.195.46  local           10.255.14.30    Accept
230.0.0.3       192.168.195.46  local           10.255.14.30    Accept
230.0.0.4       192.168.195.46  local           10.255.14.30    Accept
```

show msdp source-active brief

The output for the **show msdp source-active brief** command is identical to that for the **show msdp source-active** command. For sample output, see show msdp source-active on page 1553.

show msdp source-active detail

The output for the **show msdp source-active detail** command is identical to that for the **show msdp source-active** command. For sample output, see show msdp source-active on page 1553.

show msdp source-active source

```
user@host> show msdp source-active source 192.168.215.246
Global active source limit exceeded: 0
Global active source limit maximum: 25000
Global active source limit threshold: 24000
Global active source limit log-warning: 100
Global active source limit log interval: 0

Group address   Source address  Peer address    Originator      Flags
226.2.2.1       192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.3       192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.4       192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.5       192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.7       192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.10      192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.11      192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.13      192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.14      192.168.215.246 10.255.182.140  10.255.182.140  Accept
226.2.2.15      192.168.215.246 10.255.182.140  10.255.182.140  Accept
```

show msdp statistics

Syntax

show msdp statistics
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<peer *peer-address*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display statistics about Multicast Source Discovery Protocol (MSDP) peers.

Options

none—Display statistics about all MSDP peers for all routing instances.

instance *instance-name*—(Optional) Display statistics about a specific MSDP instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

peer *peer-address*—(Optional) Display statistics about a particular MSDP peer.

Required Privilege Level

view

Related Documentation

- clear msdp statistics on page 1407

List of Sample Output

show msdp statistics on page 1557
show msdp statistics peer on page 1557

Output Fields

Table 65 on page 1555 describes the output fields for the **show msdp statistics** command. Output fields are listed in the approximate order in which they appear.

Table 65: show msdp statistics Output Fields

Field Name	Field Description
Global active source limit exceeded	Number of times all peers have exceeded configured active source limits.
Global active source limit maximum	Configured number of active source messages accepted by the device.
Global active source limit threshold	Configured threshold for applying random early discard (RED) to drop some but not all MSDP active source messages.
Global active source limit log-warning	Threshold at which a warning message is logged (percentage of the number of active source messages accepted by the device).

Table 65: show msdp statistics Output Fields (continued)

Field Name	Field Description
Global active source limit log interval	Time (in seconds) between consecutive log messages.
Peer	Address of peer.
Last State Change	How long ago the peer state changed.
Last message received from the peer	How long ago the last message was received from the peer.
RPF Failures	Number of reverse path forwarding (RPF) failures.
Remote Closes	Number of times the remote peer closed.
Peer Timeouts	Number of peer timeouts.
SA messages sent	Number of source-active messages sent.
SA messages received	Number of source-active messages received.
SA request messages sent	Number of source-active request messages sent.
SA request messages received	Number of source-active request messages received.
SA response messages sent	Number of source-active response messages sent.
SA response messages received	Number of source-active response messages received.
SA messages with zero Entry Count received	Entry Count is a field within SA message that defines how many source/group tuples are present in the SA message. The counter is incremented each time an SA with an Entry Count of zero is received.
Active source exceeded	Number of times this peer has exceeded configured source-active limits.
Active source Maximum	Configured number of active source messages accepted by this peer.
Active source threshold	Configured threshold on this peer for applying random early discard (RED) to drop some but not all MSDP active source messages.
Active source log-warning	Configured threshold on this peer at which a warning message is logged (percentage of the number of active source messages accepted by the device).
Active source log-interval	Time (in seconds) between consecutive log messages on this peer.

Table 65: show msdp statistics Output Fields (continued)

Field Name	Field Description
Keepalive messages sent	Number of keepalive messages sent.
Keepalive messages received	Number of keepalive messages received.
Unknown messages received	Number of unknown messages received.
Error messages received	Number of error messages received.

Sample Output

show msdp statistics

```
user@host> show msdp statistics
Global active source limit exceeded: 0
Global active source limit maximum: 10
Global active source limit threshold: 8
Global active source limit log-warning: 60
Global active source limit log interval: 60

Peer: 10.255.245.39
Last State Change: 11:54:49 (00:24:59)
Last message received from peer: 11:53:32 (00:26:16)
RPF Failures: 0
Remote Closes: 0
Peer Timeouts: 0
SA messages sent: 376
SA messages received: 459
SA messages with zero Entry Count received: 0
SA request messages sent: 0
SA request messages received: 0
SA response messages sent: 0
SA response messages received: 0
Active source exceeded: 0
Active source Maximum: 10
Active source threshold: 8
Active source log-warning: 60
Active source log-interval 120
Keepalive messages sent: 17
Keepalive messages received: 19
Unknown messages received: 0
Error messages received: 0
```

show msdp statistics peer

```
user@host> show msdp statistics peer 10.255.182.140
Peer: 10.255.182.140
    Last State Change: 8:19:23 (00:01:08)
    Last message received from peer: 8:20:05 (00:00:26)
    RPF Failures: 0
    Remote Closes: 0
    Peer Timeouts: 0
```

```
SA messages sent: 17
SA messages received: 16
SA request messages sent: 0
SA request messages received: 0
SA response messages sent: 0
SA response messages received: 0
Active source exceeded: 20
Active source Maximum: 10
Active source threshold: 8
Active source log-warning: 60
Active source log-interval: 120
Keepalive messages sent: 0
Keepalive messages received: 0
Unknown messages received: 0
Error messages received: 0
```

show multicast backup-pe-groups

Syntax

show multicast backup-pe-groups
<address *pe-address*>
<group *group-name*>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced in Junos OS Release 9.0.

Description

Display backup PE router group information when ingress PE redundancy is configured. Ingress PE redundancy provides a backup resource when point-to-multipoint LSPs are configured for multicast distribution.

Options

none—Display standard information about all backup PE groups.

address *pe-address*—(Optional) Display the groups that a PE address is associated with.

group *group*—(Optional) Display the backup PE group information for a particular group.

instance *instance-name*—(Optional) Display backup PE group information for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

List of Sample Output

show multicast backup-pe-groups on page 1560

Output Fields

Table 66 on page 1559 describes the output fields for the **show multicast backup-pe-groups** command. Output fields are listed in the approximate order in which they appear.

Table 66: show multicast backup-pe-groups Output Fields

Field Name	Field Description
Backup PE Group	Group name.
Designated PE	Primary PE router. Address of the PE router that is currently forwarding traffic on the static route.
Transitions	Number of times that the designated PE router has transitioned from the most eligible PE router to a backup PE router and back again to the most eligible PE router.
Last Transition	Time of the most recent transition.
Local Address	Address of the local PE router.
Backup PE List	List of PE routers that are configured to be backups for the group.

Sample Output

show multicast backup-pe-groups

```
user@host> show multicast backup-pe-groups
Instance: master

Backup PE group: b1
    Designated PE:   10.255.165.7
    Transitions:     1
    Last Transition: 03:15:01
    Local Address:   10.255.165.7
    Backup PE List:
                     10.255.165.8

Backup PE group: b2
    Designated PE:   10.255.165.7
    Transitions:     2
    Last Transition: 02:58:20
    Local Address:   10.255.165.7
    Backup PE List:
                     10.255.165.9
                     10.255.165.8
```

show multicast flow-map

List of Syntax
Syntax on page 1561
Syntax (EX Series Switch and the QFX Series) on page 1561

Syntax
show multicast flow-map
<brief | detail>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show multicast flow-map
<brief | detail>

Release Information
Command introduced in Junos OS Release 8.2.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display configuration information about IP multicast flow maps.

Options
none—Display configuration information about IP multicast flow maps on all systems.

brief | detail—(Optional) Display the specified level of output.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show multicast flow-map on page 1562
show multicast flow-map detail on page 1562

Output Fields
Table 67 on page 1561 describes the output fields for the **show multicast flow-map** command. Output fields are listed in the approximate order in which they appear.

Table 67: show multicast flow-map Output Fields

Field Name	Field Description	Levels of Output
Name	Name of the flow map.	All levels
Policy	Name of the policy associated with the flow map.	All levels
Cache-timeout	Cache timeout value assigned to the flow map.	All levels
Bandwidth	Bandwidth setting associated with the flow map.	All levels
Adaptive	Whether or not adaptive mode is enabled for the flow map.	none

Table 67: show multicast flow-map Output Fields (continued)

Field Name	Field Description	Levels of Output
Flow-map	Name of the flow map.	**detail**
Adaptive Bandwidth	Whether or not adaptive mode is enabled for the flow map.	**detail**
Redundant Sources	Redundant sources defined for the same destination group.	**detail**

Sample Output

show multicast flow-map

```
user@host> show multicast flow-map
Instance: master
Name            Policy                  Cache timeout        Bandwidth Adaptive
map2            policy2                         never          2000000 no
map1            policy1                   60 seconds           2000000 no
```

Sample Output

show multicast flow-map detail

```
user@host> show multicast flow-map detail
Instance: master
Flow-map: map1
    Policy:             policy1
    Cache Timeout:      600 seconds
    Bandwidth:          2000000
    Adaptive Bandwidth: yes
    Redundant Sources:  10.11.11.11
    Redundant Sources:  10.11.11.12
    Redundant Sources:  10.11.11.13
```

show multicast forwarding-cache statistics

Syntax

show multicast forwarding-cache statistics
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced in Junos OS Release 12.2.
Starting in Junos OS Release 16.1, output includes general and rendezvous-point tree (RPT) suppression states.

Description

Display IP multicast forwarding cache statistics.

Options

none—Display multicast forwarding cache statistics for all supported address families for all routing instances.

inet | inet6—(Optional) Display multicast forwarding cache statistics for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display multicast forwarding cache statistics for a specific routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- clear multicast forwarding-cache on page 1410
- threshold on page 1321

List of Sample Output

show multicast forwarding cache statistics instance on page 1564
show multicast forwarding cache statistics instance (Forwarding-cache suppression is disabled) on page 1564

Output Fields

Table 68 on page 1563 describes the output fields for the **show multicast forwarding-cache statistics** command. Output fields are listed in the approximate order in which they appear.

Table 68: show multicast forwarding-cache statistics Output Fields

Field Name	Field Description
Instance	Name of the routing instance for which multicast forwarding cache statistics are displayed.
Family	Protocol family for which multicast forwarding cache statistics are displayed: **ALL**, **INET**, or **INET6**.

Table 68: show multicast forwarding-cache statistics Output Fields (continued)

Field Name	Field Description
General (or MVPN RPT) Suppression Active	Indicates whether suppression is configured.
General (or MVPN RPT) Entries Used	Number of currently used multicast forwarding cache entries.
General (or MVPN RPT) Suppress Threshold	Maximum number of multicast forwarding cache entries that can be added to the cache. When the number of entries reaches the configured threshold, the device suspends adding new multicast forwarding cache entries.
General (or MVPN RPT) Reuse Value	Number of multicast forwarding cache entries that must be reached before the device creates new multicast forwarding cache entries. When the total number of multicast forwarding cache entries is below the reuse value, the device resumes adding new multicast forwarding cache entries.

Sample Output

show multicast forwarding cache statistics instance

```
user@host> show multicast forwarding-cache statistic instance mvpn1 inet6
  Instance: mvpn1 Family: INET6
General Suppression Active                  Yes
General Entries Used                          0
General Suppress Threshold                  200
General Reuse Value                         200
MVPN RPT Suppression Active                 Yes
MVPN RPT Entries Used                         0
MVPN RPT Suppress Threshold                 200
MVPN RPT Reuse Value                        200
```

show multicast forwarding cache statistics instance (Forwarding-cache suppression is disabled)

```
user@host> show multicast forwarding-cache statistic instance mvpn1
Instance: mvpn1 Family: ALL
Forwarding-cache suppression disabled Not enabled by configuration
```

show multicast interface

List of Syntax
Syntax on page 1565
Syntax (EX Series Switch and the QFX Series) on page 1565

Syntax
show multicast interface
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show multicast interface

Release Information
Command introduced in Junos OS Release 8.3.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display bandwidth information about IP multicast interfaces.

Options
none—Display all interfaces that have multicast configured.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show multicast interface on page 1566

Output Fields
Table 69 on page 1565 describes the output fields for the **show multicast interface** command. Output fields are listed in the approximate order in which they appear.

Table 69: show multicast interface Output Fields

Field Name	Field Description
Interface	Name of the multicast interface.
Maximum bandwidth (bps)	Maximum bandwidth setting, in bits per second, for this interface.
Remaining bandwidth (bps)	Amount of bandwidth, in bits per second, remaining on the interface.

Table 69: show multicast interface Output Fields (continued)

Field Name	Field Description
Mapped bandwidth deduction (bps)	Amount of bandwidth, in bits per second, used by any flows that are mapped to the interface. NOTE: Adding the mapped bandwidth deduction value to the local bandwidth deduction value results in the total deduction value for the interface. This field does not appear in the output when the no QoS adjustment feature is disabled.
Local bandwidth deduction (bps)	Amount of bandwidth, in bits per second, used by any mapped flows that are traversing the interface. NOTE: Adding the mapped bandwidth deduction value to the local bandwidth deduction value results in the total deduction value for the interface. This field does not appear in the output when the no QoS adjustment feature is disabled.
Reverse OIF mapping	State of the reverse OIF mapping feature (**on** or **off**). NOTE: This field does not appear in the output when the no QoS adjustment feature is disabled.
Reverse OIF mapping no QoS adjustment	State of the no QoS adjustment feature (**on** or **off**) for interfaces that are using reverse OIF mapping. NOTE: This field does not appear in the output when the no QoS adjustment feature is disabled.
Leave timer	Amount of time a mapped interface remains active after the last mapping ends. NOTE: This field does not appear in the output when the no QoS adjustment feature is disabled.
No QoS adjustment	State (**on**) of the no QoS adjustment feature when this feature is enabled. NOTE: This field does not appear in the output when the no QoS adjustment feature is disabled.

Sample Output

show multicast interface

```
user@host> show multicast interface
Interface               Maximum bandwidth (bps) Remaining bandwidth (bps)
fe-0/0/3                              10000000                         0
fe-0/0/3.210                          10000000                  -2000000
fe-0/0/3.220                         100000000                 100000000
fe-0/0/3.230                          20000000                  18000000
fe-0/0/2.200                         100000000                 100000000
```

show multicast mrinfo

Syntax show multicast mrinfo
<*host*>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Display configuration information about IP multicast networks, including neighboring multicast router addresses.

Options **none**—Display configuration information about all multicast networks.

host—(Optional) Display configuration information about a particular host. Replace *host* with a hostname or IP address.

Required Privilege Level view

List of Sample Output show multicast mrinfo on page 1569

Output Fields Table 70 on page 1568 describes the output fields for the **show multicast mrinfo** command. Output fields are listed in the approximate order in which they appear.

Table 70: show multicast mrinfo Output Fields

Field Name	Field Description
source-address	Query address, hostname (DNS name or IP address of the source address), and multicast protocol version or the software version of another vendor.
ip-address-1—>ip-address-2	Queried router interface address and directly attached neighbor interface address, respectively.
(*name* or *ip-address*)	Name or IP address of neighbor.

[*metric/threshold/type/flags*]	Neighbor's multicast profile:
	• *metric*—Always has a value of **1**, because **mrinfo** queries the directly connected interfaces of a device.
	• *threshold*—Multicast threshold time-to-live (TTL). The range of values is **0** through **255**.
	• *type*—Multicast connection type: **pim** or **tunnel**.
	• *flags*—Flags for this route:
	  • **querier**—Queried router is the designated router for the neighboring session.
	  • **leaf**—Link is a leaf in the multicast network.
	  • **down**—Link status indicator.

Sample Output

show multicast mrinfo

```
user@host> show multicast mrinfo 10.35.4.1
10.35.4.1 (10.35.4.1) [version 12.0]:
  192.168.195.166 -> 0.0.0.0 (local) [1/0/pim/querier/leaf]
  10.38.20.1 -> 0.0.0.0 (local) [1/0/pim/querier/leaf]
  10.47.1.1 -> 10.47.1.2 (10.47.1.2) [1/5/pim]
  0.0.0.0 -> 0.0.0.0 (local) [1/0/pim/down]
```

show multicast next-hops

List of Syntax
Syntax on page 1570
Syntax (EX Series Switch and the QFX Series) on page 1570

Syntax
show multicast next-hops
<brief | detail | terse>
<*identifier-number*>
<inet | inet6>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show multicast next-hops
<brief | detail>
<*identifier-number*>
<inet | inet6>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 option introduced in Junos OS Release 10.0 for EX Series switches.
detail option display of next-hop ID number introduced in Junos OS Release 11.1 for M Series and T Series routers and EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional PIM added in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
terse option introduced in Junos OS Release 16.1 for the MX Series.

Description
Display the entries in the IP multicast next-hop table.

Options
none—Display standard information about all entries in the multicast next-hop table for all supported address families.

brief | detail | terse—(Optional) Display the specified level of output. Use **terse** to display the total number of outgoing interfaces (as opposed to listing them) When you include the **detail** option on M Series and T Series routers and EX Series switches, the downstream interface name includes the next-hop ID number in parentheses, in the form **fe-0/1/2.0-(1048574),** where **1048574** is the next-hop ID number.

Starting in Junos OS release 16.1, the **show multicast next-hops** statement shows the hierarchical next hops contained in the top-level next hop.

identifier-number—(Optional) Show a particular next hop by ID number. The range of values is **1** through **65,535**.

inet | inet6—(Optional) Display entries for IPv4 or IPv6 family addresses, respectively.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level view

List of Sample Output

Output Fields Table 71 on page 1571 describes the output fields for the **show multicast next-hops** command. Output fields are listed in the approximate order in which they appear.

Table 71: show multicast next-hops Output Fields

Field Name	Field Description
Family	Protocol family (such as **INET**).
ID	Next-hop identifier of the prefix. The identifier is returned by the routing device's Packet Forwarding Engine.
Refcount	Number of cache entries that are using this next hop.
KRefcount	Kernel reference count for the next hop.
Downstream interface	Interface names associated with each multicast next-hop ID.
Incoming interface list	List of interfaces that accept incoming traffic. Only shown for routes that do not use strict RPF-based forwarding, for example for bidirectional PIM.

Sample Output

show multicast next-hops

```
user@host> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
262142          4          2 so-1/0/0.0
262143          2          1 mt-1/1/0.49152
262148          2          1 mt-1/1/1/0.32769
```

show multicast next-hops (Ingress Router, Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show multicast next-hops
Family: INET
ID          Refcount KRefcount Downstream interface Addr
1048580            2         1 1048576
(0x600dc04)        1         0 1048584
(0x600ea04)        1         0 (0x600e924)
```

```
1048583            2         1 1048579
(0x600e144)        1         0 1048587
(0x600e844)        1         0 (0x600e764)
1048582            2         1 1048578
(0x600df84)        1         0 1048586
(0x600e684)        1         0 (0x600e5a4)
1048581            2         1 1048577
(0x600ddc4)        1         0 1048585
(0x600ebc4)        1         0 (0x600eae4)
```

show multicast next-hops (Egress Router, Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show multicast next-hops
Family: INET
ID          Refcount KRefcount Downstream interface Addr
(0x600e844)        8         0 1048575
1048575           16         0 distributed-gmp
```

show multicast next-hops (Bidirectional PIM)

```
user@host> show multicast next-hops
Family: INET
ID          Refcount KRefcount Downstream interface
2097151            8         4 ge-0/0/1.0

Family: INET6
ID          Refcount KRefcount Downstream interface
2097157            2         1 ge-0/0/1.0

Family: Incoming interface list
ID          Refcount KRefcount Downstream interface
513                5         2 lo0.0
                               ge-0/0/1.0
514                5         2 lo0.0
                               ge-0/0/1.0
                               xe-4/1/0.0
515                3         1 lo0.0
                               ge-0/0/1.0
                               xe-4/1/0.0
544                1         0 lo0.0
                               xe-4/1/0.0
```

show multicast next-hops brief

The output for the **show multicast next-hops brief** command is identical to that for the **show multicast next-hops** command. For sample output, see show multicast next-hops on page 1571.

show multicast next-hops detail

```
user@host> show multicast next-hops detail
Family: INET
ID          Refcount KRefcount Downstream interface Addr
1048584            2         1 1048581
                               1048580
                               Flags 0x208 type 0x18 members 0/0/2/0/0
                               Address 0xb1841c4
1048591            3         2 787
                               747
```

```
                                  Flags 0x206 type 0x18 members 0/0/2/0/0
                                  Address 0xb1847f4
1048580            4         1 ge-1/1/9.0-(1048579)
                                  Flags 0x200 type 0x18 members 0/0/0/1/0
                                  Address 0xb184134
1048581            2         0 736
                                  765
                                  Flags 0x3 type 0x18 members 0/0/2/0/0
                                  Address 0xb183dd4
1048585           18         0 787
                                  747
                                  Flags 0x203 type 0x18 members 0/0/2/0/0
                                  Address 0xb184404

Family: INET6
ID          Refcount KRefcount Downstream interface Addr
1048586            4         2 1048585
                                  1048583
                                  Flags 0x20c type 0x19 members 0/0/2/0/0
                                  Address 0xb1842e4
1048583           14         4 ge-1/1/9.0-(1048582)
                                  Flags 0x200 type 0x19 members 0/0/0/1/0
                                  Address 0xb183ef4
1048592            4         2 1048583
                                  1048591
                                  Flags 0x20c type 0x19 members 0/0/2/0/0
                                  Address 0xb184644
```

show multicast pim-to-igmp-proxy

List of Syntax
Syntax on page 1574
Syntax (EX Series Switch and the QFX Series) on page 1574

Syntax
show multicast pim-to-igmp-proxy
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show multicast pim-to-igmp-proxy
<instance *instance-name*>

Release Information
Command introduced in Junos OS Release 9.6.
Command introduced in Junos OS Release 9.6 for EX Series switches.
instance option introduced in Junos OS Release 10.3.
instance option introduced in Junos OS Release 10.3 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display configuration information about PIM-to-IGMP message translation, also known as PIM-to-IGMP proxy.

Options
none—Display configuration information about PIM-to-IGMP message translation for all routing instances.

instance *instance-name*—(Optional) Display configuration information about PIM-to-IGMP message translation for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

Related Documentation
- Configuring PIM-to-IGMP and PIM-to-MLD Message Translation on page 384

List of Sample Output
show multicast pim-to-igmp-proxy on page 1575
show multicast pim-to-igmp-proxy instance on page 1575

Output Fields
Table 72 on page 1575 describes the output fields for the **show multicast pim-to-igmp-proxy** command. Output fields are listed in the order in which they appear.

Table 72: show multicast pim-to-igmp-proxy Output Fields

Field Name	Field Description
Instance	Routing instance. Default instance is **master** (inet.0 routing table).
Proxy state	State of PIM-to-IGMP message translation, also known as PIM-to-IGMP proxy, on the configured upstream interfaces: **enabled** or **disabled**.
interface-name	Name of upstream interface (no more than two allowed) on which PIM-to-IGMP message translation is configured.

Sample Output

show multicast pim-to-igmp-proxy

```
user@host> show multicast pim-to-igmp-proxy
Instance: master Proxy state: enabled
ge-0/1/0.1
ge-0/1/0.2
```

show multicast pim-to-igmp-proxy instance

```
user@host> show multicast pim-to-igmp-proxy instance VPN-A
Instance: VPN-A Proxy state: enabled
ge-0/1/0.1
```

show multicast pim-to-mld-proxy

List of Syntax

Syntax on page 1576
Syntax (EX Series Switch and the QFX Series) on page 1576

Syntax

show multicast pim-to-mld-proxy
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

show multicast pim-to-mld-proxy
<instance *instance-name*>

Release Information

Command introduced in Junos OS Release 9.6.
Command introduced in Junos OS Release 9.6 for EX Series switches.
instance option introduced in Junos OS Release 10.3.
instance option introduced in Junos OS Release 10.3 for EX Series switches.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Command introduced in Junos OS Release 11.3 for the QFX Series.

Description

Display configuration information about PIM-to-MLD message translation, also known as PIM-to-MLD proxy.

Options

none—Display configuration information about PIM-to-MLD message translation for all routing instances.

instance *instance-name*—(Optional) Display configuration information about PIM-to-MLD message translation for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

List of Sample Output

show multicast pim-to-mld-proxy on page 1577
show multicast pim-to-mld-proxy instance on page 1577

Output Fields

Table 73 on page 1576 describes the output fields for the **show multicast pim-to-mld-proxy** command. Output fields are listed in the order in which they appear.

Table 73: show multicast pim-to-mld-proxy Output Fields

Field Name	Field Description
Proxy state	State of PIM-to-MLD message translation, also known as PIM-to-MLD proxy, on the configured upstream interfaces: **enabled** or **disabled**.

Table 73: show multicast pim-to-mld-proxy Output Fields (continued)

Field Name	Field Description
interface-name	Name of upstream interface (no more than two allowed) on which PIM-to-MLD message translation is configured.

Sample Output

show multicast pim-to-mld-proxy

```
user@host> show multicast pim-to-mld-proxy
Instance: master Proxy state: enabled
ge-0/5/0.1
ge-0/5/0.2
```

show multicast pim-to-mld-proxy instance

```
user@host> show multicast pim-to-mld-proxy instance VPN-A
Instance: VPN-A Proxy state: enabled
ge-0/5/0.1
```

show multicast route

List of Syntax

Syntax on page 1578
Syntax (EX Series Switch and the QFX Series) on page 1578

Syntax

show multicast route
<brief | detail | extensive | summary>
<active | all | inactive>
<group *group*>
<inet | inet6>
<instance *instance name*>
<logical-system (all | *logical-system-name*)>
<oif-count>
<*regular-expression*>
<source-prefix *source-prefix*>

Syntax (EX Series Switch and the QFX Series)

show multicast route
<brief | detail | extensive | summary>
<active | all | inactive>
<group *group*>
<inet | inet6>
<instance *instance name*>
<*regular-expression*>
<source-prefix *source-prefix*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional PIM added in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
oif-count option introduced in Junos OS Release 16.1 for the MX Series.
xxxSupport for PIM NSR support for VXLAN added in Junos OS Release 16.2.

Description

Display the entries in the IP multicast forwarding table. You can display similar information with the **show route table inet.1** command.

NOTE: On all SRX Series devices, when a multicast route is not available, pending sessions are not torn down, and subsequent packets are queued. If no multicast route resolve comes back, then the traffic flow has to wait for the pending session to timed out. Then packets can trigger new pending session create and route resolve.

Options

none—Display standard information about all entries in the multicast forwarding table for all routing instances.

brief | detail | extensive | summary—(Optional) Display the specified level of output.

active | all | inactive—(Optional) Display all active entries, all entries, or all inactive entries, respectively, in the multicast forwarding table.

group *group*—(Optional) Display the cache entries for a particular group.

inet | inet6—(Optional) Display multicast forwarding table entries for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display entries in the multicast forwarding table for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

oif-count —(Optional) Display a count of outgoing interfaces rather than listing them.

regular-expression—(Optional) Display information about the multicast forwarding table entries that match a UNIX OS-style regular expression.

source-prefix *source-prefix*—(Optional) Display the cache entries for a particular source prefix.

Required Privilege Level

view

Related Documentation

- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837

List of Sample Output

show multicast route on page 1581
show multicast route (Bidirectional PIM) on page 1582
show multicast route brief on page 1582
show multicast route summary on page 1582
show multicast route detail on page 1582
show multicast route extensive (Bidirectional PIM) on page 1583
show multicast route extensive (Ingress Router, Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 1584
show multicast route instance <instance-name> extensive on page 1585
show multicast route extensive instance <instance-name> on page 1586
show multicast route extensive (PIM NSR support for VXLAN on master Routing Engine) on page 1586
show multicast route extensive (PIM NSR support for VXLAN on backup Routing Engine) on page 1587
show multicast route extensive (PIM NSR support for VXLAN on backup Routing Engine) on page 1588

Output Fields

Table 74 on page 1580 describes the output fields for the **show multicast route** command. Output fields are listed in the approximate order in which they appear.

Table 74: show multicast route Output Fields

Field Name	Field Description	Level of Output
family	IPv4 address family (**INET**) or IPv6 address family (**INET6**).	All levels
Group	Group address. For any-source multicast routes, for example for bidirectional PIM, the group address includes the prefix length.	All levels
Source	Prefix and length of the source as it is in the multicast forwarding table.	All levels
Incoming interface list	List of interfaces that accept incoming traffic. Only shown for routes that do not use strict RPF-based forwarding, for example for bidirectional PIM.	All levels
Upstream interface	Name of the interface on which the packet with this source prefix is expected to arrive.	All levels
Upstream rpf interface list	When multicast-only fast reroute (MoFRR) is enabled, a PIM router propagates join messages on two upstream RPF interfaces to receive multicast traffic on both links for the same join request.	All levels
Downstream interface list	List of interface names to which the packet with this source prefix is forwarded. **distributed-gmp**— Added in Junos OS Release 17.4R1 to indicate that line cards with distributed IGMP interfaces are receiving multicast traffic for a given (s,g).	All levels
Number of outgoing interfaces	Total number of outgoing interfaces for each (S,G) entry.	**extensive**
Session description	Name of the multicast session.	**detail extensive**
Statistics	Rate at which packets are being forwarded for this source and group entry (in Kbps and pps), and number of packets that have been forwarded to this prefix. If one or more of the kilobits per second packet forwarding statistic queries fails or times out, the statistics field displays **Forwarding statistics are not available**. NOTE: On QFX Series switches and OCX Series switches, this field does not report valid statistics.	**detail extensive**
Next-hop ID	Next-hop identifier of the prefix. The identifier is returned by the routing device's Packet Forwarding Engine and is also displayed in the output of the **show multicast nexthops** command.	**detail extensive**
Incoming interface list ID	For bidirectional PIM, incoming interface list identifier. Identifiers for interfaces that accept incoming traffic. Only shown for routes that do not use strict RPF-based forwarding, for example for bidirectional PIM.	**detail extensive**

Table 74: show multicast route Output Fields (continued)

Field Name	Field Description	Level of Output
Upstream protocol	The protocol that maintains the active multicast forwarding route for this group or source. When the **show multicast route extensive** command is used with the **display-origin-protocol** option, the field name is only **Protocol** and not **Upstream Protocol**. However, this field also displays the protocol that installed the active route.	**detail extensive**
Route type	Type of multicast route. Values can be (S,G) or (*,G).	**summary**
Route state	Whether the group is **Active** or **Inactive**.	**summary extensive**
Route count	Number of multicast routes.	**summary**
Forwarding state	Whether the prefix is pruned or forwarding.	**extensive**
Cache lifetime/timeout	Number of seconds until the prefix is removed from the multicast forwarding table. A value of **never** indicates a permanent forwarding entry. A value of **forever** indicates routes that do not have keepalive times.	**extensive**
Wrong incoming interface notifications	Number of times that the upstream interface was not available.	**extensive**
Uptime	Time since the creation of a multicast route.	**extensive**

Sample Output

Starting in Junos OS Release16.1, **show multicast route** displays the top-level hierarchical next hop.

show multicast route

```
user@host> show multicast route
Family: INET

Group: 233.252.0.0
    Source: 10.255.14.144/32
    Upstream interface: local
    Downstream interface list:
        so-1/0/0.0

Group: 233.252.0.1
    Source: 10.255.14.144/32
    Upstream interface: local
    Downstream interface list:
        so-1/0/0.0

Group: 233.252.0.1
    Source: 10.255.70.15/32
    Upstream interface: so-1/0/0.0
    Downstream interface list:
```

```
        mt-1/1/0.1081344

Family: INET6
```

show multicast route (Bidirectional PIM)

```
user@host> show multicast route
Family: INET

Group: 233.252.0.1/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0
    Downstream interface list:
        ge-0/0/1.0

Group: 233.252.0.3/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0 xe-4/1/0.0
    Downstream interface list:
        ge-0/0/1.0

Group: 233.252.0.11/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0
    Downstream interface list:
        ge-0/0/1.0

Group: 233.252.0.13/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0 xe-4/1/0.0
    Downstream interface list:
        ge-0/0/1.0
Family: INET6
```

show multicast route brief

The output for the **show multicast route brief** command is identical to that for the **show multicast route** command. For sample output, see show multicast route on page 1581 or show multicast route (Bidirectional PIM) on page 1582.

show multicast route summary

```
user@host>show multicast route summary
Instance: master Family: INET

Route type   Route state    Route count
(S,G)             Active             2
(S,G)             Inactive           3

Instance: master Family: INET6
```

show multicast route detail

```
user@host> show multicast route detail
```

```
Family: INET

Group: 233.252.0.0
    Source: 10.255.14.144/32
    Upstream interface: local
    Downstream interface list:
        so-1/0/0.0
    Session description: Unknown
    Statistics: 8 kBps, 100 pps, 45272 packets
    Next-hop ID: 262142
    Upstream protocol: PIM

Group: 233.252.0.1
    Source: 10.255.14.144/32
    Upstream interface: local
    Downstream interface list:
        so-1/0/0.0
    Session description: Administratively Scoped
    Statistics: 0 kBps, 0 pps, 13404 packets
    Next-hop ID: 262142
    Upstream protocol: PIM

Group: 233.252.0.1
    Source: 10.255.70.15/32
    Upstream interface: so-1/0/0.0
    Downstream interface list:
        mt-1/1/0.1081344
    Session description: Administratively Scoped
    Statistics: 46 kBps, 1000 pps, 921077 packets

    Next-hop ID: 262143
    Upstream protocol: PIM

Family: INET6
```

show multicast route extensive (Bidirectional PIM)

```
user@host> show multicast route extensive
Family: INET

Group: 233.252.0.1/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0
    Downstream interface list:
        ge-0/0/1.0
    Number of outgoing interfaces: 1
    Session description: NOB Cross media facilities
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097153
    Incoming interface list ID: 585
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Group: 233.252.0.3/24
    Source: *
    Incoming interface list:
        lo0.0 ge-0/0/1.0 xe-4/1/0.0
```

```
    Downstream interface list:
        ge-0/0/1.0
    Number of outgoing interfaces: 1
    Session description: NOB Cross media facilities
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 2097153
    Incoming interface list ID: 589
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Family: INET6
```

show multicast route extensive (Ingress Router, Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show multicast route extensive
 Family: INET

Group: 226.0.0.1
    Source: 200.1.0.2/32
    Upstream interface: xe-3/0/0.0
    Downstream interface list:
        ge-0/1/9.0
    Number of outgoing interfaces: 1
    Session description: Unknown
    Statistics: 78 kBps, 1000 pps, 34789 packets
    Next-hop ID: 1048582
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:00:35

Group: 226.0.0.2
    Source: 200.1.0.2/32
    Upstream interface: xe-3/0/0.0
    Downstream interface list:
        ge-0/1/9.0
    Number of outgoing interfaces: 1
    Session description: Unknown
    Statistics: 78 kBps, 1000 pps, 34788 packets
    Next-hop ID: 1048583
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:00:35

Group: 226.0.0.3
    Source: 200.1.0.2/32
    Upstream interface: xe-3/0/0.0
    Downstream interface list:
        ge-0/1/9.0
    Number of outgoing interfaces: 1
    Session description: Unknown
    Statistics: 78 kBps, 1000 pps, 34786 packets
    Next-hop ID: 1048580
```

```
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:00:35

Group: 226.0.0.4
    Source: 200.1.0.2/32
    Upstream interface: xe-3/0/0.0
    Downstream interface list:
        ge-0/1/9.0
    Number of outgoing interfaces: 1
    Session description: Unknown
    Statistics: 78 kBps, 1000 pps, 34787 packets
    Next-hop ID: 1048581
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:00:35

Instance: master Family: INET6
```

show multicast route instance <instance-name> extensive

```
user@host> show multicast route instance mvpn extensive
Family: INET
roup: 233.252.0.10
    Source: 10.0.0.2/32
    Upstream interface: xe-0/0/0.102
    Downstream interface list:
        xe-10/3/0.0 xe-0/3/0.0 xe-0/0/0.106 xe-0/0/0.105
        xe-0/0/0.103 xe-0/0/0.104 xe-0/0/0.107 xe-0/0/0.108
    Session description: Administratively Scoped
    Statistics: 256 kBps, 3998 pps, 670150 packets
    Next-hop ID: 1048579
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 58
    Uptime: 00:00:04

Instance: master Family: INET

Group: 225.0.0.1
    Source: 101.0.0.2/32
    Upstream interface: ge-2/2/0.101
    Downstream interface list:
        distributed-gmp
    Number of outgoing interfaces: 1
    Session description: Unknown
    Statistics: 105 kBps, 2500 pps, 4153361 packets
    Next-hop ID: 1048575
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
```

```
    Uptime: 00:31:46

Group: 225.0.0.1
    Source: 101.0.0.3/32
    Upstream interface: ge-2/2/0.101
    Downstream interface list:
        distributed-gmp
    Number of outgoing interfaces: 1
    Session description: Unknown
    Statistics: 105 kBps, 2500 pps, 4153289 packets
    Next-hop ID: 1048575
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0
    Uptime: 00:31:46
```

show multicast route extensive instance <instance-name>

```
user@host> show multicast route extensive instance VPN-A
```

The double asterisks (**) indicate new output, specific to **min-rate** and **revert-delay** settings made under the **[edit routing-instances** *routing-instance-name* protocols mvpn hot-root-standby] hierarchy.

```
Instance: VPNA Family: INET

      Group: 227.1.1.1
          Source: 18.1.1.2/32
          Upstream rpf interface list:
              vt-2/0/10.1000 (P)
                Session Id: 0x156 Session Status: Up
                Min-rate: 10000 kbps Weight: 1
                  Sender Id: Label 299808
              vt-2/0/10.1000 (B)
                Session Id: 0x155 Session Status: Up
                Min-rate: 10000 kbps Weight: 65533
                Sender Id: Label 299824
          Downstream interface list:
              lt-2/0/10.0
          Number of outgoing interfaces: 1
          Session description: Unknown
          Statistics: 8258 kBps, 100707 pps, 513032034 packets
          RPF Next-hop ID: 803
          Next-hop ID: 1048580
          Upstream protocol: MVPN
          Route state: Active
          Forwarding state: Forwarding
          Cache lifetime/timeout: forever
          Wrong incoming interface notifications: 0
          Uptime: 01:24:55
```

show multicast route extensive (PIM NSR support for VXLAN on master Routing Engine)

```
user@host> show multicast route extensive
Instance: master Family: INET

Group: 233.252.0.1
    Source: 10.3.3.3/32
```

```
    Upstream interface: ge-3/1/2.0
    Downstream interface list:
        -(593)
    Number of outgoing interfaces: 1
    Session description: Organizational Local Scope
    Statistics: 0 kBps, 0 pps, 27 packets
    Next-hop ID: 1048576
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding (Forwarding state is set as 'Forwarding' in
master RE.)
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:06:38

Group: 233.252.0.1
    Source: 10.2.1.4/32
    Upstream interface: local
    Downstream interface list:
        ge-3/1/2.0
    Number of outgoing interfaces: 1
    Session description: Organizational Local Scope
    Statistics: 0 kBps, 0 pps, 86 packets
    Next-hop ID: 1048575
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding (Forwarding state is set as 'Forwarding' in
master RE.)
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:07:45

Instance: master Family: INET6
```

show multicast route extensive (PIM NSR support for VXLAN on backup Routing Engine)

```
user@host> show multicast route extensive
Instance: master Family: INET

Group: 233.252.0.1
    Source: 10.3.3.3/32
    Upstream interface: ge-3/1/2.0
    Number of outgoing interfaces: 0
    Session description: Organizational Local Scope
    Forwarding statistics are not available
    Next-hop ID: 0
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Pruned (Forwarding state is set as 'Pruned' in backup RE.)

    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:06:46

Group: 233.252.0.1
    Source: 10.2.1.4/32
    Upstream interface: local
    Number of outgoing interfaces: 0
    Session description: Organizational Local Scope
    Forwarding statistics are not available
    Next-hop ID: 0
```

```
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Pruned  (Forwarding state is set as 'Pruned' in backup RE.)

    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:07:54

Instance: master Family: INET6
```

show multicast route extensive (PIM NSR support for VXLAN on backup Routing Engine)

```
user@host> show multicast route extensive
Instance: master Family: INET

Group: 233.252.0.1
    Source: 10.3.3.3/32
    Upstream interface: ge-3/1/2.0
    Downstream interface list:
        -(593)
    Number of outgoing interfaces: 1
    Session description: Organisational Local Scope
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 1048576
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding (Forwarding state is set as 'Forwarding' in
backup RE.)
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:06:38

Group: 233.252.0.1
    Source: 10.2.1.4/32
    Upstream interface: local
    Downstream interface list:
        ge-3/1/2.0
    Number of outgoing interfaces: 1
    Session description: Organisational Local Scope
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 1048575
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding (Forwarding state is set as 'Forwarding' in
backup RE.)
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0
    Uptime: 00:07:45

Instance: master Family: INET6
```

show multicast rpf

List of Syntax

Syntax on page 1589
Syntax (EX Series Switch and the QFX Series) on page 1589

Syntax

show multicast rpf
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<*prefix*>
<summary>

Syntax (EX Series Switch and the QFX Series)

show multicast rpf
<inet | inet6>
<instance *instance-name*>
<*prefix*>
<summary>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display information about multicast reverse-path-forwarding (RPF) calculations.

Options

none—Display RPF calculation information for all supported address families.

inet | inet6—(Optional) Display the RPF calculation information for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display information about multicast RPF calculations for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

prefix—(Optional) Display the RPF calculation information for the specified prefix.

summary—(Optional) Display a summary of all multicast RPF information.

Required Privilege Level

view

List of Sample Output

show multicast rpf on page 1590
show multicast rpf inet6 on page 1591
show multicast rpf prefix on page 1592
show multicast rpf summary on page 1592

Output Fields Table 75 on page 1590 describes the output fields for the **show multicast rpf** command. Output fields are listed in the approximate order in which they appear.

Table 75: show multicast rpf Output Fields

Field Name	Field Description
Instance	Name of the routing instance. (Displayed when multicast is configured within a routing instance.)
Source prefix	Prefix and length of the source as it exists in the multicast forwarding table.
Protocol	How the route was learned.
Interface	Upstream RPF interface. NOTE: The displayed interface information does not apply to bidirectional PIM RP addresses. This is because the **show multicast rpf** command does not take into account equal-cost paths or the designated forwarder. For accurate upstream RPF interface information, always use the **show pim join extensive** command when bidirectional PIM is configured.
Neighbor	Upstream RPF neighbor. NOTE: The displayed neighbor information does not apply to bidirectional PIM. This is because the **show multicast rpf** command does not take into account equal-cost paths or the designated forwarder. For accurate upstream RPF neighbor information, always use the **show pim join extensive** command when bidirectional PIM is configured.

Sample Output

show multicast rpf

```
user@host> show multicast rpf

Multicast RPF table: inet.0, 12 entries

0.0.0.0/0
    Protocol: Static

10.255.14.132/32
    Protocol: Direct
    Interface: lo0.0

10.255.245.91/32
    Protocol: IS-IS
    Interface: so-1/1/1.0
    Neighbor: 192.168.195.21

172.16.0.1/32
Inactive172.16.0.0/12
Protocol: Static
Interface: fxp0.0
```

```
Neighbor: 192.168.14.254

192.168.0.0/16
Protocol: Static
Interface: fxp0.0
Neighbor: 192.168.14.254

192.168.14.0/24
Protocol: Direct
Interface: fxp0.0

192.168.14.132/32
Protocol: Local

192.168.195.20/30
Protocol: Direct
Interface: so-1/1/1.0

192.168.195.22/32
Protocol: Local

192.168.195.36/30
Protocol: IS-IS
Interface: so-1/1/1.0
Neighbor: 192.168.195.21
```

show multicast rpf inet6

```
user@host> show multicast rpf inet6

Multicast RPF table: inet6.0, 12 entries

::10.255.14.132/128
    Protocol: Direct
    Interface: lo0.0

::10.255.245.91/128
Protocol: IS-IS
Interface: so-1/1/1.0
Neighbor: 2001:db8::2a0:a5ff:fe28:2e8c

::192.168.195.20/126
Protocol: Direct
Interface: so-1/1/1.0

::192.168.195.22/128
Protocol: Local

::192.168.195.36/126
Protocol: IS-IS
Interface: so-1/1/1.0
Neighbor: 2001:db8::2a0:a5ff:fe28:2e8c

::192.168.195.76/126
Protocol: Direct
Interface: fe-2/2/0.0

::192.168.195.77/128
Protocol: Local
```

```
2001:db8::/64
Protocol: Direct
Interface: so-1/1/1.0

2001:db8::290:69ff:fe0c:993a/128
Protocol: Local

2001:db8::2a0:a5ff:fe12:84f/128
Protocol: Direct
Interface: lo0.0

2001:db8::2/128
Protocol: PIM

2001:db8::d/128
Protocol: PIM
```

show multicast rpf prefix

```
user@host> show multicast rpf 2001:db8::/16

Multicast RPF table: inet6.0, 13 entries

2001:db8::2/128
    Protocol: PIM

2001:db8::d/128
    Protocol: PIM

...
```

show multicast rpf summary

```
user@host> show multicast rpf summary

Multicast RPF table: inet.0, 16 entries
Multicast RPF table: inet6.0, 12 entries
```

show multicast scope

List of Syntax
Syntax on page 1593
Syntax (EX Series Switch and the QFX Series) on page 1593

Syntax

show multicast scope
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

show multicast scope
<inet | inet6>
<instance *instance-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display administratively scoped IP multicast information.

Options

none—Display standard information about administratively scoped multicast information for all supported address families in all routing instances.

inet | inet6—(Optional) Display scoped multicast information for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display administratively scoped information for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

List of Sample Output

show multicast scope on page 1594
show multicast scope inet on page 1594
show multicast scope inet6 on page 1594

Output Fields

Table 76 on page 1593 describes the output fields for the **show multicast scope** command. Output fields are listed in the approximate order in which they appear.

Table 76: show multicast scope Output Fields

Field Name	Field Description
Scope name	Name of the multicast scope.

Table 76: show multicast scope Output Fields (continued)

Field Name	Field Description
Group Prefix	Range of multicast groups that are scoped.
Interface	Interface that is the boundary of the administrative scope.
Resolve Rejects	Number of kernel resolve rejects.

Sample Output

show multicast scope

```
user@host> show multicast scope
                                                          Resolve
Scope name           Group Prefix       Interface         Rejects
233-net              233.252.0.0/16     fe-0/0/0.1              0
local                233.252.0.1/16     fe-0/0/0.1              0
local                2001:db8::/16          fe-0/0/0.1                0
larry                2001:db8::1234/128     fe-0/0/0.1                0
```

show multicast scope inet

```
user@host> show multicast scope inet
                                                          Resolve
Scope name           Group Prefix       Interface         Rejects
233-net              233.252.0.0/16     fe-0/0/0.1              0
local                233.252.0.0/16     fe-0/0/0.1              0
```

show multicast scope inet6

```
user@host> show multicast scope inet6
                                                          Resolve
Scope name           Group Prefix       Interface         Rejects
local                2001:db8::/16          fe-0/0/0.1                0
larry                2001:db8::1234/128     fe-0/0/0.1                0
```

show multicast sessions

List of Syntax
Syntax on page 1595
Syntax (EX Series Switch and the QFX Series) on page 1595

Syntax
show multicast sessions
<brief | detail | extensive>
<logical-system (all | *logical-system-name*)>
<*regular-expression*>

Syntax (EX Series Switch and the QFX Series)
show multicast sessions
<brief | detail | extensive>
<*regular-expression*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display information about announced IP multicast sessions.

NOTE: On all SRX Series devices, only 100 packets can be queued during pending (S, G) route. However, when multiple multicast sessions enter the route resolve process at the same time, buffer resources are not sufficient to queue 100 packets for each session.

Options
none—Display standard information about all multicast sessions for all routing instances.

brief | detail | extensive—(Optional) Display the specified level of output.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

regular-expression—(Optional) Display information about announced sessions that match a UNIX-style regular expression.

Required Privilege Level
view

List of Sample Output
show multicast sessions on page 1597
show multicast sessions regular-expression detail on page 1597

Output Fields
Table 77 on page 1596 describes the output fields for the **show multicast sessions** command. Output fields are listed in the approximate order in which they appear.

Table 77: show multicast sessions Output Fields

Field Name	Field Description
session-name	Name of the known announced multicast sessions.

Sample Output

show multicast sessions

```
user@host> show multicast sessions
1-Department of Biological Sciences, LSU
  ...
  Monterey Bay - DockCam
  Monterey Bay - JettyCam
  Monterey Bay - StandCam
  Monterey DockCam
  Monterey DockCam / ROV cam
  ...
  NASA TV (MPEG-1)
  ...
  UO Broadcast - NASA Videos - 25 Years of Progress
  UO Broadcast - NASA Videos - Journey through the Solar System
  UO Broadcast - NASA Videos - Life in the Universe
  UO Broadcast - NASA Videos - Nasa and the Airplane
  UO Broadcasts OPB's Oregon Story
  UO DOD News Clips
  UO Medical Management of Biological Casualties (1)
  UO Medical Management of Biological Casualties (2)
  UO Medical Management of Biological Casualties (3)
...
  376 active sessions.
```

show multicast sessions regular-expression detail

```
user@host> show multicast sessions "NASA TV" detail
SDP Version: 0  Originated by: -@10.223.83.33
 Session: NASA TV (MPEG-1)
 Description: NASA television in MPEG-1 format, provided by Private University.
Please contact the UO if you have problems with this feed.
 Email: Your Name Here <multicast@lists.private.edu>
 Phone: Your Name Here <888/555-1212>
Bandwidth: AS:1000
Start time: permanent
Stop time: none
Attribute: type:broadcast
Attribute: tool:IP/TV Content Manager 3.4.14
Attribute: live:capture:1
Attribute: x-iptv-capture:mp1s
Media: video 54302 RTP/AVP 32 31 96 97
Connection Data: 233.252.0.45 ttl 127
Attribute: quality:8
Attribute: framerate:30
Attribute: rtpmap:96 WBIH/90000
Attribute: rtpmap:97 MP4V-ES/90000
Attribute: x-iptv-svr:video 10.223.91.191 live
Attribute: fmtp:32 type=mpeg1
Media: audio 28848 RTP/AVP 14 0 96 3 5 97 98 99 100 101 102 10 11 103 104 105 106
 Connection Data: 224.2.145.37 ttl 127
Attribute: rtpmap:96 X-WAVE/8000
Attribute: rtpmap:97 L8/8000/2
Attribute: rtpmap:98 L8/8000
Attribute: rtpmap:99 L8/22050/2
Attribute: rtpmap:100 L8/22050
Attribute: rtpmap:101 L8/11025/2
Attribute: rtpmap:102 L8/11025
```

```
Attribute: rtpmap:103 L16/22050/2
Attribute: rtpmap:104 L16/22050

  1 matching sessions.
```

show multicast snooping next-hops

Syntax
show multicast snooping next-hops
<brief | detail>
<identifier *next-hop-ID*>
<inet>
<inet6>
<logical-system *logical-system-name*>

Release Information
Command introduced in Junos OS Release 11.2.

Description
Display information about the IP multicast snooping next-hops.

Options
brief | detail—(Optional) Display the specified level of output.

inet—(Optional) Display information for IPv4 multicast next hops only. If a family is not specified, both IPv4 and IPv6 results will be shown.

inet6—(Optional) Display information for IPv6 multicast next hops only. If a family is not specified, both IPv4 and IPv6 results will be shown.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

Required Privilege Level
view

List of Sample Output
show multicast snooping next-hops on page 1601
show multicast snooping next-hops (IGMP snooping enabled on a VPLS) on page 1601

Output Fields
Table 78 on page 1599 describes the output fields for the **show multicast snooping next-hops** command. Output fields are listed in the approximate order in which they appear.

Table 78: show multicast snooping next-hops Output Fields

Field Name	Field Description
Family	Protocol family for which multicast snooping next hops are displayed: **INET** or **INET6**.
Refcount	Number of cache entries that are using this next hop.
KRefcount	Kernel reference count for the next hop.
Downstream interface	Interface names associated with each multicast next-hop ID.

Table 78: show multicast snooping next-hops Output Fields (continued)

Field Name	Field Description
Nexthop Id	Identifier for the next-hop. NOTE: To see the next-hop ID for a given PE mesh group, **igmp-snooping** must be enabled for the relevant VPLS routing instance. (Junos OS creates a default CE and VE mesh groups for each VPLS routing instance. The next hop of the VE mesh group is the set of VE mesh-group interfaces of the remaining PEs in the same VPLS routing instance.)

Sample Output

show multicast snooping next-hops

```
user@host> show multicast snooping next-hops
Family: INET
ID          Refcount KRefcount Downstream interface Nexthop Id
1048574            4         1 ge-0/1/0.1000
                               ge-0/1/2.1000
                               ge-0/1/3.1000

1048574            4         1 ge-0/1/0.1000-(2000)
                               1048575
                               1048576

1048575            2         0 ge-0/1/2.1000-(2001)
                               ge-0/1/3.1000-(2002)

1048576            2         0 lsi.1048578-(2003)
                               lsi.1048579-(2004)
```

show multicast snooping next-hops (IGMP snooping enabled on a VPLS)

In thIS example, ID 1048585 is the VE next-hop ID created for the VE next hop that is holding VE interfaces for the routing instance. It only appears if igmp snooping is enabled on the VPLS.

```
user@host> show multicast snooping next-hops
Family: INET
ID          Refcount KRefcount Downstream interface Addr
1048588            2         1 1048585
1048589            2         1 1048585
                               ge-0/0/5.100
0                  2         0 ge-0/0/0.100
                               ge-0/0/1.100
1048583            2         1 local
1048587            2         1 local
                               1048585
1048586            4         2 local
                               1048585
                               ge-0/0/5.100
1048584            2         1 local
                               ge-0/0/5.100
1048582            6         2 ge-0/0/5.100
0                  2         0 ge-0/0/0.200
                               ge-0/0/2.200
0                  2         0 ge-0/0/0.300
                               ge-0/0/2.300
0                  1         0 vt-0/0/10.17825792
                               vt-0/0/10.17825793
0                  1         0 vt-0/0/10.1048576
                               vt-0/0/10.1048578
1048585            5         0 vt-0/0/10.1048577
                               vt-0/0/10.1048579
0                  1         0 vt-0/0/10.34603008
                               vt-0/0/10.34603009
```

show multicast snooping route

Syntax

show multicast snooping route
<*regexp*>
<active>
<all>
<bridge-domain *bridge-domain-name*>
<brief >
<control>
<data>
<detail >
<extensive>
<group *group*>
<inactive>
<inet>
<inet6>
<instance *instance-name*>
<logical-system *logical-system-name*>
<mesh-group *mesh-group-name*>
<qualified-vlan *vlan-id*>
<source-prefix *source-prefix*>
<vlan *vlan-id*>

Release Information

Command introduced in Junos OS Release 8.5.
Support for **control**, **data**, **qualified-vlan** and **vlan** options introduced in Junos OS Release 13.3 for EX Series switches.

Description

Display the entries in the IP multicast snooping forwarding table. You can display some of this information with the **show route table inet.1** command.

Options

none—Display standard information about all entries in the multicast snooping table for all virtual switches and all bridge domains.

active | all | inactive —(Optional) Display all active entries, all entries, or all inactive entries, respectively, in the multicast snooping table.

bridge-domain *bridge-domain*—(Optional) Display the entries for a particular bridge domain.

brief | detail | extensive—(Optional) Display the specified level of output.

control—(Optional) Display control route entries.

data—(Optional) Display data route entries.

group *group*—(Optional) Display the entries for a particular group.

inet—(Optional) Display IPv4 information.

inet6—(Optional) Display IPv6 information.

instance *instance-name*—(Optional) Display the entries for a multicast instance.

logical-system ***logical-system-name***—(Optional) Display information about a particular logical system, or type 'all'.

mesh-group ***mesh-group-name***—(Optional) Display the entries for a particular mesh group.

qualified-vlan ***vlan-id***—(Optional) Display the entries for a particular qualified VLAN.

regexp—(Optional) Display information about the multicast forwarding table entries that match a UNIX-style regular expression.

source-prefix ***source-prefix***—(Optional) Display the entries for a particular source prefix.

vlan ***vlan-id***—(Optional) Display the entries for a particular VLAN.

Required Privilege Level view

List of Sample Output
show multicast snooping route bridge-domain on page 1604
show multicast snooping route instance vs on page 1604
show multicast snooping route extensive on page 1604

Output Fields Table 79 on page 1603 describes the output fields for the **show multicast snooping route** command. Output fields are listed in the approximate order in which they appear.

Table 79: show multicast snooping route Output Fields

Field Name	Field Description	Level of Output
Nexthop Bulking	Displays whether next-hop bulk updating is **ON** or **OFF** (only for routing-instance type of **virtual switch** or **vpls**).	All levels
Family	IPv4 address family (**INET**) or IPv6 address family (**INET6**).	All levels
Group	Group address.	All levels
Source	Prefix and length of the source as it is in the multicast forwarding table. For (*,G) entries, this field is set to "*".	All levels
Routing-instance	Name of the routing instance to which this routing information applies. (Displayed when multicast is configured within a routing instance.)	All levels
Learning Domain	Name of the learning domain to which this routing information applies.	**detail extensive**
Statistics	Rate at which packets are being forwarded for this source and group entry (in Kbps and pps), and number of packets that have been forwarded to this prefix.	**detail extensive**
Next-hop ID	Next-hop identifier of the prefix. The identifier is returned by the router's Packet Forwarding Engine and is also displayed in the output of the **show multicast nexthops** command.	**detail extensive**
Route state	Whether the group is **Active** or **Inactive**.	**extensive**

Table 79: show multicast snooping route Output Fields (continued)

Field Name	Field Description	Level of Output
Forwarding state	Whether the prefix is **Pruned** or **Forwarding**.	**extensive**
Cache lifetime/timeout	Number of seconds until the prefix is removed from the multicast forwarding table. A value of **never** indicates a permanent forwarding entry.	**extensive**

Sample Output

show multicast snooping route bridge-domain

```
user@host> show multicast snooping route bridge-domain br-dom-1 extensive
Family: INET

Group: 232.1.1.1
    Source: 192.168.3.100/32
    Downstream interface list:
        ge-0/1/0.200
    Statistics: 0 kBps, 0 pps, 1 packets
    Next-hop ID: 1048577
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 240 seconds
```

show multicast snooping route instance vs

```
user@host> show multicast snooping route instance vs
Nexthop Bulking: ON

Family: INET

Group: 224.0.0.0
    Bridge-domain: vsid500

Group: 225.1.0.1
    Bridge-domain: vsid500
    Downstream interface list: vsid500
       ge-0/3/8.500 ge-1/1/9.500 ge1/2/5.500
```

show multicast snooping route extensive

```
user@host> show multicast snooping route extensive inet6 group ff03::1
Nexthop Bulking: OFF

Family: INET6
Group: ff03::1/128
    Source: ::
    Bridge-domain: BD-1
    Mesh-group: __all_ces__
    Downstream interface list:
        ae0.1 -(562) 1048576
    Statistics: 2697 kBps, 3875 pps, 758819039 packets
    Next-hop ID: 1048605
    Route state: Active
    Forwarding state: Forwarding
```

```
Group: ff03::1/128
    Source: 6666::2/128
    Bridge-domain: BD-1
    Mesh-group: __all_ces__
    Downstream interface list:
        ae0.1 -(562) 1048576
    Statistics: 0 kBps, 0 pps, 0 packets
    Next-hop ID: 1048605
    Route state: Active
    Forwarding state: Forwarding
```

show multicast statistics

Syntax

show multicast statistics
<inet | inet6>
<instance *instance-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

Release Information

Command introduced before Junos OS Release 7.4.
interface option introduced in Junos OS Release 16.1 for the MX Series.

Description

Display IP multicast statistics.

Options

none—Display multicast statistics for all supported address families for all routing instances.

inet | inet6—(Optional) Display multicast statistics for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display statistics for a specific routing instance.

interface *interface-name*—(Optional) Display statistics for a specific interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Additional Information

The input and output interface multicast statistics are consistent, but not timely. They are constructed from the forwarding statistics, which are gathered at 30-second intervals. Therefore, the output from this command always lags the true count by up to 30 seconds.

Required Privilege Level

view

Related Documentation

- clear multicast statistics on page 1414

List of Sample Output

show multicast statistics on page 1609
show multicast statistics interface on page 1609

Output Fields

Table 80 on page 1606 describes the output fields for the **show multicast statistics** command. Output fields are listed in the approximate order in which they appear.

Table 80: show multicast statistics Output Fields

Field Name	Field Description
Family	Protocol family for which multicast statistics are displayed: **INET** or **INET6**.

Table 80: show multicast statistics Output Fields (continued)

Field Name	Field Description
Interface	Name of the interface for which statistics are being reported.
Routing Protocol	Primary multicast protocol on the interface: **PIM**, **DVMRP for INET**, or **PIM for INET6**.
Mismatch	Number of multicast packets that did not arrive on the correct upstream interface.
Kernel Resolve	Number of resolve requests processed by the primary multicast protocol on the interface.
Resolve No Route	Number of resolve requests that were ignored because there was no route to the source.
In Kbytes	Total accumulated incoming packets (in KB) since the last time the **clear multicast statistics** command was issued.
Out Kbytes	Total accumulated outgoing packets (in KB) since the last time the **clear multicast statistics** command was issued.
Mismatch error	Number of mismatches that were ignored because of internal errors.
Mismatch No Route	Number of mismatches that were ignored because there was no route to the source.
Routing Notify	Number of times that the multicast routing system has been notified of a new multicast source by a multicast routing protocol .
Resolve Error	Number of resolve requests that were ignored because of internal errors.
In Packets	Total number of incoming packets since the last time the **clear multicast statistics** command was issued.
Out Packets	Total number of outgoing packets since the last time the **clear multicast statistics** command was issued.
Resolve requests on interfaces not enabled for multicast *n*	Number of resolve requests on interfaces that are not enabled for multicast that have accumulated since the **clear multicast statistics** command was last issued.
Resolve requests with no route to source *n*	Number of resolve requests with no route to the source that have accumulated since the **clear multicast statistics** command was last issued.
Routing notifications on interfaces not enabled for multicast *n*	Number of routing notifications on interfaces not enabled for multicast that have accumulated since the **clear multicast statistics** command was last issued.
Routing notifications with no route to source *n*	Number of routing notifications with no route to the source that have accumulated since the **clear multicast statistics** command was last issued.
Interface Mismatches on interfaces not enabled for multicast *n*	Number of interface mismatches on interfaces not enabled for multicast that have accumulated since the **clear multicast statistics** command was last issued.

Table 80: show multicast statistics Output Fields (continued)

Field Name	Field Description
Group Membership on interfaces not enabled for multicast *n*	Number of group memberships on interfaces not enabled for multicast that have accumulated since the **clear multicast statistics** command was last issued.

Sample Output

show multicast statistics

```
user@host> show multicast statistics
    Address family: INET
    Interface: fe-0/0/0
        Routing Protocol:          PIM   Mismatch error:                0
        Mismatch:                    0   Mismatch No Route:             0
        Kernel Resolve:             10   Routing Notify:                0
        Resolve No Route:            0   Resolve Error:                 0
        In Kbytes:                4641   In Packets:                50454
        Out Kbytes:                  0   Out Packets:                   0
    Interface: so-0/1/1.0
        Routing Protocol:          PIM   Mismatch error:                0
        Mismatch:                    0   Mismatch No Route:             0
        Kernel Resolve:              0   Routing Notify:                0
        Resolve No Route:            0   Resolve Error:                 0
        In Kbytes:                   0   In Packets:                    0
        Out Kbytes:               4641   Out Packets:               50454

    Resolve requests on interfaces not enabled for multicast 0
    Resolve requests with no route to source 0
    Routing notifications on interfaces not enabled for multicast 0
    Routing notifications with no route to source 0
    Interface Mismatches on interfaces not enabled for multicast 0
    Group Membership on interfaces not enabled for multicast 25

    Address family: INET6
    Interface: fe-0/0/0.0
        Routing Protocol:          PIM   Mismatch error:                0
        Mismatch:                    0   Mismatch No Route:             0
        Kernel Resolve:              0   Routing Notify:                0
        Resolve No Route:            0   Resolve Error:                 0
        In Kbytes:                   0   In Packets:                    0
        Out Kbytes:                  0   Out Packets:                   0
    Interface: so-0/1/1.0
        Routing Protocol:          PIM   Mismatch error:                0
        Mismatch:                    0   Mismatch No Route:             0
        Kernel Resolve:              0   Routing Notify:                0
        Resolve No Route:            0   Resolve Error:                 0
        In Kbytes:                   0   In Packets:                    0
        Out Kbytes:                  0   Out Packets:                   0

    Resolve requests on interfaces not enabled for multicast 0
    Resolve requests with no route to source 0
    Routing notifications on interfaces not enabled for multicast 0
    Routing notifications with no route to source 0
    Interface Mismatches on interfaces not enabled for multicast 0
    Group Membership on interfaces not enabled for multicast 0
```

show multicast statistics interface

```
user@host> show multicast statistics interface vt-3/0/10.2097152
Instance: master Family: INET
Interface: vt-3/0/10.2097152
    Routing protocol:          PIM   Mismatch error:              0
    Mismatch:                    0   Mismatch no route:           0
    Kernel resolve:              0   Routing notify:              0
    Resolve no route:            0   Resolve error:               0
```

```
Resolve filtered:              0   Notify filtered:              0
In kbytes:                     0   In packets:                   0
Out kbytes:                    0   Out packets:                  0
```

show multicast usage

List of Syntax
Syntax on page 1611
Syntax (EX Series Switch and the QFX Series) on page 1611

Syntax
show multicast usage
<brief | detail>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show multicast usage
<brief | detail>
<inet | inet6>
<instance *instance-name*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display usage information about the 10 most active Distance Vector Multicast Routing Protocol (DVMRP) or Protocol Independent Multicast (PIM) groups.

Options
none—Display multicast usage information for all supported address families for all routing instances.

brief | detail—(Optional) Display the specified level of output.

inet | inet6—(Optional) Display usage information for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display information about the most active DVMRP or PIM groups for a specific multicast instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show multicast usage on page 1612
show multicast usage brief on page 1612
show multicast usage instance on page 1612
show multicast usage detail on page 1613

Output Fields
Table 81 on page 1612 describes the output fields for the **show multicast usage** command. Output fields are listed in the approximate order in which they appear.

Table 81: show multicast usage Output Fields

Field Name	Field Description
Instance	Name of the routing instance. (Displayed when multicast is configured within a routing instance.)
Group	Group address.
Sources	Number of sources.
Packets	Number of packets that have been forwarded to this prefix. If one or more of the packets forwarded statistic queries fails or times out, the packets field displays **unavailable**.
Bytes	Number of bytes that have been forwarded to this prefix. If one or more of the packets forwarded statistic queries fails or times out, the bytes field displays **unavailable**.
Prefix	IP address.
/len	Prefix length.
Groups	Number of multicast groups.

Sample Output

show multicast usage

```
user@host> show multicast usage
Group           Sources Packets             Bytes
233.252.0.0       1       52847               4439148
233.252.0.1       2       13450               1125530


Prefix          /len Groups Packets             Bytes
10.255.14.144   /32  2      66254               5561304
10.255.70.15    /32  1      43                  3374...
```

show multicast usage brief

The output for the **show multicast usage brief** command is identical to that for the **show multicast usage** command. For sample output, see show multicast usage on page 1612.

show multicast usage instance

```
user@host> show multicast usage instance VPN-A
Group           Sources Packets             Bytes
233.252.0.254   1       5538                509496
233.252.0.39    1       13                  624
233.252.0.40    1       13                  624

Prefix          /len Groups Packets             Bytes
192.168.195.34  /32  1      5538                509496
```

```
10.255.14.30    /32  1      13                  624
10.255.245.91   /32  1      13                  624
...
```

show multicast usage detail

```
user@host> show multicast usage detail
Group           Sources Packets              Bytes
233.252.0.0       1       53159                4465356
    Source: 10.255.14.144   /32 Packets: 53159 Bytes: 4465356
233.252.0.1       2       13450                1125530
    Source: 10.255.14.144   /32 Packets: 13407 Bytes: 1122156
    Source: 10.255.70.15    /32 Packets: 43 Bytes: 3374


Prefix          /len Groups Packets              Bytes
10.255.14.144   /32  2      66566                5587512
    Group: 233.252.0.0       Packets: 53159 Bytes: 4465356
    Group: 233.252.0.1       Packets: 13407 Bytes: 1122156
10.255.70.15    /32  1      43                   3374
    Group: 233.252.0.1       Packets: 43 Bytes: 3374
```

show mvpn c-multicast

Syntax show mvpn c-multicast
<extensive | summary>
<instance-name *instance-name*>
<source-pe>

Release Information Command introduced in Junos OS Release 8.4.
Option to show **source-pe** introduced in Junos OS Release 15.1.

Description Display the multicast VPN customer multicast route information.

Options **extensive | summary**—(Optional) Display the specified level of output.

instance-name *instance-name*—(Optional) Display output for the specified routing instance.

source-pe—(Optional) Display source-pe output for the specified c-multicast entries.

Required Privilege Level view

List of Sample Output
show mvpn c-multicast on page 1615
show mvpn c-multicast summary on page 1615
show mvpn c-multicast extensive on page 1615
show mvpn c-multicast source-pe on page 1616

Output Fields Table 82 on page 1614 lists the output fields for the **show mvpn c-multicast** command. Output fields are listed in the approximate order in which they appear.

Table 82: show mvpn c-multicast Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the VPN routing instance.	summary extensive none
C-mcast IPv4 (S:G)	Customer router IPv4 multicast address.	extensive none
Ptnl	Provider tunnel attributes, *tunnel type:tunnel source, tunnel destination group*.	extensive none
St	State: • DS—Represents (S,G) and is created due to (*,G) • RM—Remote VPN route learned from the remote PE router • St display blank—SSM group join	extensive none
MVPN instance	Name of the multicast VPN routing instance	extensive none

Table 82: show mvpn c-multicast Output Fields (continued)

Field Name	Field Description	Level of Output
C-multicast IPv4 route count	Number of customer multicast IPv4 routes associated with the multicast VPN routing instance.	summary
C-multicast IPv6 route count	Number of customer multicast IPv6 routes associated with the multicast VPN routing instance.	summary

Sample Output

show mvpn c-multicast

```
user@host> show mvpn c-multicast
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:203.0.113.1/24 PIM-SM:10.255.14.144, 198.51.100.1        RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:203.0.113.0/24 PIM-SM:10.255.14.144, 198.51.100.2        RM
```

show mvpn c-multicast summary

```
user@host> show mvpn c-multicast summary
MVPN Summary:
Family: INET
Family: INET6

Instance: mvpn1
  C-multicast IPv6 route count: 1
```

show mvpn c-multicast extensive

```
user@host> show mvpn c-multicast extensive
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                              St
```

```
  192.168.195.78/32:203.0.113.1/24 PIM-SM:10.255.14.144, 198.51.100.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:203.0.113.0/24 PIM-SM:10.255.14.144, 198.51.100.2         RM
```

show mvpn c-multicast source-pe

```
user@host> show mvpn c-multicast source-pe
Family : INET
Family : INET6

Instance : mvpn1
  MVPN Mode : RPT-SPT
    C-Multicast route address: ::/0:ff05::1/128
        MVPN Source-PE1:
                extended-community: no-advertise target:10.1.0.0:9
                Route Distinguisher: 10.1.0.0:1
                Autonomous system number: 1
                Interface: ge-0/0/9.1 Index: 343
        PIM Source-PE1:
                extended-community: target:10.1.0.0:9
                Route Distinguisher: 10.1.0.0:1
                Autonomous system number: 1
                Interface: ge-0/0/9.1 Index: 343
```

show mvpn instance

Syntax

show mvpn instance
<*instance-name*>
<display-tunnel-name>
<extensive | summary>
<inet | inet6>
<logical-system>

Release Information

Command introduced in Junos OS Release 8.4.
Additional details in output for extensive option introduced in Junos OS Release 15.1.

Description

Display the multicast VPN routing instance information according the options specified.

Options

instance-name—(Optional) Display statistics for the specified routing instance, or press Enter without specifying an instance name to show output for all instances.

display-tunnel-name—(Optional) Display the ingress provider tunnel name rather than the attribute.

extensive | summary—(Optional) Display the specified level of output.

inet | inet6—(Optional) Display output for the specified IP type.

inet | inet6—(Optional) Display output for the specified IP type.

logical-system—(Optional) Display details for the specified logical system, or type “all”.

Required Privilege Level

view

List of Sample Output

show mvpn instance on page 1618
show mvpn instance summary on page 1619
show mvpn instance extensive on page 1619
show mvpn instance summary (IPv6) on page 1620

Output Fields

Table 83 on page 1617 lists the output fields for the **show mvpn instance** command. Output fields are listed in the approximate order in which they appear.

Table 83: show mvpn instance Output Fields

Field Name	Field Description	Level of Output
MVPN instance	Name of the multicast VPN routing instance	extensive none
Instance	Name of the VPN routing instance.	summary extensive none
Provider tunnel	Provider tunnel attributes, *tunnel type:tunnel source, tunnel destination group*.	extensive none

Table 83: show mvpn instance Output Fields (continued)

Field Name	Field Description	Level of Output
Neighbor	Address, type of provider tunnel (I-P-tnl, inclusive provider tunnel and S-P-tnl, selective provider tunnel) and provider tunnel for each neighbor.	extensive none
C-mcast IPv4 (S:G)	Customer IPv4 router multicast address.	extensive none
C-mcast IPv6 (S:G)	Customer IPv6 router multicast address.	extensive none
Ptnl	Provider tunnel attributes, *tunnel type:tunnel source, tunnel destination group*.	extensive none
St	State: • DS—Represents (S,G) and is created due to (*,G) • RM—Remote VPN route learned from the remote PE router • St display blank—SSM group join	extensive none
Neighbor count	Number of neighbors associated with the multicast VPN routing instance.	summary
C-multicast IPv4 route count	Number of customer multicast IPv4 routes associated with the multicast VPN routing instance.	summary
C-multicast IPv6 route count	Number of customer multicast IPv6 routes associated with the multicast VPN routing instance.	summary

Sample Output

show mvpn instance

```
user@host> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 198.51.100.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 198.51.100.1
  10.255.70.17                          PIM-SM:10.255.70.17, 198.51.100.1
  C-mcast IPv4 (S:G)            Ptnl                                St
  192.168.195.78/32:203.0.113.0/24 PIM-SM:10.255.14.144, 198.51.100.1        RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
```

```
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 198.51.100.2
  Neighbor                                I-P-tnl
  10.255.14.160                           PIM-SM:10.255.14.160, 198.51.100.2
  10.255.70.17                            PIM-SM:10.255.70.17, 198.51.100.2
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:203.0.113.1/24 PIM-SM:10.255.14.144, 198.51.100.2        RM
```

Sample Output

show mvpn instance summary

```
user@host> show mvpn instance summary
MVPN Summary:
Family: INET
Family: INET6

Instance: mvpn1
  Sender-Based RPF: Disabled. Reason: Not enabled by configuration.
  Hot Root Standby: Disabled. Reason: Not enabled by configuration.
  Neighbor count: 3
  C-multicast IPv6 route count: 1
```

Sample Output

show mvpn instance extensive

```
user@host> show mvpn instance extensive
MVPN instance:
Family : INET

Instance : vpn_blue
    Customer Source: 10.1.1.1
        RT-Import Target: 192.168.1.1:100
        Route-Distinguisher: 192.168.1.1:100
        Source-AS: 65000
        Via unicast route: 10.1.0.0/16 in vpn-blue.inet.0
        Candidate Source PE Set:
            RT-Import 192.168.1.1:100, RD 1111:22222, Source-AS 65000
            RT-Import 192.168.2.2:100, RD 1111:22222, Source-AS 65000
            RT-Import 192.168.3.3:100, RD 1111:22222, Source-AS 65000

‘Extensive’ output will show everything in ‘detail’ output and add the list of
bound c-multicast routes.

> show mvpn source 10.1.1.1 instance vpn_blue extensive

Family : INET

Instance : vpn_blue
    Customer Source: 10.1.1.1
        RT-Import Target: 192.168.1.1:100
        Route-Distinguisher: 192.168.1.1:100
        Source-AS: 65000
        Via unicast route: 10.1.0.0/16 in vpn-blue.inet.0
        Candidate Source PE Set:
            RT-Import 192.168.1.1:100, RD 1111:22222, Source-AS 65000
            RT-Import 192.168.2.2:100, RD 1111:22222, Source-AS 65000
            RT-Import 192.168.3.3:100, RD 1111:22222, Source-AS 65000
        Customer-Multicast Routes:
```

```
                    10.1.1.1/32:198.51.100.3/24
                    10.1.1.1/32:198.51.100.3/24
```

show mvpn instance summary (IPv6)

```
user@host> show mvpn instance summary
MVPN Summary:
Instance: VPN-A
  C-multicast IPv6 route count: 2
Instance: VPN-B
  C-multicast IPv6 route count: 2
```

show mvpn neighbor

Syntax

show mvpn neighbor
<extensive | summary>
<inet | inet6>
<instance *instance-name* | neighbor-address *address*>
<logical-system *logical-system-name*>

Release Information

Command introduced in Junos OS Release 8.4.

Description

Display multicast VPN neighbor information.

Options

extensive | summary—(Optional) Display the specified level of output for all multicast VPN neighbors.

inet | inet6—(Optional) Display IPv4 or IPv6 information for all multicast VPN neighbors.

instance *instance-name* | neighbor-address *address*—(Optional) Display multicast VPN neighbor information for the specified instance or the specified neighbor.

logical-system *logical-system-name*—(Optional) Display multicast VPN neighbor information for the specified logical system.

Required Privilege Level

view

List of Sample Output

show mvpn neighbor on page 1622
show mvpn neighbor extensive on page 1622
show mvpn neighbor extensive on page 1623
show mvpn neighbor instance-name on page 1623
show mvpn neighbor neighbor-address on page 1623
show mvpn neighbor neighbor-address summary on page 1624
show mvpn neighbor neighbor-address extensive on page 1624
show mvpn neighbor neighbor-address instance-name on page 1624
show mvpn neighbor summary on page 1625

Output Fields

Table 84 on page 1621 lists the output fields for the **show mvpn neighbor** command. Output fields are listed in the approximate order in which they appear.

Table 84: show mvpn neighbor Output Fields

Field Name	Field Description	Level of Output
MVPN instance	Name of the multicast VPN routing instance	**extensive** none
Instance	Name of the VPN routing instance.	**summary extensive** none

Table 84: show mvpn neighbor Output Fields (continued)

Field Name	Field Description	Level of Output
Neighbor	Address, type of provider tunnel (**I-P-tnl**, inclusive provider tunnel and **S-P-tnl**, selective provider tunnel) and provider tunnel for each neighbor.	**extensive** none
Provider tunnel	Provider tunnel attributes, ***tunnel type:tunnel source, tunnel destination group***.	**extensive** none

Sample Output

show mvpn neighbor

```
user@host> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.1
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.2
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.2
```

Sample Output

show mvpn neighbor extensive

```
user@host> show mvpn neighbor extensive
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.1
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel
```

```
Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.2
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.2
```

show mvpn neighbor extensive

```
user@host> show mvpn neighbor extensive
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: mvpn-a
  Neighbor                              I-P-tnl
  10.255.72.45
  10.255.72.50                          LDP P2MP:10.255.72.50, lsp-id 1
```

Sample Output

show mvpn neighbor instance-name

```
user@host> show mvpn neighbor instance-name VPN-A
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.1
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.1
```

Sample Output

show mvpn neighbor neighbor-address

```
user@host> show mvpn neighbor neighbor-address 10.255.14.160
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel
```

```
Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 192.0.2.2
```

Sample Output

show mvpn neighbor neighbor-address summary

```
user@host> show mvpn neighbor neighbor-address 10.255.70.17 summary
MVPN Summary:
Instance: VPN-A
Instance: VPN-B
```

Sample Output

show mvpn neighbor neighbor-address extensive

```
user@host> show mvpn neighbor neighbor-address 10.255.70.17 extensive
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.2
```

Sample Output

show mvpn neighbor neighbor-address instance-name

```
user@host> show mvpn neighbor neighbor-address 10.255.70.17 instance-name VPN-A
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.70.17                          PIM-SM:10.255.70.17, 192.0.2.1
```

Sample Output

show mvpn neighbor summary

```
user@host> show mvpn neighbor summary
MVPN Summary:
Family: INET
Family: INET6

Instance: mvpn1
  Neighbor count: 3
```

show mvpn suppressed

Syntax show mvpn suppressed
<*instance-name*>
<general | mvpn-rpt>
<inet | inet6>

Release Information Command introduced in Junos OS Release16.1.

Description MVPN maintains a list of suppressed customer-multicast states and the reason they were suppressed. Display it, for example, to help understand the enforcement of forwarding-cache limits

Options ***instance-name***—(Optional) Display statistics for the specified routing instance, or press Enter without specifying an instance name to show output for all instances.

general | mvpn-rpt —(Optional) Display suppressed multicast prefixes and reason they were suppressed.

inet | inet6—(Optional) Display output for the specified IP type.

Required Privilege Level view

List of Sample Output show mvpn suppressed on page 1626
show mvpn suppressed summary on page 1627

Output Fields Table 83 on page 1617 lists the output fields for the **show mvpn suppressed** command. Output fields are listed in the approximate order in which they appear.

Table 85: show mvpn suppressed Output Fields

Field Name	Field Description
MVPN instance	Name of the multicast VPN routing instance.
Prefix	Shown as a single line per prefix, group followed by source.
reason	MVPN *,G entries are deleted either because they exceed either the general forwarding-cache limit or because they exceed the forwarding-cache limit set for MVPN RPT.

Sample Output

show mvpn suppressed

```
user@host> show mvpn suppressed instance name
Instance: mvpn1 Family: INET

  Prefix 0.0.0.0/0:239.1.1.1/32, Suppressed due to MVPN RPT forwarding-cache limit
```



```
  Instance: mvpn1 Family: INET6
  Prefix ::91.1.1.1/128:Ff05::1/128, Suppressed due to general forwarding-cache
limit
  Prefix ::/0:ff05::2/128, Suppressed due to general forwarding-cache limit
  Prefix ::/0:ff05::3/128, Suppressed due to MVPN RPT forwarding-cache limit
```

Sample Output

show mvpn suppressed summary

```
user@host> show mvpn suppressed instance name summary
Instance: mvpn1 Family: INET

  General entries suppressed:    5
  MVPN RPT entries suppressed:   1

  Instance: mvpn1 Family: INET6
  General entries suppressed:    5
  MVPN RPT entries suppressed:   1
```

show policy

List of Syntax
Syntax on page 1628
Syntax (EX Series Switches) on page 1628

Syntax
show policy
<logical-system (all | *logical-system-name*)>
<*policy-name*>
<*statistics* >

Syntax (EX Series Switches)
show policy
<*policy-name*>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
statistics option introduced in Junos OS Release 16.1 for MX Series routers.

Description
Display information about configured routing policies.

Options
none—List the names of all configured routing policies.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

policy-name—(Optional) Show the contents of the specified policy.

statistics—(Optional) Use in conjunction with the **test policy** command to show the length of time (in microseconds) required to evaluate a given policy and the number of times it has been executed. This information can be used, for example, to help structure a policy so it is evaluated efficiently. Timers shown are per route; times are not cumulative. Statistics are incremented even when the router is learning (and thus evaluating) routes from peering routers.

Required Privilege Level
view

Related Documentation
- *show policy damping*
- *test policy*

List of Sample Output
show policy on page 1629
show policy policy-name on page 1629
show policy statistics policy-name on page 1629
show policy (Multicast Scoping) on page 1630
show policy (Route Filter and source Address Filter Lists) on page 1630

Output Fields Table 86 on page 1629 lists the output fields for the **show policy** command. Output fields are listed in the approximate order in which they appear.

Table 86: show policy Output Fields

Field Name	Field Description
policy-name	Name of the policy listed.
term	Name of the user-defined policy term. The term name **unnamed** is used for policy elements that occur outside of user defined terms
from	Match condition for the policy.
then	Action for the policy.

Sample Output

show policy

```
user@host> show policy
Configured policies:
__vrf-export-red-internal__
__vrf-import-red-internal__
red-export
rf-test-policy
multicast-scoping
```

show policy policy-name

```
user@host> show policy vrf-import-red-internal
Policy vrf-import-red-internal:
    from
         203.0.113.0/28  accept
         203.0.113.32/28  accept
    then reject
```

show policy statistics policy-name

```
user@host> show policy statistics iBGP-v4-RR-Import
Policy iBGP-v4-RR-Import:
    [1243328] Term Lab-Infra:
        from [1243328 0]  proto BGP
             [28 0] route filter:
                 10.11.0.0/8 orlonger
                 10.13.0.0/8 orlonger
        then [28 0] accept
    [1243300] Term External:
        from [1243300 1]  proto BGP
         [1243296 0]  community Ext-Com1 [64496:1515 ]
         [1243296 0]  prefix-list-filter Customer-Routes
         [1243296 0]  aspath AS6221
             [1243296 1] route filter:
                 172.16.49.0/12 orlonger
                 172.16.50.0/12 orlonger
                 172.16.51.0/12 orlonger
```

```
                    172.16.52.0/12 orlonger
                    172.16.56.0/12 orlonger
                    172.16.60.0/12 orlonger
                then [1243296 2] community + Ext-Com2 [64496:2000 ] [1243296 0] accept
            [4] Term Final:
                then [4 0] reject
```

show policy (Multicast Scoping)

```
user@host> show policy multicast-scoping
Policy multicast-scoping:
    from
         multicast-scope == 8
  then
    accept
```

show policy (Route Filter and source Address Filter Lists)

```
user@host> show policy rf-test-policy
Policy rf-test-policy:
    Term term1:
        from source-address-filter-list saf-list-1
             source-address filter:
                 192.0.2.0/29 longer
                 192.0.2.64/28 exact
                 192.0.2.128/28 exact
                 192.0.2.160/28 orlonger
    Term term2:
        from route-filter-list rf-list-1
             route filter:
                 198.51.100.0/29 upto 198.51.100.0/30
                 198.51.100.8/29 upto 198.51.100.8/30 accept
    Term unnamed:
        then reject
```

show pim bidirectional df-election

Syntax

show pim bidirectional df-election
<brief | detail >
<inet | inet6>
<instance *instance name*>
<logical-system (all | *logical-system-name*)>
<rpa *address*>

Release Information

Command introduced in Junos OS Release 12.1.

Description

For bidirectional PIM, display the designated forwarder (DF) election results for each interface grouped by the rendezvous point addresses (RPAs).

Options

none—Display standard information about all interfaces.

brief | detail—(Optional) Display the specified level of output.

inet | inet6—(Optional) Display DF election results for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display DF election results for a specific routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

rpa *address*—(Optional) Display the DF election results for an RP address.

Required Privilege Level

view

List of Sample Output

show pim bidirectional df-election on page 1632
show pim bidirectional df-election brief on page 1632

Output Fields

Table 87 on page 1631 describes the output fields for the **show pim bidirectional df-election** command. Output fields are listed in the approximate order in which they appear.

Table 87: show pim bidirectional df-election Output Fields

Field Name	Field Description	Level of Output
Family	IPv4 address family (**INET**) or IPv6 address family (**INET6**).	All levels
Instance	Name of the routing instance.	All levels
RPA	RP address.	All levels
Group ranges	Address ranges of the multicast groups mapped to this RP address.	All levels

Table 87: show pim bidirectional df-election Output Fields (continued)

Field Name	Field Description	Level of Output
Interfaces	Bidirectional PIM interfaces on this routing device. An interface can win the DF election (**Win**), lose the DF election (**Lose**), or be the RP link (**RPL**). The RP link is the interface directly connected to a subnet that contains a phantom RP address. A phantom RP address is an RP address that is not assigned to a routing device interface.	All levels **brief** displays the DF election winner only.
DF	IP address of the designated forwarder.	All levels

Sample Output

show pim bidirectional df-election

```
user@host> show pim bidirectional df-election
Instance: PIM.master Family: INET

RPA: 10.10.1.3
Group ranges: 224.1.3.0/24, 225.1.3.0/24
Interfaces:
    ge-0/0/1.0     (RPL)      DF: none
    lo0.0          (Win)      DF: 10.255.179.246
    xe-4/1/0.0     (Win)      DF: 10.10.2.1

RPA: 10.10.13.2
Group ranges: 224.1.1.0/24, 225.1.1.0/24
Interfaces:
    ge-0/0/1.0     (Lose)     DF: 10.10.1.2
    lo0.0          (Win)      DF: 10.255.179.246
    xe-4/1/0.0     (Lose)     DF: 10.10.2.2

Instance: PIM.master Family: INET6

RPA: fec0::10:10:1:3
Group ranges: ff00::/8
Interfaces:
    ge-0/0/1.0     (Lose)     DF: fe80::b2c6:9aff:fe95:86fa
    lo0.0          (Win)      DF: fe80::2a0:a50f:fc64:e661
    xe-4/1/0.0     (Win)      DF: fe80::226:88ff:fec5:3c37

RPA: fec0::10:10:13:2
Group ranges: ff00::/8
Interfaces:
    ge-0/0/1.0     (Lose)     DF: fe80::b2c6:9aff:fe95:86fa
    lo0.0          (Win)      DF: fe80::2a0:a50f:fc64:e661
    xe-4/1/0.0     (Win)      DF: fe80::226:88ff:fec5:3c37
```

show pim bidirectional df-election brief

```
user@host> show pim bidirectional df-election brief
Instance: PIM.master Family: INET

RPA: 10.10.1.3
Group ranges: 224.1.3.0/24, 225.1.3.0/24
Interfaces:
    lo0.0          (Win)      DF: 10.255.179.246
```

```
    xe-4/1/0.0    (Win)      DF: 10.10.2.1

RPA: 10.10.13.2
Group ranges: 224.1.1.0/24, 225.1.1.0/24
Interfaces:
    lo0.0         (Win)      DF: 10.255.179.246

Instance: PIM.master Family: INET6

RPA: fec0::10:10:1:3
Group ranges: ff00::/8
Interfaces:
    lo0.0         (Win)      DF: fe80::2a0:a50f:fc64:e661
    xe-4/1/0.0    (Win)      DF: fe80::226:88ff:fec5:3c37

RPA: fec0::10:10:13:2
Group ranges: ff00::/8
Interfaces:
    lo0.0         (Win)      DF: fe80::2a0:a50f:fc64:e661
    xe-4/1/0.0    (Win)      DF: fe80::226:88ff:fec5:3c37
```

show pim bidirectional df-election interface

Syntax show pim bidirectional df-election interface
<inet | inet6>
<instance *instance name*>
<*interface-name*>
<logical-system (all | *logical-system-name*)>

Release Information Command introduced in Junos OS Release 12.1.

Description For bidirectional PIM, display the default and the configured designated forwarder (DF) election parameters for each interface.

Options **none**—Display standard information about all interfaces.

inet | inet6—(Optional) Display DF election parameters for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display DF election parameters for a specific routing instance.

interface-name—(Optional) Display DF election parameters for a specific interface.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level view

List of Sample Output show pim bidirectional df-election interface on page 1635

Output Fields Table 88 on page 1634 describes the output fields for the **show pim bidirectional df-election interface** command. Output fields are listed in the approximate order in which they appear.

Table 88: show pim bidirectional df-election interface Output Fields

Field Name	Field Description
Instance	Name of the routing instance.
Family	IPv4 address family (**INET**) or IPv6 address family (**INET6**).
Interface	Name of the bidirectional PIM interface.
Robustnes Count	Minimum number of DF election messages that must fail to be received for DF election to fail.
Offer Period	Interval between repeated DF election messages.

Table 88: show pim bidirectional df-election interface Output Fields (continued)

Field Name	Field Description
Backoff Period	Period that the acting DF waits between receiving a better DF Offer and sending the Pass message to transfer DF responsibility.
RPA	RP address.
State	For each RP address, state of each interface with respect to the DF election: **Offer** (when the election is in progress), **Win**, or **Lose**.
DF	IP address of the designated forwarder.

Sample Output

show pim bidirectional df-election interface

```
user@host> show pim bidirectional df-election interface
Instance: PIM.master Family: INET

Interface: ge-0/0/1.0
  Robustness Count: 3
  Offer Period: 100 ms
  Backoff Period: 1000 ms

  RPA                           State  DF
  10.10.1.3                     Offer  none
  10.10.13.2                    Lose   10.10.1.2

Interface: lo0.0
  Robustness Count: 3
  Offer Period: 100 ms
  Backoff Period: 1000 ms

  RPA                           State  DF
  10.10.1.3                     Win    10.255.179.246
  10.10.13.2                    Win    10.255.179.246

Interface: xe-4/1/0.0
  Robustness Count: 3
  Offer Period: 100 ms
  Backoff Period: 1000 ms

  RPA                           State  DF
  10.10.1.3                     Win    10.10.2.1
  10.10.13.2                    Lose   10.10.2.2

Instance: PIM.master Family: INET6

Interface: ge-0/0/1.0
  Robustness Count: 3
  Offer Period: 100 ms
  Backoff Period: 1000 ms

  RPA                           State  DF
  fec0::10:10:1:3               Lose   fe80::b2c6:9aff:fe95:86fa
  fec0::10:10:13:2              Lose   fe80::b2c6:9aff:fe95:86fa
```

```
Interface: lo0.0
  Robustness Count: 3
  Offer Period: 100 ms
  Backoff Period: 1000 ms

  RPA                             State  DF
  fec0::10:10:1:3                 Win    fe80::2a0:a50f:fc64:e661
  fec0::10:10:13:2                Win    fe80::2a0:a50f:fc64:e661

Interface: xe-4/1/0.0
  Robustness Count: 3
  Offer Period: 100 ms
  Backoff Period: 1000 ms

  RPA                             State  DF
  fec0::10:10:1:3                 Win    fe80::226:88ff:fec5:3c37
  fec0::10:10:13:2                Win    fe80::226:88ff:fec5:3c37
```

show pim bootstrap

List of Syntax
Syntax on page 1637
Syntax (EX Series Switch and the QFX Series) on page 1637

Syntax

show pim bootstrap
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

show pim bootstrap
<instance *instance-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
instance option introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

For sparse mode only, display information about Protocol Independent Multicast (PIM) bootstrap routers.

Options

none—Display PIM bootstrap router information for all routing instances.

instance *instance-name*—(Optional) Display information about bootstrap routers for a specific PIM-enabled routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

List of Sample Output
show pim bootstrap on page 1638
show pim bootstrap instance on page 1638

Output Fields

Table 89 on page 1637 describes the output fields for the **show pim bootstrap** command. Output fields are listed in the approximate order in which they appear.

Table 89: show pim bootstrap Output Fields

Field Name	Field Description
Instance	Name of the routing instance.
BSR	Bootstrap router.
Pri	Priority of the routing device as elected to be the bootstrap router.

Table 89: show pim bootstrap Output Fields (continued)

Field Name	Field Description
Local address	Local routing device address.
Pri	Local routing device address priority to be elected as the bootstrap router.
State	Local routing device election state: **Candidate**, **Elected**, or **Ineligible**.
Timeout	How long until the local routing device declares the bootstrap router to be unreachable, in seconds.

Sample Output

show pim bootstrap

```
user@host> show pim bootstrap
Instance: PIM.master

BSR                     Pri Local address           Pri State      Timeout
None                      0 10.255.71.46              0 InEligible       0
2001:db8:1:1:1:0:aff:785c 34 2001:db8:1:1:1:0:aff:7c12   0 InEligible       0
```

show pim bootstrap instance

```
user@host> show pim bootstrap instance VPN-A
Instance: PIM.VPN-A

BSR             Pri Local address   Pri State      Timeout
None              0 192.168.196.105   0 InEligible       0
```

show pim interfaces

List of Syntax
Syntax on page 1639
Syntax (EX Series Switch and the QFX Series) on page 1639

Syntax
show pim interfaces
<inet | inet6>
<instance (*instance-name* | all)>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show pim interfaces
<inet | inet6>
<instance (*instance-name* | all)>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Commmand introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional PIM added in Junos OS Release 12.1.
Support for the **instance all** option added in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display information about the interfaces on which Protocol Independent Multicast (PIM) is configured.

Options
none—Display interface information for all family addresses for the main instance.

inet | inet6—(Optional) Display interface information for IPv4 or IPv6 family addresses, respectively.

instance (*instance-name* | all)—(Optional) Display information about interfaces for a specific PIM-enabled routing instance or for all routing instances.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show pim interfaces on page 1641

Output Fields
Table 90 on page 1639 describes the output fields for the **show pim interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 90: show pim interfaces Output Fields

Field Name	Field Description
Instance	Name of the routing instance.

Table 90: show pim interfaces Output Fields (continued)

Field Name	Field Description
Name	Interface name.
State	State of the interface. The state also is displayed in the **show interfaces** command.
Mode	PIM mode running on the interface: • **B**—In bidirectional mode, multicast groups are carried across the network over bidirectional shared trees. This type of tree minimizes PIM routing state, which is especially important in networks with numerous and dispersed senders and receivers. • **S**—In sparse mode, routing devices must join and leave multicast groups explicitly. Upstream routing devices do not forward multicast traffic to this routing device unless this device has sent an explicit request (using a join message) to receive multicast traffic. • **Dense**—Unlike sparse mode, where data is forwarded only to routing devices sending an explicit request, dense mode implements a flood-and-prune mechanism, similar to DVMRP (the first multicast protocol used to support the multicast backbone). (Not supported on QFX Series.) • **Sparse-Dense**—Sparse-dense mode allows the interface to operate on a per-group basis in either sparse or dense mode. A group specified as **dense** is not mapped to a rendezvous point (RP). Instead, data packets destined for that group are forwarded using PIM-Dense Mode (PIM-DM) rules. A group specified as **sparse** is mapped to an RP, and data packets are forwarded using PIM-Sparse Mode (PIM-SM) rules. When sparse-dense mode is configured, the output includes both **S** and **D**. When bidirectional-sparse mode is configured, the output includes **S** and **B**. When bidirectional-sparse-dense mode is configured, the output includes **B**, **S**, and **D**.
IP	Version number of the address family on the interface: **4** (IPv4) or **6** (IPv6).
V	PIM version running on the interface: **1** or **2**.
State	State of PIM on the interface: • **Active**—Bidirectional mode is enabled on the interface and on all PIM neighbors. • **DR**—Designated router. • **NotCap**—Bidirectional mode is not enabled on the interface. This can happen when bidirectional PIM is not configured locally, when one of the neighbors is not configured for bidirectional PIM, or when one of the neighbors has not implemented the bidirectional PIM protocol. • **NotDR**—Not the designated router. • **P2P**—Point to point.
NbrCnt	Number of neighbors that have been seen on the interface.
JoinCnt(sg)	Number of (s,g) join messages that have been seen on the interface.
JointCnt(*g)	Number of (*,g) join messages that have been seen on the interface.
DR address	Address of the designated router.

Sample Output

show pim interfaces

```
user@host> show pim interfaces
Stat = Status, V = Version, NbrCnt = Neighbor Count,
S = Sparse, D = Dense, B = Bidirectional,
DR = Designated Router, P2P = Point-to-point link,
Active = Bidirectional is active, NotCap = Not Bidirectional Capable

Name               Stat Mode IP V State        NbrCnt JoinCnt(sg/*g) DR address
ge-0/3/0.0         Up   S     4 2 NotDR,NotCap      1 0/0            40.0.0.3
ge-0/3/3.50        Up   S     4 2 DR,NotCap         1 9901/100       50.0.0.2
ge-0/3/3.51        Up   S     4 2 DR,NotCap         1 0/0            51.0.0.2
pe-1/2/0.32769     Up   S     4 2 P2P,NotCap        0 0/0
```

show pim join

List of Syntax

Syntax on page 1642
Syntax (EX Series Switch and the QFX Series) on page 1642

Syntax

show pim join
<brief | detail | extensive | summary>
<bidirectional | dense | sparse>
<downstream-count>
<exact>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<*range*>
<rp *ip-address/prefix* | source *ip-address/prefix*>
<sg | star-g>

Syntax (EX Series Switch and the QFX Series)

show pim join
<brief | detail | extensive | summary>
<dense | sparse>
<exact>
<inet | inet6>
<instance *instance-name*>
<*range*>
<rp *ip-address/prefix* | source *ip-address/prefix*>
<sg | star-g>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
summary option introduced in Junos OS Release 9.6.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Support for bidirectional PIM added in Junos OS Release 12.1.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Multiple new filter options introduced in Junos OS Release 13.2.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
downstream-count option introduced in Junos OS Release 16.1.
Support for PIM NSR support for VXLAN added in Junos OS Release 16.2
Support for RFC 5496 (via **rpf-vector**) added in Junos OS Release 17.3R1.

Description

Display information about Protocol Independent Multicast (PIM) groups for all PIM modes.

For bidirectional PIM, display information about PIM group ranges (*,G-range) for each active bidirectional RP group range, in addition to each of the joined (*,G) routes.

Options

none—Display the standard information about PIM groups for all supported family addresses for all routing instances.

brief | detail | extensive | summary—(Optional) Display the specified level of output.

bidirectional | dense | sparse—(Optional) Display information about PIM bidirectional mode, dense mode, or sparse and source-specific multicast (SSM) mode entries.

downstream-count—(Optional) Display the downstream count instead of a list.

exact—(Optional) Display information about only the group that exactly matches the specified group address.

inet | inet6—(Optional) Display PIM group information for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display information about groups for the specified PIM-enabled routing instance only.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

range—(Optional) Address range of the group, specified as ***prefix/prefix-length***.

rp *ip-address/prefix* | source *ip-address/prefix*—(Optional) Display information about the PIM entries with a specified rendezvous point (RP) address and prefix or with a specified source address and prefix. You can omit the prefix.

sg | star-g—(Optional) Display information about PIM (S,G) or (*,G) entries.

Required Privilege Level

view

Related Documentation

- clear pim join on page 1416
- Example: Configuring Multicast-Only Fast Reroute in a PIM Domain on page 837
- Example: Configuring Bidirectional PIM on page 343
- Example: Configuring PIM State Limits on page 756

List of Sample Output

show pim join extensive (Ingress Node with Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 1654
show pim join extensive (Egress Node with Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 1655

Output Fields Table 91 on page 1644 describes the output fields for the **show pim join** command. Output fields are listed in the approximate order in which they appear.

Table 91: show pim join Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the routing instance.	**brief detail extensive summary** none
Family	Name of the address family: **inet** (IPv4) or **inet6** (IPv6).	**brief detail extensive summary** none
Route type	Type of multicast route: (S,G) or (*,G).	**summary**
Route count	Number of (S,G) routes and number of (*,G) routes.	**summary**
R	Rendezvous Point Tree.	**brief detail extensive** none
S	Sparse.	**brief detail extensive** none
W	Wildcard.	**brief detail extensive** none
Group	Group address.	**brief detail extensive** none
Bidirectional group prefix length	For bidirectional PIM, length of the IP prefix for RP group ranges.	All levels
Source	Multicast source: • ***** (wildcard value) • ***ipv4-address*** • ***ipv6-address***	**brief detail extensive** none
RP	Rendezvous point for the PIM group.	**brief detail extensive** none
Flags	PIM flags: • **bidirectional**—Bidirectional mode entry. • **dense**—Dense mode entry. • **rptree**—Entry is on the rendezvous point tree. • **sparse**—Sparse mode entry. • **spt**—Entry is on the shortest-path tree for the source. • **wildcard**—Entry is on the shared tree.	**brief detail extensive** none

Table 91: show pim join Output Fields (continued)

Field Name	Field Description	Level of Output
Upstream interface	RPF interface toward the source address for the source-specific state (S,G) or toward the rendezvous point (RP) address for the non-source-specific state (*,G). For bidirectional PIM, **RP Link** means that the interface is directly connected to a subnet that contains a phantom RP address. A pseudo multipoint LDP (M-LDP) interface appears on egress nodes in M-LDP point-to-multipoint LSPs with inband signaling.	**brief detail extensive** none
Upstream neighbor	Information about the upstream neighbor: **Direct**, **Local**, **Unknown**, or a specific IP address. For bidirectional PIM, **Direct** means that the interface is directly connected to a subnet that contains a phantom RP address. The multipoint LDP (M-LDP) root appears on egress nodes in M-LDP point-to-multipoint LSPs with inband signaling.	**extensive**
Upstream rpf-vector	Information about the upstream Reverse Path Forwarding (RPF) vector; appears in conjunction with the rpf-vector command.	**extensive**
Active upstream interface	When multicast-only fast reroute (MoFRR) is configured in a PIM domain, the upstream interface for the active path. A PIM router propagates join messages on two upstream RPF interfaces to receive multicast traffic on both links for the same join request. Preference is given to two paths that do not converge to the same immediate upstream router. PIM installs appropriate multicast routes with upstream neighbors as RPF next hops with two (primary and backup) interfaces.	**extensive**
Active upstream neighbor	On the MoFRR primary path, the IP address of the neighbor that is directly connected to the active upstream interface.	**extensive**
MoFRR Backup upstream interface	The MoFRR upstream interface that is used when the primary path fails. When the primary path fails, the backup path is upgraded to primary, and traffic is forwarded accordingly. If there are alternate paths available, a new backup path is calculated and the appropriate multicast route is updated or installed.	**extensive**
MoFRR Backup upstream neighbor	IP address of the MoFRR upstream neighbor.	extensive

Table 91: show pim join Output Fields (continued)

Field Name	Field Description	Level of Output
Upstream state	Information about the upstream interface: • **Join to RP**—Sending a join to the rendezvous point. • **Join to Source**—Sending a join to the source. • **Local RP**—Sending neither join messages nor prune messages toward the RP, because this routing device is the rendezvous point. • **Local Source**—Sending neither join messages nor prune messages toward the source, because the source is locally attached to this routing device. • **No Prune to RP**—Automatically sent to RP when SPT and RPT are on the same path. • **Prune to RP**—Sending a prune to the rendezvous point. • **Prune to Source**—Sending a prune to the source. NOTE: RP group range entries have **None** in the **Upstream state** field because RP group ranges do not trigger actual PIM join messages between routing devices.	**extensive**
Downstream neighbors	Information about downstream interfaces: • **Interface**—Interface name for the downstream neighbor. A pseudo PIM-SM interface appears for all IGMP-only interfaces. A pseudo multipoint LDP (Pseudo-MLDP) interface appears on ingress root nodes in M-LDP point-to-multipoint LSPs with inband signaling. • **Interface address**—Address of the downstream neighbor. • **State**—Information about the downstream neighbor: **join** or **prune**. • **Flags**—PIM join flags: **R (RPtree)**, **S (Sparse)**, **W (Wildcard)**, or **zero**. • **Uptime**—Time since the downstream interface joined the group. • **Time since last Join**—Time since the last join message was received from the downstream interface. • **Time since last Prune**—Time since the last prune message was received from the downstream interface. • **rpf-vector**—IP address of the RPF vector TLV .	**extensive**
Number of downstream interfaces	Total number of outgoing interfaces for each (S,G) entry.	**extensive**
Assert Timeout	Length of time between assert cycles on the downstream interface. Not displayed if the assert timer is null.	**extensive**
Keepalive timeout	Time remaining until the downstream join state is updated (in seconds). If the downstream join state is not updated before this keepalive timer reaches zero, the entry is deleted. If there is a directly connected host, **Keepalive timeout** is **Infinity**.	**extensive**
Uptime	Time since the creation of (S,G) or (*,G) state. The uptime is not refreshed every time a PIM join message is received for an existing (S,G) or (*,G) state.	**extensive**

Table 91: show pim join Output Fields (continued)

Field Name	Field Description	Level of Output
Bidirectional accepting interfaces	Interfaces on the routing device that forward bidirectional PIM traffic. The reasons for forwarding bidirectional PIM traffic are that the interface is the winner of the designated forwarder election (**DF Winner**), or the interface is the reverse path forwarding (RPF) interface toward the RP (**RPF**).	**extensive**

Sample Output

show pim join summary

```
user@host> show pim join summary
Instance: PIM.master Family: INET

Route type           Route count
(s,g)                2
(*,g)                1

Instance: PIM.master Family: INET6
```

show pim join (PIM Sparse Mode)

```
user@host> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Source: *
    RP: 10.255.14.144
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 233.252.0.1
    Source: 10.255.14.144
    Flags: sparse,spt
    Upstream interface: Local

Group: 233.252.0.1
    Source: 10.255.70.15
    Flags: sparse,spt
    Upstream interface: so-1/0/0.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

show pim join (Bidirectional PIM)

```
user@host> show pim join
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Bidirectional group prefix length: 24
    Source: *
```

```
    RP: 10.10.13.2
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0

Group: 233.252.0.2
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.1.3
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0 (RP Link)

Group: 233.252.0.3
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.13.2
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0

Group: 233.252.0.4
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.1.3
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0 (RP Link)

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

show pim join inet6

```
user@host> show pim join inet6
Instance: PIM.master Family: INET6
 R = Rendezvous Point Tree, S = Sparse, W = Wildcard

 Group: 2001:db8::e000:101
     Source: *
     RP: ::46.0.0.13
     Flags: sparse,rptree,wildcard
     Upstream interface: Local

 Group: 2001:db8::e000:101
     Source: ::1.1.1.1
     Flags: sparse
     Upstream interface: unknown (no neighbor)

 Group: 2001:db8::e800:101
     Source: ::1.1.1.1
     Flags: sparse
     Upstream interface: unknown (no neighbor)

 Group: 2001:db8::e800:101
     Source: ::1.1.1.2
     Flags: sparse
     Upstream interface: unknown (no neighbor)
```

show pim join inet6 star-g

```
user@host> show pim join inet6 star-g
Instance: PIM.master Family: INET6
 R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

```
 Group: 2001:db8::e000:101
     Source: *
     RP: ::46.0.0.13
     Flags: sparse,rptree,wildcard
     Upstream interface: Local
```

show pim join instance <instance-name>

```
user@host> show pim join instance VPN-A
Instance: PIM.VPN-A Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.2
    Source: *
    RP: 10.10.47.100
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 233.252.0.2
    Source: 192.168.195.74
    Flags: sparse,spt
    Upstream interface: at-0/3/1.0

Group: 233.252.0.2
    Source: 192.168.195.169
    Flags: sparse
    Upstream interface: so-1/0/1.0

Instance: PIM.VPN-A Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

show pim join instance <instance-name> downstream-count

```
user@host> show pim join instance VPN-A downstream-count
Instance: PIM.SML_VRF_4 Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Source: *
    RP: 10.11.11.6
    Flags: sparse,rptree,wildcard
    Upstream interface: mt-1/2/10.32813
    Number of downstream interfaces: 4

Group: 233.252.0.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: ge-0/0/3.5
    Number of downstream interfaces: 5
```

show pim join instance <instance-name> downstream-count extensive

```
user@host> show pim join instance VPN-A downstream-count extensive
Instance: PIM.SML_VRF_4 Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Source: *
    RP: 10.11.11.6
```

```
    Flags: sparse,rptree,wildcard
    Upstream interface: mt-1/2/10.32813
    Upstream neighbor: 10.2.2.7 (assert winner)
    Upstream state: Join to RP
    Uptime: 02:51:41
    Number of downstream interfaces: 4
    Number of downstream neighbors: 4

Group: 233.252.0.1
    Source: 10.1.1.1
    Flags: sparse,spt
    Upstream interface: ge-0/0/3.5
    Upstream neighbor: 10.1.1.17
    Upstream state: Join to Source, Prune to RP
    Keepalive timeout: 0
    Uptime: 02:51:42
    Number of downstream interfaces: 5
    Number of downstream neighbors: 7
```

show pim join detail

```
user@host> show pim join detail
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Source: *
    RP: 10.255.14.144
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 233.252.0.1
    Source: 10.255.14.144
    Flags: sparse,spt
    Upstream interface: Local

Group: 233.252.0.1
    Source: 10.255.70.15
    Flags: sparse,spt
    Upstream interface: so-1/0/0.0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

show pim join extensive (PIM Resolve TLV for Multicast in Seamless MPLS)

```
user@host> show pim join extensive
Group: 228.26.1.5
    Source: 60.0.0.101
    Flags: sparse,spt
    Upstream interface: ge-5/0/0.1
    Upstream neighbor: 10.100.1.13
    Upstream state: Join to Source
Upstream rpf-vector: 10.100.20.1
    Keepalive timeout: 178
    Uptime: 17:44:38
    Downstream neighbors:
        Interface: xe-2/0/3.1
            203.21.2.190 State: Join Flags: S Timeout: 156
            Uptime: 17:44:38 Time since last Join: 00:00:54
```

```
                rpf-vector: 10.100.20.1
            Interface: xe-2/0/2.1
                203.21.1.190 State: Join Flags: S Timeout: 156
                Uptime: 17:44:38 Time since last Join: 00:00:54
                rpf-vector: 10.100.20.2
        Number of downstream interfaces: 2
        Number of downstream neighbors: 2
```

show pim join extensive (PIM Sparse Mode)

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Source: *
    RP: 10.255.14.144
    Flags: sparse,rptree,wildcard
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local RP
    Uptime: 00:03:49
    Downstream neighbors:
        Interface: so-1/0/0.0
            10.111.10.2 State: Join Flags: SRW Timeout: 174
            Uptime: 00:03:49 Time since last Join: 00:01:49
        Interface: mt-1/1/0.32768
            10.10.47.100 State: Join Flags: SRW  Timeout: Infinity
            Uptime: 00:03:49 Time since last Join: 00:01:49
    Number of downstream interfaces: 2

Group: 233.252.0.1
    Source: 10.255.14.144
    Flags: sparse,spt
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local Source, Local RP
    Keepalive timeout: 344
    Uptime: 00:03:49
    Downstream neighbors:
        Interface: so-1/0/0.0
            10.111.10.2 State: Join Flags: S Timeout: 174
            Uptime: 00:03:49 Time since last Prune: 00:01:49
        Interface: mt-1/1/0.32768
            10.10.47.100 State: Join Flags: S   Timeout: Infinity
            Uptime: 00:03:49 Time since last Prune: 00:01:49
    Number of downstream interfaces: 2

Group: 233.252.0.1
    Source: 10.255.70.15
    Flags: sparse,spt
    Upstream interface: so-1/0/0.0
    Upstream neighbor: 10.111.10.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 344
    Uptime: 00:03:49
    Downstream neighbors:
        Interface: Pseudo-GMP
            fe-0/0/0.0 fe-0/0/1.0 fe-0/0/3.0
        Interface: so-1/0/0.0 (pruned)
            10.111.10.2 State: Prune Flags: SR Timeout: 174
```

```
            Uptime: 00:03:49 Time since last Prune: 00:01:49
        Interface: mt-1/1/0.32768
            10.10.47.100 State: Join Flags: S   Timeout: Infinity
            Uptime: 00:03:49 Time since last Prune: 00:01:49
    Number of downstream interfaces: 3

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

show pim join extensive (Bidirectional PIM)

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.0
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.13.2
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0
    Upstream neighbor: 10.10.1.2
    Upstream state: None
    Uptime: 00:03:49
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)
    Number of downstream interfaces: 0

Group: 233.252.0.1
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.13.2
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0
    Upstream neighbor: 10.10.1.2
    Upstream state: None
    Uptime: 00:03:49
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)
     Downstream neighbors:
         Interface: lt-1/0/10.24
             10.0.24.4 State: Join   RW  Timeout: 185
         Interface: lt-1/0/10.23
             10.0.23.3 State: Join   RW  Timeout: 184
     Number of downstream interfaces: 2

Group: 233.252.0.2
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.1.3
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0 (RP Link)
    Upstream neighbor: Direct
    Upstream state: Local RP
    Uptime: 00:03:49
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)
        Interface: xe-4/1/0.0    (DF Winner)
```

```
    Number of downstream interfaces: 0

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

show pim join extensive (Bidirectional PIM with a Directly Connected Phantom RP)

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.0
    Bidirectional group prefix length: 24
    Source: *
    RP: 10.10.1.3
    Flags: bidirectional,rptree,wildcard
    Upstream interface: ge-0/0/1.0 (RP Link)
    Upstream neighbor: Direct
    Upstream state: Local RP
    Uptime: 00:03:49
    Bidirectional accepting interfaces:
        Interface: ge-0/0/1.0    (RPF)
        Interface: lo0.0         (DF Winner)
        Interface: xe-4/1/0.0    (DF Winner)
    Number of downstream interfaces: 0
```

show pim join instance <instance-name> extensive

```
user@host> show pim join instance VPN-A extensive
Instance: PIM.VPN-A Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.2
    Source: *
    RP: 10.10.47.100
    Flags: sparse,rptree,wildcard
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local RP
    Uptime: 00:03:49
    Downstream neighbors:
        Interface: mt-1/1/0.32768
            10.10.47.101 State: Join Flags: SRW Timeout: 156
            Uptime: 00:03:49 Time since last Join: 00:01:49
    Number of downstream interfaces: 1

Group: 233.252.0.2
    Source: 192.168.195.74
    Flags: sparse,spt
    Upstream interface: at-0/3/1.0
    Upstream neighbor: 10.111.30.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 156
    Uptime: 00:14:52

Group: 233.252.0.2
    Source: 192.168.195.169
    Flags: sparse
    Upstream interface: so-1/0/1.0
    Upstream neighbor: 10.111.20.2
```

```
    Upstream state: Local RP, Join to Source
    Keepalive timeout: 156
    Uptime: 00:14:52
```

show pim join extensive (Ingress Node with Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.1
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/3/1.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 11:27:55
    Downstream neighbors:
        Interface: Pseudo-MLDP
        Interface: lt-1/2/0.25
            10.2.5.2 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:27:55 Time since last Join: 11:27:55

Group: 233.252.0.2
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/3/1.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 11:27:41
    Downstream neighbors:
        Interface: Pseudo-MLDP

Group: 233.252.0.3
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream interface: fe-1/3/1.0
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 11:27:41
    Downstream neighbors:
        Interface: Pseudo-MLDP

Group: 233.252.0.22
    Source: 10.2.7.7
    Flags: sparse,spt
    Upstream interface: lt-1/2/0.27
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 11:27:25
    Downstream neighbors:
        Interface: Pseudo-MLDP

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 2001:db8::1:2
```

```
    Source: 2001:db8::1:2:7:7
    Flags: sparse,spt
    Upstream interface: lt-1/2/0.27
    Upstream neighbor: Direct
    Upstream state: Local Source
    Keepalive timeout:
    Uptime: 11:27:26
    Downstream neighbors:
        Interface: Pseudo-MLDP
```

show pim join extensive (Egress Node with Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 233.252.0.0
    Source: *
    RP: 10.1.1.1
    Flags: sparse,rptree,wildcard
    Upstream interface: Local
    Upstream neighbor: Local
    Upstream state: Local RP
    Uptime: 11:31:33
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: SRW  Timeout: Infinity
            Uptime: 11:31:33 Time since last Join: 11:31:32

Group: 233.252.0.1
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 11:31:32
    Downstream neighbors:
        Interface: so-0/1/3.0
            192.168.92.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:31:30 Time since last Join: 11:31:30
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:31:32 Time since last Join: 11:31:32

Group: 233.252.0.2
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 11:31:32
    Downstream neighbors:
        Interface: so-0/1/3.0
            192.168.92.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:31:30 Time since last Join: 11:31:30
    Downstream neighbors:
```

```
        Interface: lt-1/2/0.14
            10.1.4.4 State: Join Flags: S Timeout: 177
            Uptime: 11:30:33 Time since last Join: 00:00:33
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:31:32 Time since last Join: 11:31:32

Group: 233.252.0.3
    Source: 192.168.219.11
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 11:31:32
    Downstream neighbors:
        Interface: fe-1/3/0.0
            192.168.209.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:31:32 Time since last Join: 11:31:32

Group: 233.252.0.22
    Source: 10.2.7.7
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 11:31:30
    Downstream neighbors:
        Interface: so-0/1/3.0
            192.168.92.9 State: Join Flags: S   Timeout: Infinity
            Uptime: 11:31:30 Time since last Join: 11:31:30

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 2001:db8::1:2
    Source: 2001:db8::1:2:7:7
    Flags: sparse,spt
    Upstream protocol: MLDP
    Upstream interface: Pseudo MLDP
    Upstream neighbor: MLDP LSP root <10.1.1.2>
    Upstream state: Join to Source
    Keepalive timeout:
    Uptime: 11:31:32
    Downstream neighbors:
        Interface: fe-1/3/0.0
           2001:db8::21f:12ff:fea5:c4db State: Join Flags: S   Timeout: Infinity

            Uptime: 11:31:32 Time since last Join: 11:31:32
```

show pim neighbors

List of Syntax
Syntax on page 1657
Syntax (EX Series Switch and the QFX Series) on page 1657

Syntax
show pim neighbors
<brief | detail>
<inet | inet6>
<instance (*instance-name* | all)>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)
show pim neighbors
<brief | detail>
<inet | inet6>
<instance (*instance-name* | all)>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional PIM added in Junos OS Release 12.1.
Support for the **instance all** option added in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.
Support for RFC 5496 (via **rpf-vector**) added in Junos OS Release 17.3R1.

Description
Display information about Protocol Independent Multicast (PIM) neighbors.

Options
none—(Same as **brief**) Display standard information about PIM neighbors for all supported family addresses for the main instance.

brief | detail—(Optional) Display the specified level of output.

inet | inet6—(Optional) Display information about PIM neighbors for IPv4 or IPv6 family addresses, respectively.

instance (*instance-name* | all)—(Optional) Display information about neighbors for the specified PIM-enabled routing instance or for all routing instances.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show pim neighbors on page 1659
show pim neighbors instance on page 1659
show pim neighbors detail on page 1659
show pim neighbors detail (With BFD) on page 1660

Output Fields Table 92 on page 1658 describes the output fields for the **show pim neighbors** command. Output fields are listed in the approximate order in which they appear.

Table 92: show pim neighbors Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the routing instance.	All levels
Interface	Interface through which the neighbor is reachable.	All levels
Neighbor addr	Address of the neighboring PIM routing device.	All levels
IP	IP version: **4** or **6**.	All levels
V	PIM version running on the neighbor: **1** or **2**.	All levels
Mode	PIM mode of the neighbor: **Sparse**, **Dense**, **SparseDense**, or **Unknown**. When the neighbor is running PIM version 2, this mode is always **Unknown**.	All levels
Option	Can be one or more of the following: • **B**—Bidirectional Capable. • **G**—Generation Identifier. • **H**—Hello Option Holdtime. • **L**—Hello Option LAN Prune Delay. • **P**—Hello Option DR Priority. • **T**—Tracking bit. • **A**—Join attribute; used in conjunction with **pim rpf-vector**.	**brief** none
Uptime	Time the neighbor has been operational since the PIM process was last initialized. Starting in Junos OS release 17.3R1, **uptime** is not reset during ISSU.The time format is as follows: ***dd:hh:mm:ss*** **ago** for less than a week and ***nwnd:hh:mm:ss*** **ago** for more than a week.	All levels
Address	Address of the neighboring PIM routing device.	**detail**
BFD	Status and operational state of the Bidirectional Forwarding Detection (BFD) protocol on the interface: **Enabled**, **Operational state is up**, or **Disabled**.	**detail**
Hello Option Holdtime	Time for which the neighbor is available, in seconds. The range of values is **0** through **65,535.**	**detail**
Hello Default Holdtime	Default holdtime and the time remaining if the **holdtime** option is not in the received hello message.	**detail**
Hello Option DR Priority	Designated router election priority. The range of values is **0** through **255**.	**detail**
Hello Option Join Attribute	Appears in conjunction with the **rpf-vector** command. The Join attribute is included in the PIM join messages of PIM routers that can receive type 1 Encoded-Source Address.	**detail**

Table 92: show pim neighbors Output Fields (continued)

Field Name	Field Description	Level of Output
Hello Option Generation ID	9-digit or 10-digit number used to tag hello messages.	**detail**
Hello Option Bi-Directional PIM supported	Neighbor can process bidirectional PIM messages.	**detail**
Hello Option LAN Prune Delay	Time to wait before the neighbor receives prune messages, in the format **delay** *nnn* **ms override** *nnnn* **ms**.	**detail**
Join Suppression supported	Neighbor is capable of join suppression.	**detail**
Rx Join	Information about joins received from the neighbor. • **Group**—Group addresses in the join message. • **Source**—Address of the source in the join message. • **Timeout**—Time for which the join is valid.	**detail**

Sample Output

show pim neighbors

```
user@host> show pim neighbors
Instance: PIM.master
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking bit
A = Hello Option Join Attribute

Instance: PIM.master
Interface  IP V Mode        Option       Uptime Neighbor addr
ae0.0      4  2             HPLGTA       19:01:24 20.0.0.13
ae1.0      4  2             HPLGTA       19:01:24 20.0.0.149
```

show pim neighbors instance

```
user@host> show pim neighbors instance VPN-A
Instance: PIM.VPN-A
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking bit

Interface           IP V Mode        Option      Uptime Neighbor addr
at-0/3/1.0           4 2             HPLG      00:07:54 10.111.30.2
mt-1/1/0.32768       4 2             HPLG      00:07:22 10.10.47.101
so-1/0/1.0           4 2             HPLG      00:07:50 10.111.20.2
```

show pim neighbors detail

```
user@host> show pim neighbors detail
```

```
Instance: PIM.master
Interface: ae1.0

Address: 20.0.0.149, IPv4, PIM v2, sg Join Count: 0, tsg Join Count: 332
        BFD: Disabled
        Hello Option Holdtime: 105 seconds 86 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 853386212
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                  Join Suppression supported
        Hello Option Join Attribute supported

    Address: 20.0.0.150, IPv4, PIM v2, Mode: SparseDense, sg Join Count: 0, tsg
Join Count: 0
        Hello Option Holdtime: 65535 seconds
        Hello Option DR Priority: 1
        Hello Option Generation ID: 358917871
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                  Join Suppression supported
             Hello Option Join Attribute supported

Interface: lo0.0

    Address: 10.255.179.246,    IPv4, PIM v2, Mode: SparseDense, sg Join Count:
0, tsg Join Count: 0
        Hello Option Holdtime: 65535 seconds
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1997462267
        Hello Option Bi-Directional PIM supported
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                       Join Suppression supported
```

show pim neighbors detail (With BFD)

```
user@host> show pim neighbors detail
Instance: PIM.master
Interface: fe-1/0/0.0
    Address: 192.168.11.1,        IPv4, PIM v2, Mode: Sparse
        Hello Option Holdtime: 65535 seconds
        Hello Option DR Priority: 1
        Hello Option Generation ID: 836607909
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms

    Address: 192.168.11.2,        IPv4, PIM v2
        BFD: Enabled, Operational state is up
        Hello Default Holdtime: 105 seconds 104 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1907549685
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms

Interface: fe-1/0/1.0
    Address: 192.168.12.1,        IPv4, PIM v2
        BFD: Disabled
        Hello Default Holdtime: 105 seconds 80 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1971554705
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
```

show pim snooping interfaces

Syntax

show pim snooping interfaces
<brief | detail>
<instance *instance-name*>
<interface *interface-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-identifier*>

Release Information

Command introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Command introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Display information about PIM snooping interfaces.

Options

none—Display detailed information.

brief | detail—(Optional) Display the specified level of output.

instance *<instance-name>*—(Optional) Display PIM snooping interface information for the specified routing instance.

interface *<interface-name>*—(Optional) Display PIM snooping information for the specified interface only.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

vlan-id *<vlan-identifier>*—(Optional) Display PIM snooping interface information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- PIM Snooping for VPLS on page 886

List of Sample Output

show pim snooping interfaces on page 1662
show pim snooping interfaces instance vpls1 on page 1662
show pim snooping interfaces interface <interface-name> on page 1663
show pim snooping interfaces vlan-id <vlan-id> on page 1663

Output Fields

Table 93 on page 1661 lists the output fields for the **show pim snooping interface** command. Output fields are listed in the approximate order in which they appear.

Table 93: show pim snooping interface Output Fields

Field Name	Field Description	Level of Output
Instance	Routing instance for PIM snooping.	All levels

Table 93: show pim snooping interface Output Fields (continued)

Field Name	Field Description	Level of Output
Learning-Domain	Learning domain for snooping.	All levels
Name	Router interfaces that are part of this learning domain.	All levels
State	State of the interface: **Up**, or **Down**.	All levels
IP-Version	Version of IP used: **4** for IPv4, or **6** for IPv6.	All levels
NbrCnt	Number of neighboring routers connected through the specified interface.	All levels
DR address	IP address of the designated router.	All levels

Sample Output

show pim snooping interfaces

```
user@host> show pim snooping interfaces
Instance: vpls1
Learning-Domain: vlan-id 10
Name State IP-Version NbrCnt
ge-1/3/1.10 Up 4 1
ge-1/3/3.10 Up 4 1
ge-1/3/5.10 Up 4 1
ge-1/3/7.10 Up 4 1
DR address: 192.0.2.5
DR flooding is ON

Learning-Domain: vlan-id 20
Name State IP-Version NbrCnt
ge-1/3/1.20 Up 4 1
ge-1/3/3.20 Up 4 1
ge-1/3/5.20 Up 4 1
ge-1/3/7.20 Up 4 1
DR address: 192.0.2.6
DR flooding is ON
```

show pim snooping interfaces instance vpls1

```
user@host> show pim snooping interfaces instance vpls1
Instance: vpls1

Learning-Domain: vlan-id 10
Name State IP-Version NbrCnt
ge-1/3/1.10 Up 4 1
ge-1/3/3.10 Up 4 1
ge-1/3/5.10 Up 4 1
ge-1/3/7.10 Up 4 1
DR address: 192.0.2.5
DR flooding is ON

Learning-Domain: vlan-id 20
Name State IP-Version NbrCnt
```

```
ge-1/3/1.20 Up 4 1
ge-1/3/3.20 Up 4 1
ge-1/3/5.20 Up 4 1
ge-1/3/7.20 Up 4 1
DR address: 192.0.2.6
DR flooding is ON
```

show pim snooping interfaces interface <interface-name>

```
user@host> show pim snooping interfaces interface ge-1/3/1.10
Instance: vpls1
Learning-Domain: vlan-id 10

Name State IP-Version NbrCnt
ge-1/3/1.10 Up 4 1
DR address: 192.0.2.5
DR flooding is ON

Learning-Domain: vlan-id 20
DR address: 192.0.2.6
DR flooding is ON
```

show pim snooping interfaces vlan-id <vlan-id>

```
user@host> show pim snooping interfaces vlan-id 10
Instance: vpls1
Learning-Domain: vlan-id 10

Name State IP-Version NbrCnt
ge-1/3/1.10 Up 4 1
ge-1/3/3.10 Up 4 1
ge-1/3/5.10 Up 4 1
ge-1/3/7.10 Up 4 1
DR address: 192.0.2.5
DR flooding is ON
```

show pim snooping join

Syntax

show pim snooping join
<brief | detail | extensive>
<instance *instance-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-id*>

Release Information

Command introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Command introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Display information about Protocol Independent Multicast (PIM) snooping joins.

Options

none—Display detailed information.

brief | detail | extensive—(Optional) Display the specified level of output.

instance *instance-name*—(Optional) Display PIM snooping join information for the specified routing instance.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

vlan-id *vlan-identifier*—(Optional) Display PIM snooping join information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- PIM Snooping for VPLS on page 886

List of Sample Output

show pim snooping join on page 1666
show pim snooping join extensive on page 1666
show pim snooping join instance on page 1666
show pim snooping join vlan-id on page 1667

Output Fields

Table 94 on page 1664 lists the output fields for the **show pim snooping join** command. Output fields are listed in the approximate order in which they appear.

Table 94: show pim snooping join Output Fields

Field Name	Field Description	Level of Output
Instance	Routing instance for PIM snooping.	All levels
Learning-Domain	Learning domain for PIM snooping.	All levels
Group	Multicast group address.	All levels

Table 94: show pim snooping join Output Fields (continued)

Field Name	Field Description	Level of Output
Source	Multicast source address: • * (wildcard value) • *<ipv4-address>* • *<ipv6-address>*	All levels
Flags	PIM flags: • **bidirectional**—Bidirectional mode entry. • **dense**—Dense mode entry. • **rptree**—Entry is on the rendezvous point tree. • **sparse**—Sparse mode entry. • **spt**—Entry is on the shortest-path tree for the source. • **wildcard**—Entry is on the shared tree.	All levels
Upstream state	Information about the upstream interface: • **Join to RP**—Sending a join to the rendezvous point. • **Join to Source**—Sending a join to the source. • **Local RP**—Sending neither join messages nor prune messages toward the RP, because this router is the rendezvous point. • **Local Source**—Sending neither join messages nor prune messages toward the source, because the source is locally attached to this routing device. • **Prune to RP**—Sending a prune to the rendezvous point. • **Prune to Source**—Sending a prune to the source. NOTE: RP group range entries have **None** in the **Upstream state** field because RP group ranges do not trigger actual PIM join messages between routers.	All levels
Upstream neighbor	Information about the upstream neighbor: **Direct**, **Local**, **Unknown**, or a specific IP address. For bidirectional PIM, **Direct** means that the interface is directly connected to a subnet that contains a phantom RP address.	All levels
Upstream port	RPF interface toward the source address for the source-specific state (S,G) or toward the rendezvous point (RP) address for the non-source-specific state (*,G). For bidirectional PIM, **RP Link** means that the interface is directly connected to a subnet that contains a phantom RP address.	All levels
Downstream port	Information about downstream interfaces.	**extensive**
Downstream neighbors	Address of the downstream neighbor.	**extensive**
Timeout	Time remaining until the downstream join state is updated (in seconds).	**extensive**

Sample Output

show pim snooping join

```
user@host> show pim snooping join
Instance: vpls1

Learning-Domain: vlan-id 10
Group: 198.51.100.2
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 192.0.2.4, port: ge-1/3/5.10


Learning-Domain: vlan-id 20
Group: 198.51.100.3
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 203.0.113.4, port: ge-1/3/5.20
```

show pim snooping join extensive

```
user@host> show pim snooping join extensive
Instance: vpls1
Learning-Domain: vlan-id 10

Group: 198.51.100.2
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 192.0.2.4, port: ge-1/3/5.10
Downstream port: ge-1/3/1.10
Downstream neighbors:
192.0.2.2 State: Join Flags: SRW Timeout: 166

Learning-Domain: vlan-id 20
Group: 198.51.100.3
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 203.0.113.4, port: ge-1/3/5.20
Downstream port: ge-1/3/3.20
Downstream neighbors:
203.0.113.3 State: Join Flags: SRW Timeout: 168
```

show pim snooping join instance

```
user@host> show pim snooping join instance vpls1
Instance: vpls1

Learning-Domain: vlan-id 10
Group: 198.51.100.2
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 192.0.2.4, port: ge-1/3/5.10
```

```
Learning-Domain: vlan-id 20
Group: 198.51.100.3
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 203.0.113.4, port: ge-1/3/5.20
```

show pim snooping join vlan-id

```
user@host> show pim snooping join vlan-id 10
Instance: vpls1
Learning-Domain: vlan-id 10
Group: 198.51.100.2
Source: *
Flags: sparse,rptree,wildcard
Upstream state: None
Upstream neighbor: 192.0.2.4, port: ge-1/3/5.10
```

show pim snooping neighbors

Syntax

show pim snooping neighbors
<brief | detail>
<instance *instance-name*>
<interface *interface-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-identifier*>

Release Information

Command introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Command introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Display information about Protocol Independent Multicast (PIM) snooping neighbors.

Options

none—Display detailed information.

brief | detail—(Optional) Display the specified level of output.

instance *instance-name*—(Optional) Display PIM snooping neighbor information for the specified routing instance.

interface *interface-name*—(Optional) Display information for the specified PIM snooping neighbor interface.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

vlan-id *vlan-identifier*—(Optional) Display PIM snooping neighbor information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- Configuring Interface Priority for PIM Designated Router Selection on page 302
- Modifying the PIM Hello Interval on page 190
- PIM Snooping for VPLS on page 886
- **show pim neighbors on page 1657**

List of Sample Output

show pim snooping neighbors on page 1669
show pim snooping neighbors detail on page 1670
show pim snooping neighbors instance on page 1671
show pim snooping neighbors interface on page 1671
show pim snooping neighbors vlan-id on page 1672

Output Fields

Table 95 on page 1669 lists the output fields for the **show pim snooping neighbors** command. Output fields are listed in the approximate order in which they appear.

Table 95: show pim snooping neighbors Output Fields

Field Name	Field Description	Level of Output
Instance	Routing instance for PIM snooping.	All levels
Learning-Domain	Learning domain for PIM snooping.	All levels
Interface	Router interface for which PIM snooping neighbor details are displayed.	All levels
Option	PIM snooping options available on the specified interface: • H = Hello Option Holdtime • P = Hello Option DR Priority • L = Hello Option LAN Prune Delay • G = Generation Identifier • T = Tracking Bit	All levels
Uptime	Time the neighbor has been operational since the PIM process was last initialized, in the format ***dd*:*hh*:*mm*:*ss* ago** for less than a week and ***nwnd*:*hh*:*mm*:*ss* ago** for more than a week.	All levels
Neighbor addr	IP address of the PIM snooping neighbor connected through the specified interface.	All levels
Address	IP address of the specified router interface.	All levels
Hello Option Holdtime	Time for which the neighbor is available, in seconds. The range of values is **0** through **65,535**.	**detail**
Hello Option DR Priority	Designated router election priority. The range of values is **0** through **4294967295**. NOTE: By default, every PIM interface has an equal probability (priority 1) of being selected as the DR.	**detail**
Hello Option Generation ID	9-digit or 10-digit number used to tag hello messages.	**detail**
Hello Option LAN Prune Delay	Time to wait before the neighbor receives prune messages, in the format **delay *nnn* ms override *nnnn* ms**.	**detail**

Sample Output

show pim snooping neighbors

```
user@host> show pim snooping neighbors
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking Bit

Instance: vpls1
Learning-Domain: vlan-id 10

Interface Option Uptime Neighbor addr
ge-1/3/1.10 HPLGT 00:43:33 192.0.2.2
```

```
ge-1/3/3.10 HPLGT 00:43:33 192.0.2.3
ge-1/3/5.10 HPLGT 00:43:33 192.0.2.4
ge-1/3/7.10 HPLGT 00:43:33 192.0.2.5

Learning-Domain: vlan-id 20

Interface Option Uptime Neighbor addr
ge-1/3/1.20 HPLGT 00:43:33 192.0.2.12
ge-1/3/3.20 HPLGT 00:43:33 192.0.2.13
ge-1/3/5.20 HPLGT 00:43:33 192.0.2.14
ge-1/3/7.20 HPLGT 00:43:33 192.0.2.15
```

show pim snooping neighbors detail

```
user@host> show pim snooping neighbors detail
Instance: vpls1
Learning-Domain: vlan-id 10

Interface: ge-1/3/1.10
Address: 192.0.2.2
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 83 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 830908833
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported

Interface: ge-1/3/3.10
Address: 192.0.2.3
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 97 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 2056520742
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported

Interface: ge-1/3/5.10
Address: 192.0.2.4
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 81 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 1152066227
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported

Interface: ge-1/3/7.10
Address: 192.0.2.5
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 96 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 1113200338
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported
Learning-Domain: vlan-id 20

Interface: ge-1/3/1.20
Address: 192.0.2.12
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 81 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 963205167
```

```
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported

Interface: ge-1/3/3.20
Address: 192.0.2.13
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 104 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 166921538
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported

Interface: ge-1/3/5.20
Address: 192.0.2.14
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 88 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 789422835
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported

Interface: ge-1/3/7.20
Address: 192.0.2.15
Uptime: 00:44:51
Hello Option Holdtime: 105 seconds 88 remaining
Hello Option DR Priority: 1
Hello Option Generation ID: 1563649680
Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
Tracking is supported
```

show pim snooping neighbors instance

```
user@host> show pim snooping neighbors instance vpls1
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking Bit

Instance: vpls1
Learning-Domain: vlan-id 10

Interface Option Uptime Neighbor addr
ge-1/3/1.10 HPLGT 00:46:03 192.0.2.2
ge-1/3/3.10 HPLGT 00:46:03 192.0.2.3
ge-1/3/5.10 HPLGT 00:46:03 192.0.2.4
ge-1/3/7.10 HPLGT 00:46:03 192.0.2.5


Learning-Domain: vlan-id 20

Interface Option Uptime Neighbor addr
ge-1/3/1.20 HPLGT 00:46:03 192.0.2.12
ge-1/3/3.20 HPLGT 00:46:03 192.0.2.13
ge-1/3/5.20 HPLGT 00:46:03 192.0.2.14
ge-1/3/7.20 HPLGT 00:46:03 192.0.2.15
```

show pim snooping neighbors interface

```
user@host> show pim snooping neighbors interface ge-1/3/1.20
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
```

```
P = Hello Option DR Priority, T = Tracking Bit

Instance: vpls1
Learning-Domain: vlan-id 10
Learning-Domain: vlan-id 20

Interface Option Uptime Neighbor addr
ge-1/3/1.20 HPLGT 00:48:04 192.0.2.12
```

show pim snooping neighbors vlan-id

```
user@host> show pim snooping neighbors vlan-id 10
B = Bidirectional Capable, G = Generation Identifier,
H = Hello Option Holdtime, L = Hello Option LAN Prune Delay,
P = Hello Option DR Priority, T = Tracking Bit

Instance: vpls1
Learning-Domain: vlan-id 10

Interface Option Uptime Neighbor addr
ge-1/3/1.10 HPLGT 00:49:12 192.0.2.2
ge-1/3/3.10 HPLGT 00:49:12 192.0.2.3
ge-1/3/5.10 HPLGT 00:49:12 192.0.2.4
ge-1/3/7.10 HPLGT 00:49:12 192.0.2.5
```

show pim snooping statistics

Syntax

show pim snooping statistics
<instance *instance-name*>
<interface *interface-name*>
<logical-system *logical-system-name*>
<vlan-id *vlan-id*>

Release Information

Command introduced in Junos OS Release 12.3 for MX Series 3D Universal Edge devices.
Command introduced in Junos OS Release 13.2 for M Series Multiservice Edge devices.

Description

Display Protocol Independent Multicast (PIM) snooping statistics.

Options

none—Display PIM statistics.

instance *instance-name*—(Optional) Display statistics for a specific routing instance enabled by Protocol Independent Multicast (PIM) snooping.

interface *interface-name*—(Optional) Display statistics about the specified interface for PIM snooping.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

vlan-id *vlan-identifier*—(Optional) Display PIM snooping statistics information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- PIM Snooping for VPLS on page 886
- clear pim snooping statistics on page 1424

List of Sample Output

show pim snooping statistics on page 1674
show pim snooping statistics instance on page 1675
show pim snooping statistics interface on page 1676
show pim snooping statistics vlan-id on page 1676

Output Fields

Table 96 on page 1673 lists the output fields for the **show pim snooping statistics** command. Output fields are listed in the approximate order in which they appear.

Table 96: show pim snooping statistics Output Fields

Field Name	Field Description	Level of Output
Instance	Routing instance for PIM snooping.	All levels
Learning-Domain	Learning domain for PIM snooping.	All levels

Table 96: show pim snooping statistics Output Fields (continued)

Field Name	Field Description	Level of Output
Tx J/P messages	Total number of transmitted join/prune packets.	All levels
RX J/P messages	Total number of received join/prune packets.	All levels
Rx J/P messages -- seen	Number of join/prune packets seen but not received on the upstream interface.	All levels
Rx J/P messages -- received	Number of join/prune packets received on the downstream interface.	All levels
Rx Hello messages	Total number of received hello packets.	All levels
Rx Version Unknown	Number of packets received with an unknown version number.	All levels
Rx Neighbor Unknown	Number of packets received from an unknown neighbor.	All levels
Rx Upstream Neighbor Unknown	Number of packets received with unknown upstream neighbor information.	All levels
Rx Bad Length	Number of packets received containing incorrect length information.	All levels
Rx J/P Busy Drop	Number of join/prune packets dropped while the router is busy.	All levels
Rx J/P Group Aggregate 0	Number of join/prune packets received containing the aggregate group information.	All levels
Rx Malformed Packet	Number of malformed packets received.	All levels
Rx No PIM Interface	Number of packets received without the interface information.	All levels
Rx No Upstream Neighbor	Number of packets received without upstream neighbor information.	All levels
Rx Unknown Hello Option	Number of hello packets received with unknown options.	All levels

Sample Output

show pim snooping statistics

```
user@host> show pim snooping statistics
Instance: vpls1
Learning-Domain: vlan-id 10

Tx J/P messages 0
RX J/P messages 8
```

```
Rx J/P messages -- seen 0
Rx J/P messages -- received 8
Rx Hello messages 37
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0
Rx No PIM Interface 0
Rx No Upstream Neighbor 0
Rx Bad Length 0
Rx Neighbor Unknown 0
Rx Unknown Hello Option 0
Rx Malformed Packet 0


Learning-Domain: vlan-id 20

Tx J/P messages 0
RX J/P messages 2
Rx J/P messages -- seen 0
Rx J/P messages -- received 2
Rx Hello messages 39
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0
Rx No PIM Interface 0
Rx No Upstream Neighbor 0
Rx Bad Length 0
Rx Neighbor Unknown 0
Rx Unknown Hello Option 0
Rx Malformed Packet 0
```

show pim snooping statistics instance

```
user@host> show pim snooping statistics instance vpls1
Instance: vpls1
Learning-Domain: vlan-id 10

Tx J/P messages 0
RX J/P messages 9
Rx J/P messages -- seen 0
Rx J/P messages -- received 9
Rx Hello messages 45
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0
Rx No PIM Interface 0
Rx No Upstream Neighbor 0
Rx Bad Length 0
Rx Neighbor Unknown 0
```

```
Rx Unknown Hello Option 0
Rx Malformed Packet 0


Learning-Domain: vlan-id 20

Tx J/P messages 0
RX J/P messages 3
Rx J/P messages -- seen 0
Rx J/P messages -- received 3
Rx Hello messages 47
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0
Rx No PIM Interface 0
Rx No Upstream Neighbor 0
Rx Bad Length 0
Rx Neighbor Unknown 0
Rx Unknown Hello Option 0
Rx Malformed Packet 0
```

show pim snooping statistics interface

```
user@host> show pim snooping statistics interface ge-1/3/1.20
Instance: vpls1
Learning-Domain: vlan-id 10
Learning-Domain: vlan-id 20

PIM Interface statistics for ge-1/3/1.20
Tx J/P messages 0
RX J/P messages 0
Rx J/P messages -- seen 0
Rx J/P messages -- received 0
Rx Hello messages 13
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0
```

show pim snooping statistics vlan-id

```
user@host> show pim snooping statistics vlan-id 10
Instance: vpls1
Learning-Domain: vlan-id 10

Tx J/P messages 0
RX J/P messages 11
Rx J/P messages -- seen 0
Rx J/P messages -- received 11
Rx Hello messages 64
Rx Version Unknown 0
Rx Neighbor Unknown 0
Rx Upstream Neighbor Unknown 0
```

```
Rx Bad Length 0
Rx J/P Busy Drop 0
Rx J/P Group Aggregate 0
Rx Malformed Packet 0
Rx No PIM Interface 0
Rx No Upstream Neighbor 0
Rx Bad Length 0
Rx Neighbor Unknown 0
```

show pim rps

List of Syntax

Syntax on page 1678
Syntax (EX Series Switch and the QFX Series) on page 1678

Syntax

show pim rps
<brief | detail | extensive>
<*group-address*>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switch and the QFX Series)

show pim rps
<brief | detail | extensive>
<*group-address*>
<inet | inet6>
<instance *instance-name*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional PIM added in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display information about Protocol Independent Multicast (PIM) rendezvous points (RPs).

Options

none—Display standard information about PIM RPs for all groups and family addresses for all routing instances.

brief | detail | extensive—(Optional) Display the specified level of output.

group-address—(Optional) Display the RPs for a particular group. If you specify a group address, the output lists the routing device that is the RP for that group.

inet | inet6—(Optional) Display information for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display information about RPs for a specific PIM-enabled routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level

view

Related Documentation

- Example: Configuring Bidirectional PIM on page 337

List of Sample Output

Output Fields

Table 97 on page 1679 describes the output fields for the **show pim rps** command. Output fields are listed in the approximate order in which they appear.

Table 97: show pim rps Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the routing instance.	All levels
Family or Address family	Name of the address family: **inet** (IPv4) or **inet6** (IPv6).	All levels
RP address	Address of the rendezvous point.	All levels
Type	Type of RP: • **auto-rp**—Address of the RP known through the Auto-RP protocol. • **bootstrap**—Address of the RP known through the bootstrap router protocol (BSR). • **embedded**—Address of the RP known through an embedded RP (IPv6). • **static**—Address of RP known through static configuration.	**brief** none
Holdtime	How long to keep the RP active, with time remaining, in seconds.	All levels
Timeout	How long until the local routing device determines the RP to be unreachable, in seconds.	All levels
Groups	Number of groups currently using this RP.	All levels
Group prefixes	Addresses of groups that this RP can span.	**brief** none
Learned via	Address and method by which the RP was learned.	**detail extensive**

Table 97: show pim rps Output Fields (continued)

Field Name	Field Description	Level of Output
Mode	The PIM mode of the RP: bidirectional or sparse. If a sparse and bidirectional RPs are configured with the same RP address, they appear as separate entries in both formats.	All levels
Time Active	How long the RP has been active, in the format ***hh:mm:ss***.	**detail extensive**
Device Index	Index value of the order in which Junos OS finds and initializes the interface. For bidirectional RPs, the **Device Index** output field is omitted because bidirectional RPs do not require encapsulation and de-encapsulation interfaces.	**detail extensive**
Subunit	Logical unit number of the interface. For bidirectional RPs, the **Subunit** output field is omitted because bidirectional RPs do not require encapsulation and de-encapsulation interfaces.	**detail extensive**
Interface	Either the encapsulation or the de-encapsulation logical interface, depending on whether this routing device is a designated router (DR) facing an RP router, or is the local RP, respectively. For bidirectional RPs, the **Interface** output field is omitted because bidirectional RPs do not require encapsulation and de-encapsulation interfaces.	**detail extensive**
Group Ranges	Addresses of groups that this RP spans.	**detail extensive** *group-address*
Active groups using RP	Number of groups currently using this RP.	**detail extensive**
total	Total number of active groups for this RP.	**detail extensive**

Table 97: show pim rps Output Fields (continued)

Field Name	Field Description	Level of Output
Register State for RP	Current register state for each group: • **Group**—Multicast group address. • **Source**—Multicast source address for which the PIM register is sent or received, depending on whether this router is a designated router facing an RP router, or is the local RP, respectively: • **First Hop**—PIM-designated routing device that sent the Register message (the source address in the IP header). • **RP Address**—RP to which the Register message was sent (the destination address in the IP header). • **State**: On the designated router: ◦ **Send**—Sending Register messages. ◦ **Probe**—Sent a null register. If a Register-Stop message does not arrive in 5 seconds, the designated router resumes sending Register messages. ◦ **Suppress**—Received a Register-Stop message. The designated router is waiting for the timer to resume before changing to **Probe** state. • On the RP: ◦ **Receive**—Receiving Register messages.	**extensive**
Anycast-PIM rpset	If anycast RP is configured, the addresses of the RPs in the set.	**extensive**
Anycast-PIM local address used	If anycast RP is configured, the local address used by the RP.	**extensive**
Anycast-PIM Register State	If anycast RP is configured, the current register state for each group: • **Group**—Multicast group address. • **Source**—Multicast source address for which the PIM register is sent or received, depending on whether this routing device is a designated router facing an RP router, or is the local RP, respectively. • **Origin**—How the information was obtained: ◦ **DIRECT**—From a local attachment ◦ **MSDP**—From the Multicast Source Discovery Protocol (MSDP) ◦ **DR**—From the designated router	**extensive**
RP selected	For sparse mode and bidirectional mode, the identity of the RP for the specified group address.	*group-address*

Sample Output

show pim rps

```
user@host> show pim rps
Instance: PIM.master

Address-family INET
RP address      Type       Mode   Holdtime Timeout Groups Group prefixes
```

```
10.100.100.100 auto-rp     sparse      150      146      0 233.252.0.0/8
                                                           233.252.0.1/24
10.200.200.200 auto-rp     sparse      150      146      0 233.252.0.2/4

address-family INET6
```

show pim rps brief

The output for the **show pim rps brief** command is identical to that for the **show pim rps** command. For sample output, see show pim rps on page 1681.

show pim rps <group-address>

```
user@host> show pim rps 233.252.0.0
Instance: PIM.master
Instance: PIM.master

RP selected: 10.100.100.100
```

show pim rps <group-address>

```
user@host> show pim rps 233.252.0.0
Instance: PIM.master
Instance: PIM.master

RP selected: 10.100.100.100
```

show pim rps <group-address> (Bidirectional PIM)

```
user@host> show pim rps 233.252.0.1
Instance: PIM.master

233.252.0.0/16
        10.4.12.75 (Bidirectional)

RP selected: 10.4.12.75
```

show pim rps <group-address> (PIM Dense Mode)

```
user@host> show pim rps 233.252.0.1
Instance: PIM.master

Dense Mode active for group 233.252.0.1
```

show pim rps <group-address> (SSM Range Without asm-override-ssm Configured)

```
user@host> show pim rps 233.252.0.1
Instance: PIM.master

Source-specific Mode (SSM) active for group 233.252.0.1
```

show pim rps <group-address> (SSM Range With asm-override-ssm Configured and a Sparse-Mode RP)

```
user@host> show pim rps 233.252.0.1
Instance: PIM.master

Source-specific Mode (SSM) active with Sparse Mode ASM override for group
233.252.0.1
```

```
233.252.0.0/16
        10.4.12.75

RP selected: 10.4.12.75
```

show pim rps <group-address> (SSM Range With asm-override-ssm Configured and a Bidirectional RP)

```
user@host> show pim rps 233.252.0.1
Instance: PIM.master

Source-specific Mode (SSM) active with Sparse Mode ASM override for group
233.252.0.1

233.252.0.0/16
        10.4.12.75 (Bidirectional)

RP selected: (null)
```

show pim rps instance

```
user@host> show pim rps instance VPN-A
Instance: PIM.VPN-A
Address family INET
RP address              Type        Holdtime Timeout Groups Group prefixes
10.10.47.100            static             0    None      1 233.252.0.0/4

Address family INET6
```

show pim rps extensive (PIM Sparse Mode)

```
user@host> show pim rps extensive
Instance: PIM.master

Family: INET
RP: 10.255.245.91
Learned via: static configuration
Time Active: 00:05:48
Holdtime: 45 with 36 remaining
Device Index: 122
Subunit: 32768
Interface: pd-6/0/0.32768
Group Ranges:
    233.252.0.0/4, 36s remaining
Active groups using RP:
    233.252.0.1

    total 1 groups active

Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
233.252.0.1       192.168.195.78  10.255.14.132   10.255.245.91   Receive
 0
```

show pim rps extensive (Bidirectional PIM)

```
user@host> show pim rps extensive
Instance: PIM.master
Address family INET
```

```
RP: 10.10.1.3
Learned via: static configuration
Mode: Bidirectional
Time Active: 01:58:07
Holdtime: 150
Group Ranges:
        233.252.0.0/24
        233.252.0.01/24

RP: 10.10.13.2
Learned via: static configuration
Mode: Bidirectional
Time Active: 01:58:07
Holdtime: 150
Group Ranges:
       233.252.0.3/24
       233.252.0.4/24
```

show pim rps extensive (PIM Anycast RP in Use)

```
user@host> show pim rps extensive
Instance: PIM.master

Family: INET
RP: 10.10.10.2
Learned via: static configuration
Time Active: 00:54:52
Holdtime: 0
Device Index: 130
Subunit: 32769
Interface: pimd.32769
Group Ranges:
    233.252.0.0/4
Active groups using RP:
    233.252.0.10

    total 1 groups active

Anycast-PIM rpset:
           10.100.111.34
           10.100.111.17
           10.100.111.55

Anycast-PIM local address used: 10.100.111.1
Anycast-PIM Register State:
Group                     Source                                Origin
233.252.0.1               10.10.95.2                            DIRECT
233.252.0.2               10.10.95.2                            DIRECT
233.252.0.3               10.10.70.1                            MSDP
233.252.0.4               10.10.70.1                            MSDP
233.252.0.5               10.10.71.1                            DR

Address family INET6

Anycast-PIM rpset:
           ab::1
           ab::2
Anycast-PIM local address used: cd::1

Anycast-PIM Register State:
Group                   Source                              Origin
```

```
::224.1.1.1             ::10.10.95.2                    DIRECT
::224.1.1.2             ::10.10.95.2                    DIRECT
::224.20.20.1           ::10.10.71.1                    DR
```

show pim source

List of Syntax

Syntax on page 1686
Syntax (EX Series Switch and the QFX Series) on page 1686

Syntax

show pim source
<brief | detail>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<*source-prefix*>

Syntax (EX Series Switch and the QFX Series)

show pim source
<brief | detail>
<inet | inet6>
<instance *instance-name*>
<*source-prefix*>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description

Display information about the Protocol Independent Multicast (PIM) source reverse path forwarding (RPF) state.

Options

none—Display standard information about the PIM RPF state for all supported family addresses for all routing instances.

brief | detail—(Optional) Display the specified level of output.

inet | inet6—(Optional) Display information for IPv4 or IPv6 family addresses, respectively.

instance *instance-name*—(Optional) Display information about the RPF state for a specific PIM-enabled routing instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

source-prefix—(Optional) Display the state for source RPF states in the given range.

Required Privilege Level

view

List of Sample Output

show pim source on page 1687
show pim source brief on page 1687
show pim source detail on page 1688

show pim source (Egress Node with Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 1688

Output Fields Table 98 on page 1687 describes the output fields for the **show pim source** command. Output fields are listed in the approximate order in which they appear.

Table 98: show pim source Output Fields

Field Name	Field Description
Instance	Name of the routing instance.
Source	Address of the source or reverse path.
Prefix/length	Prefix and prefix length for the route used to reach the RPF address.
Upstream Protocol	Protocol toward the source address.
Upstream interface	RPF interface toward the source address. A pseudo multipoint LDP (M-LDP) interface appears on egress nodes in M-LDP point-to-multipoint LSPs with inband signaling.
Upstream Neighbor	Address of the RPF neighbor used to reach the source address. The multipoint LDP (M-LDP) root appears on egress nodes in M-LDP point-to-multipoint LSPs with inband signaling.

Sample Output

show pim source

```
user@host> show pim source
Instance: PIM.master Family: INET

Source 10.255.14.144
    Prefix 10.255.14.144/32
    Upstream interface Local
    Upstream neighbor Local

Source 10.255.70.15
    Prefix 10.255.70.15/32
    Upstream interface so-1/0/0.0
    Upstream neighbor 10.111.10.2

Instance: PIM.master Family: INET6
```

show pim source brief

The output for the **show pim source brief** command is identical to that for the **show pim source** command. For sample output, see show pim source on page 1687.

show pim source detail

```
user@host> show pim source detail
Instance: PIM.master Family: INET

Source 10.255.14.144
    Prefix 10.255.14.144/32
    Upstream interface Local
    Upstream neighbor Local
    Active groups:233.252.0.0
        233.252.0.1
        233.252.0.1

Source 10.255.70.15
    Prefix 10.255.70.15/32
    Upstream interface so-1/0/0.0
    Upstream neighbor 10.111.10.2
    Active groups:233.252.0.1

Instance: PIM.master Family: INET6
```

show pim source (Egress Node with Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show pim source
Instance: PIM.master Family: INET

Source 10.1.1.1
    Prefix 10.1.1.1/32
    Upstream interface Local
    Upstream neighbor Local

Source 10.2.7.7
    Prefix 10.2.7.0/24
    Upstream protocol MLDP
    Upstream interface Pseudo MLDP
    Upstream neighbor MLDP LSP root <10.1.1.2>

Source 192.168.219.11
    Prefix 192.168.219.0/28
    Upstream protocol MLDP
    Upstream interface Pseudo MLDP
    Upstream neighbor via MLDP-inband
   Upstream interface fe-1/3/0.0
    Upstream neighbor 192.168.140.1
    Upstream neighbor MLDP LSP root <10.1.1.2>

Instance: PIM.master Family: INET6
Source 2001:db8::1:2:7:7
    Prefix 2001:db8::1:2:7:0/120
    Upstream protocol MLDP
    Upstream interface Pseudo MLDP
    Upstream neighbor via MLDP-inband
   Upstream interface fe-1/3/0.0
    Upstream neighbor 192.168.140.1
Upstream neighbor MLDP LSP root <10.1.1.2>
```

show pim statistics

List of Syntax
Syntax on page 1689
Syntax (EX Series Switch and the QFX Series) on page 1689

Syntax

```
show pim statistics
<inet |inet6)>
<instance instance-name>
<interface interface-name>
<logical-system (all | logical-system-name)>
```

Syntax (EX Series Switch and the QFX Series)

```
show pim statistics
<inet |inet6)>
<instance instance-name>
<interface interface-name>
```

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
inet6 and **instance** options introduced in Junos OS Release 10.0 for EX Series switches.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Support for bidirectional PIM added in Junos OS Release 12.1.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description
Display Protocol Independent Multicast (PIM) statistics.

Options
none—Display PIM statistics.

inet | inet6—(Optional) Display IPv4 or IPv6 PIM statistics, respectively.

instance *instance-name*—(Optional) Display statistics for a specific routing instance enabled by Protocol Independent Multicast (PIM).

interface *interface-name*—(Optional) Display statistics about the specified interface.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

Related Documentation
- clear pim statistics on page 1426

List of Sample Output
show pim statistics on page 1696
show pim statistics inet interface <interface-name> on page 1698
show pim statistics inet6 interface <interface-name> on page 1698
show pim statistics instance <instance-name> on page 1699
show pim statistics interface <interface-name> on page 1701

Output Fields Table 99 on page 1690 describes the output fields for the **show pim statistics** command. Output fields are listed in the approximate order in which they appear.

Table 99: show pim statistics Output Fields

Field Name	Field Description
Instance	Name of the routing instance. This field only appears if you specify an interface, for example: • **inet interface *interface-name*** • **inet6 interface *interface-name*** • **interface *interface-name***
Family	Output is for IPv4 or IPv6 PIM statistics. **INET** indicates IPv4 statistics, and **INET6** indicates IPv6 statistics. This field only appears if you specify an interface, for example: • **inet interface *interface-name*** • **inet6 interface *interface-name*** • **interface *interface-name***
PIM statistics	PIM statistics for all interfaces or for the specified interface.
PIM message type	Message type for which statistics are displayed.
Received	Number of received statistics.
Sent	Number of messages sent of a certain type.
Rx errors	Number of received packets that contained errors.
V2 Hello	PIM version 2 hello packets.
V2 Register	PIM version 2 register packets.
V2 Register Stop	PIM version 2 register stop packets.
V2 Join Prune	PIM version 2 join and prune packets.
V2 Bootstrap	PIM version 2 bootstrap packets.
V2 Assert	PIM version 2 assert packets.
V2 Graft	PIM version 2 graft packets.
V2 Graft Ack	PIM version 2 graft acknowledgment packets.
V2 Candidate RP	PIM version 2 candidate RP packets.

Table 99: show pim statistics Output Fields (continued)

Field Name	Field Description
V2 State Refresh	PIM version 2 control messages related to PIM dense mode (PIM-DM) state refresh. State refresh is an extension to PIM-DM. It not supported in Junos OS.
V2 DF Election	PIM version 2 send and receive messages associated with bidirectional PIM designated forwarder election.
V1 Query	PIM version 1 query packets.
V1 Register	PIM version 1 register packets.
V1 Register Stop	PIM version 1 register stop packets.
V1 Join Prune	PIM version 1 join and prune packets.
V1 RP Reachability	PIM version 1 RP reachability packets.
V1 Assert	PIM version 1 assert packets.
V1 Graft	PIM version 1 graft packets.
V1 Graft Ack	PIM version 1 graft acknowledgment packets.
AutoRP Announce	Auto-RP announce packets.
AutoRP Mapping	Auto-RP mapping packets.
AutoRP Unknown type	Auto-RP packets with an unknown type.
Anycast Register	Auto-RP announce packets.
Anycast Register Stop	Auto-RP announce packets.
Global Statistics	Summary of PIM statistics for all interfaces.
Hello dropped on neighbor policy	Number of hello packets dropped because of a configured neighbor policy.
Unknown type	Number of PIM control packets received with an unknown type.
V1 Unknown type	Number of PIM version 1 control packets received with an unknown type.
Unknown Version	Number of PIM control packets received with an unknown version. The version is not version 1 or version 2.

Table 99: show pim statistics Output Fields (continued)

Field Name	Field Description
Neighbor unknown	Number of PIM control packets received (excluding PIM hello) without first receiving the hello packet.
Bad Length	Number of PIM control packets received for which the packet size does not match the PIM length field in the packet.
Bad Checksum	Number of PIM control packets received for which the calculated checksum does not match the checksum field in the packet.
Bad Receive If	Number of PIM control packets received on an interface that does not have PIM configured.
Rx Bad Data	Number of PIM control packets received that contain data for TCP Bad register packets.
Rx Intf disabled	Number of PIM control packets received on an interface that has PIM disabled.
Rx V1 Require V2	Number of PIM version 1 control packets received on an interface configured for PIM version 2.
Rx V2 Require V1	Number of PIM version 2 control packets received on an interface configured for PIM version 1.
Rx Register not RP	Number of PIM register packets received when the routing device is not the RP for the group.
Rx Register no route	Number of PIM register packets received when the RP does not have a unicast route back to the source.
Rx Register no decap if	Number of PIM register packets received when the RP does not have a de-encapsulation interface.
Null Register Timeout	Number of NULL register timeout packets.
RP Filtered Source	Number of PIM packets received when the routing device has a source address filter configured for the RP.
Rx Unknown Reg Stop	Number of register stop messages received with an unknown type.
Rx Join/Prune no state	Number of join and prune messages received for which the routing device has no state.
Rx Join/Prune on upstream if	Number of join and prune messages received on the interface used to reach the upstream routing device, toward the RP.
Rx Join/Prune for invalid group	Number of join or prune messages received for invalid multicast group addresses.

Table 99: show pim statistics Output Fields (continued)

Field Name	Field Description
Rx Join/Prune messages dropped	Number of join and prune messages received and dropped.
Rx sparse join for dense group	Number of PIM sparse mode join messages received for a group that is configured for dense mode.
Rx Graft/Graft Ack no state	Number of graft and graft acknowledgment messages received for which the router or switch has no state.
Rx Graft on upstream if	Number of graft messages received on the interface used to reach the upstream routing device, toward the RP.
Rx CRP not BSR	Number of BSR messages received in which the PIM message type is Candidate-RP-Advertisement, not Bootstrap.
Rx BSR when BSR	Number of BSR messages received in which the PIM message type is Bootstrap.
Rx BSR not RPF if	Number of BSR messages received on an interface that is not the RPF interface.
Rx unknown hello opt	Number of PIM hello packets received with options that Junos OS does not support.
Rx data no state	Number of PIM control packets received for which the routing device has no state for the data type.
Rx RP no state	Number of PIM control packets received for which the routing device has no state for the RP.
Rx aggregate	Number of PIM aggregate MDT packets received.
Rx malformed packet	Number of PIM control packets received with a malformed IP unicast or multicast address family.
No RP	Number of PIM control packets received with no RP address.
No register encap if	Number of PIM register packets received when the first-hop routing device does not have an encapsulation interface.
No route upstream	Number of PIM control packets received when the routing device does not have a unicast route to the the interface used to reach the upstream routing device, toward the RP.
Nexthop Unusable	Number of PIM control packets with an unusable nexthop. A path can be unusable if the route is hidden or the link is down.
RP mismatch	Number of PIM control packets received for which the routing device has an RP mismatch.

Table 99: show pim statistics Output Fields (continued)

Field Name	Field Description
RP mode mismatch	RP mode (sparse or bidirectional) mismatches encountered when processing join and prune messages.
RPF neighbor unknown	Number of PIM control packets received for which the routing device has an unknown RPF neighbor for the source.
Rx Joins/Prunes filtered	The number of join and prune messages filtered because of configured route filters and source address filters.
Tx Joins/Prunes filtered	The number of join and prune messages filtered because of configured route filters and source address filters.
Embedded-RP invalid addr	Number of packets received with an invalid embedded RP address in PIM join messages and other types of messages sent between routing domains.
Embedded-RP limit exceed	Number of times the limit configured with the **maximum-rps** statement is exceeded. The **maximum-rps** statement limits the number of embedded RPs created in a specific routing instance. The range is from 1 through 500. The default is 100.
Embedded-RP added	Number of packets in which the embedded RP for IPv6 is added. The following receive events trigger extraction of an IPv6 embedded RP address on the routing device: • Multicast Listener Discovery (MLD) report for an embedded RP multicast group address • PIM join message with an embedded RP multicast group address • Static embedded RP multicast group address associated with an interface • Packets sent to an embedded RP multicast group address received on the DR An embedded RP node discovered through these receive events is added if it does not already exist on the routing platform.
Embedded-RP removed	Number of packets in which the embedded RP for IPv6 is removed. The embedded RP is removed whenever all PIM join states using this RP are removed or the configuration changes to remove the embedded RP feature.
Rx Register msgs filtering drop	Number of received register messages dropped because of a filter configured for PIM register messages.
Tx Register msgs filtering drop	Number of register messages dropped because of a filter configured for PIM register messages.
Rx Bidir Join/Prune on non-Bidir if	Error counter for join and prune messages received on non-bidirectional PIM interfaces.

Table 99: show pim statistics Output Fields (continued)

Field Name	Field Description
Rx Bidir Join/Prune on non-DF if	Error counter for join and prune messages received on non-designated forwarder interfaces.
V4 (S,G) Maximum	Maximum number of (S,G) IPv4 multicast routes accepted for the VPN routing and forwarding (VRF) routing instance. If this number is met, additional (S,G) entries are not accepted.
V4 (S,G) Accepted	Number of accepted (S,G) IPv4 multicast routes.
V4 (S,G) Threshold	Threshold at which a warning message is logged (percentage of the maximum number of (S,G) IPv4 multicast routes accepted by the device).
V4 (S,G) Log Interval	Time (in seconds) between consecutive log messages.
V6 (S,G) Maximum	Maximum number of (S,G) IPv6 multicast routes accepted for the VPN routing and forwarding (VRF) routing instance. If this number is met, additional (S,G) entries are not accepted.
V6 (S,G) Accepted	Number of accepted (S,G) IPv6 multicast routes.
V6 (S,G) Threshold	Threshold at which a warning message is logged (percentage of the maximum number of (S,G) IPv6 multicast routes accepted by the device).
V6 (S,G) Log Interval	Time (in seconds) between consecutive log messages.
V4 (grp-prefix, RP) Maximum	Maximum number of group-to-rendezvous point (RP) IPv4 multicast mappings accepted for the VRF routing instance. If this number is met, additional mappings are not accepted.
V4 (grp-prefix, RP) Accepted	Number of accepted group-to-RP IPv4 multicast mappings.
V4 (grp-prefix, RP) Threshold	Threshold at which a warning message is logged (percentage of the maximum number of group-to-RP IPv4 multicast mappings accepted by the device).
V4 (grp-prefix, RP) Log Interval	Time (in seconds) between consecutive log messages.
V6 (grp-prefix, RP) Maximum	Maximum number of group-to RP IPv6 multicast mappings accepted for the VRF routing instance. If this number is met, additional mappings are not accepted.
V6 (grp-prefix, RP) Accepted	Number of accepted group-to-RP IPv6 multicast mappings.

Table 99: show pim statistics Output Fields (continued)

Field Name	Field Description
V6 (grp-prefix, RP) Threshold	Threshold at which a warning message is logged (percentage of the maximum number of group-to-RP IPv6 multicast mappings accepted by the device).
V6 (grp-prefix, RP) Log Interval	Time (in seconds) between consecutive log messages.
V4 Register Maximum	Maximum number of IPv4 PIM registers accepted for the VRF routing instance. If this number is met, additional PIM registers are not accepted. You configure the register limits on the RP.
V4 Register Accepted	Number of accepted IPv4 PIM registers.
V4 Register Threshold	Threshold at which a warning message is logged (percentage of the maximum number of IPv4 PIM registers accepted by the device).
V4 Register Log Interval	Time (in seconds) between consecutive log messages.
V6 Register Maximum	Maximum number of IPv6 PIM registers accepted for the VRF routing instance. If this number is met, additional PIM registers are not accepted. You configure the register limits on the RP.
V6 Register Accepted	Number of accepted IPv6 PIM registers.
V6 Register Threshold	Threshold at which a warning message is logged (percentage of the maximum number of IPv6 PIM registers accepted by the device).
V6 Register Log Interval	Time (in seconds) between consecutive log messages.
(*,G) Join drop due to SSM range check	PIM join messages that are dropped because the multicast addresses are outside of the SSM address range of 232.0.0.0 through 232.255.255.255. You can extend the accepted SSM address range by configuring the ssm-groups statement.

Sample Output

show pim statistics

```
user@host> show pim statistics
PIM Message type        Received       Sent  Rx errors
V2 Hello                      15         32          0
V2 Register                    0        362          0
V2 Register Stop             483          0          0
V2 Join Prune                 18        518          0
V2 Bootstrap                   0          0          0
V2 Assert                      0          0          0
V2 Graft                       0          0          0
```

```
V2 Graft Ack                    0          0          0
V2 Candidate RP                 0          0          0
V2 State Refresh                0          0          0
V2 DF Election                  0          0          0
V1 Query                        0          0          0
V1 Register                     0          0          0
V1 Register Stop                0          0          0
V1 Join Prune                   0          0          0
V1 RP Reachability              0          0          0
V1 Assert                       0          0          0
V1 Graft                        0          0          0
V1 Graft Ack                    0          0          0
AutoRP Announce                 0          0          0
AutoRP Mapping                  0          0          0
AutoRP Unknown type             0
Anycast Register                0          0          0
Anycast Register Stop           0          0          0

Global Statistics

Hello dropped on neighbor policy              0
Unknown type                                  0
V1 Unknown type                               0
Unknown Version                               0
ipv4 BSR pkt drop due to excessive rate       0
ipv6 BSR pkt drop due to excessive rate       0
Neighbor unknown                              0
Bad Length                                    0
Bad Checksum                                  0
Bad Receive If                                0
Rx Bad Data                                   0
Rx Intf disabled                              0
Rx V1 Require V2                              0
Rx V2 Require V1                              0
Rx Register not RP                            0
Rx Register no route                          0
Rx Register no decap if                       0
Null Register Timeout                         0
RP Filtered Source                            0
Rx Unknown Reg Stop                           0
Rx Join/Prune no state                        0
Rx Join/Prune on upstream if                  0
Rx Join/Prune for invalid group               5
Rx Join/Prune messages dropped                0
Rx sparse join for dense group                0
Rx Graft/Graft Ack no state                   0
Rx Graft on upstream if                       0
Rx CRP not BSR                                0
Rx BSR when BSR                               0
Rx BSR not RPF if                             0
Rx unknown hello opt                          0
Rx data no state                              0
Rx RP no state                                0
Rx aggregate                                  0
Rx malformed packet                           0
Rx illegal TTL                                0
Rx illegal destination address                0
No RP                                         0
No register encap if                          0
No route upstream                             0
Nexthop Unusable                              0
```

```
RP mismatch                                0
RP mode mismatch                           0
RPF neighbor unknown                       0
Rx Joins/Prunes filtered                   0
Tx Joins/Prunes filtered                   0
Embedded-RP invalid addr                   0
Embedded-RP limit exceed                   0
Embedded-RP added                          0
Embedded-RP removed                        0
Rx Register msgs filtering drop            0
Tx Register msgs filtering drop            0
Rx Bidir Join/Prune on non-Bidir if        0
Rx Bidir Join/Prune on non-DF if           0
(*,G) Join drop due to SSM range check     0
```

Sample Output

show pim statistics inet interface <interface-name>

```
user@host> show pim statistics inet interface ge-0/3/0.0
Instance: PIM.master Family: INET

PIM Interface statistics for ge-0/3/0.0

PIM Message type        Received       Sent  Rx errors
V2 Hello                       0          4          0
V2 Register                    0          0          0
V2 Register Stop               0          0          0
V2 Join Prune                  0          0          0
V2 Bootstrap                   0          0          0
V2 Assert                      0          0          0
V2 Graft                       0          0          0
V2 Graft Ack                   0          0          0
V2 Candidate RP                0          0          0
V1 Query                       0          0          0
V1 Register                    0          0          0
V1 Register Stop               0          0          0
V1 Join Prune                  0          0          0
V1 RP Reachability             0          0          0
V1 Assert                      0          0          0
V1 Graft                       0          0          0
V1 Graft Ack                   0          0          0
AutoRP Announce                0          0          0
AutoRP Mapping                 0          0          0
AutoRP Unknown type            0
Anycast Register               0          0          0
Anycast Register Stop          0          0          0
```

Sample Output

show pim statistics inet6 interface <interface-name>

```
user@host> show pim statistics inet6 interface ge-0/3/0.0
Instance: PIM.master Family: INET6

PIM Interface statistics for ge-0/3/0.0

PIM Message type        Received       Sent  Rx errors
V2 Hello                       0          4          0
V2 Register                    0          0          0
```

```
V2 Register Stop              0          0          0
V2 Join Prune                 0          0          0
V2 Bootstrap                  0          0          0
V2 Assert                     0          0          0
V2 Graft                      0          0          0
V2 Graft Ack                  0          0          0
V2 Candidate RP               0          0          0
Anycast Register              0          0          0
Anycast Register Stop         0          0          0
```

show pim statistics instance <instance-name>

```
user@host> show pim statistics instance VPN-A
PIM Message type        Received       Sent  Rx errors
V2 Hello                      31         37          0
V2 Register                    0          0          0
V2 Register Stop               0          0          0
V2 Join Prune                  0         16          0
V2 Bootstrap                   0          0          0
V2 Assert                      0          0          0
V2 Graft                       0          0          0
V2 Graft Ack                   0          0          0
V2 Candidate RP                0          0          0
V2 State Refresh               0          0          0
V2 DF Election                 0          0          0
V1 Query                       0          0          0
V1 Register                    0          0          0
V1 Register Stop               0          0          0
V1 Join Prune                  0          0          0
V1 RP Reachability             0          0          0
V1 Assert                      0          0          0
V1 Graft                       0          0          0
V1 Graft Ack                   0          0          0
AutoRP Announce                0          0          0
AutoRP Mapping                 0          0          0
AutoRP Unknown type            0
Anycast Register               0          0          0
Anycast Register Stop          0          0          0

Global Statistics

Hello dropped on neighbor policy              0
Unknown type                                  0
V1 Unknown type                               0
Unknown Version                               0
Neighbor unknown                              0
Bad Length                                    0
Bad Checksum                                  0
Bad Receive If                                0
Rx Bad Data                                   0
Rx Intf disabled                              0
Rx V1 Require V2                              0
Rx V2 Require V1                              0
Rx Register not RP                            0
Rx Register no route                          0
Rx Register no decap if                       0
Null Register Timeout                         0
RP Filtered Source                            0
Rx Unknown Reg Stop                           0
Rx Join/Prune no state                        0
Rx Join/Prune on upstream if                  0
```

```
Rx Join/Prune for invalid group              0
Rx Join/Prune messages dropped               0
Rx sparse join for dense group               0
Rx Graft/Graft Ack no state                  0
Rx Graft on upstream if                      0
Rx CRP not BSR                               0
Rx BSR when BSR                              0
Rx BSR not RPF if                            0
Rx unknown hello opt                         0
Rx data no state                             0
Rx RP no state                               0
Rx aggregate                                 0
Rx malformed packet                          0
Rx illegal TTL                               0
Rx illegal destination address               0
No RP                                        0
No register encap if                         0
No route upstream                           28
Nexthop Unusable                             0
RP mismatch                                  0
RP mode mismatch                             0
RPF neighbor unknown                         0
Rx Joins/Prunes filtered                     0
Tx Joins/Prunes filtered                     0
Embedded-RP invalid addr                     0
Embedded-RP limit exceed                     0
Embedded-RP added                            0
Embedded-RP removed                          0
Rx Register msgs filtering drop              0
Tx Register msgs filtering drop              0
Rx Bidir Join/Prune on non-Bidir if          0
Rx Bidir Join/Prune on non-DF if             0
V4 (S,G) Maximum                            10
V4 (S,G) Accepted                            9
V4 (S,G) Threshold                          80
V4 (S,G) Log Interval                       80
V6 (S,G) Maximum                             8
V6 (S,G) Accepted                            8
V6 (S,G) Threshold                          50
V6 (S,G) Log Interval                      100
V4 (grp-prefix, RP) Maximum                100
V4 (grp-prefix, RP) Accepted                 5
V4 (grp-prefix, RP) Threshold               80
V4 (grp-prefix, RP) Log Interval            10
V6 (grp-prefix, RP) Maximum                 20
V6 (grp-prefix, RP) Accepted                 0
V6 (grp-prefix, RP) Threshold               90
V6 (grp-prefix, RP) Log Interval            20
V4 Register Maximum                        100
V4 Register Accepted                        10
V4 Register Threshold                       80
V4 Register Log Interval                    10
V6 Register Maximum                         20
V6 Register Accepted                         0
V6 Register Threshold                       90
V6 Register Log Interval                    20
(*,G) Join drop due to SSM range check       0
```

Sample Output

show pim statistics interface <interface-name>

```
user@host> show pim statistics interface ge-0/3/0.0
Instance: PIM.master Family: INET

PIM Interface statistics for ge-0/3/0.0

PIM Message type        Received       Sent  Rx errors
V2 Hello                       0          3          0
V2 Register                    0          0          0
V2 Register Stop               0          0          0
V2 Join Prune                  0          0          0
V2 Bootstrap                   0          0          0
V2 Assert                      0          0          0
V2 Graft                       0          0          0
V2 Graft Ack                   0          0          0
V2 Candidate RP                0          0          0
V1 Query                       0          0          0
V1 Register                    0          0          0
V1 Register Stop               0          0          0
V1 Join Prune                  0          0          0
V1 RP Reachability             0          0          0
V1 Assert                      0          0          0
V1 Graft                       0          0          0
V1 Graft Ack                   0          0          0
AutoRP Announce                0          0          0
AutoRP Mapping                 0          0          0
AutoRP Unknown type            0
Anycast Register               0          0          0
Anycast Register Stop          0          0          0

Instance: PIM.master Family: INET6

PIM Interface statistics for ge-0/3/0.0

PIM Message type        Received       Sent  Rx errors
V2 Hello                       0          3          0
V2 Register                    0          0          0
V2 Register Stop               0          0          0
V2 Join Prune                  0          0          0
V2 Bootstrap                   0          0          0
V2 Assert                      0          0          0
V2 Graft                       0          0          0
V2 Graft Ack                   0          0          0
V2 Candidate RP                0          0          0
Anycast Register               0          0          0
Anycast Register Stop          0          0          0
```

show pim mdt

Syntax

show pim mdt instance *instance-name*
<brief | detail | extensive>
data-mdt-joins
data-mdt-limit
inet
inet6
<incoming | outgoing>
<logical-system (all | logical-system-name)>
<*range*>

Release Information

Command introduced before Junos OS Release 7.4.
Support for IPv6 added in Junos OS Release 17.3R1.

Description

Display information about Protocol Independent Multicast (PIM) default multicast distribution tree (MDT) and the data MDTs in a Layer 3 VPN environment for a routing instance.

Options

instance *instance-name*—Display information about data-MDTs for a specific PIM-enabled routing instance.

brief | detail | extensive—(Optional) Display the specified level of output.

data-mdt-joins— Show received PIM data-mdt-joins.

data-mdt-limits— Show received PIM data-mdt-limits.

incoming | outgoing—(Optional) Display incoming or outgoing multicast data tunnels, respectively.

inet | inet6—Display IPv4 or IPv6 multicast data tunnels.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

range—(Optional) Display information about an IP address with optional prefix length representing a particular multicast group.

Required Privilege Level

view

List of Sample Output

show pim mdt <variables> instance on page 1703
show pim mdt instance detail on page 1704
show pim mdt instance extensive on page 1704
show pim mdt instance incoming on page 1705
show pim mdt instance outgoing on page 1705
show pim mdt instance (SSM Mode) on page 1705

Output Fields Table 100 on page 1703 describes the output fields for the **show pim mdt** command. Output fields are listed in the approximate order in which they appear.

Table 100: show pim mdt Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the routing instance.	All levels
Tunnel direction	Direction the tunnel faces, from the router's perspective: **Outgoing** or **Incoming**.	All levels
Tunnel mode	Mode the tunnel is operating in: **PIM-SSM** or **PIM-ASM**.	All levels
Default group address	Default multicast group address using this tunnel.	All levels
Default source address	Default multicast source address using this tunnel.	All levels
Default tunnel interface	Default multicast tunnel interface.	All levels
Default tunnel source	Address used as the source address for outgoing PIM control messages.	All levels
C-Group	Customer-facing multicast group address using this tunnel. If you enable dynamic reuse of data MDT group addresses, more than one group address can use the same data MDT.	**detail**
C-Source	IP address of the multicast source in the customer's address space. If you enable dynamic reuse of data MDT group addresses, more than one source address can use the same data MDT.	**detail**
P-Group	Service provider-facing multicast group address using this tunnel.	**detail**
Data tunnel interface	Multicast data tunnel interface that set up the data-MDT tunnel.	**detail**
Last known forwarding rate	Last known rate, in kilobits per second, at which the tunnel was forwarding traffic.	**detail**
Configured threshold rate	Rate, in kilobits per second, above which a data-MDT tunnel is created and below which it is deleted.	**detail**
Tunnel uptime	Time that this data-MDT tunnel has existed. The format is ***hours:minutes:seconds***.	**detail**

Sample Output

show pim mdt <variables> instance

Use this command to display MDT information for default MDT and data-MDT for IPv4 and/or IPv6 traffic.)

```
user@host> show pim mdt inet | inet6 instance VPN-A
Instance: PIM.VPN-A  Family: INET
Tunnel direction: Outgoing
Tunnel mode: PIM-SM
Default group address: 224.1.1.1
Default source address: 0.0.0.0
Default tunnel interface: mt-0/0/0.32768
Default tunnel source: 0.0.0.0

C-group address   C-source address   P-group address    Data tunnel interface
227.1.1.1         18.1.1.2           228.1.1.1          mt-0/0/0.32769

Instance: PIM.VPN-A
Tunnel direction: Incoming
Tunnel mode: PIM-SM
Default group address: 224.1.1.1
Default source address: 0.0.0.0
Default tunnel interface: mt-0/0/0.1081344
Default tunnel source: 0.0.0.0

Instance: PIM.VPN-A Family: INET6
```

show pim mdt instance detail

```
user@host> show pim mdt instance VPN-A detail
Instance: PIM.VPN-A
Tunnel direction: Outgoing
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.32768
Default tunnel source: 192.168.7.1

C-Group: 235.1.1.2
    C-Source: 192.168.195.74
    P-Group : 228.0.0.0
    Data tunnel interface      : mt-1/1/0.32769
    Last known forwarding rate : 48 kbps (6 kBps)
    Configured threshold rate  : 10 kbps
    Tunnel uptime              : 00:00:34

Instance: PIM.VPN-A
Tunnel direction: Incoming
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.1081344
```

show pim mdt instance extensive

```
user@host> show pim mdt instance VPN-A extensive
Instance: PIM.VPN-A
Tunnel direction: Outgoing
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.32768
Default tunnel source: 192.168.7.1

C-Group: 235.1.1.2
    C-Source: 192.168.195.74
    P-Group : 228.0.0.0
    Data tunnel interface      : mt-1/1/0.32769
    Last known forwarding rate : 48 kbps (6 kBps)
    Configured threshold rate  : 10 kbps
    Tunnel uptime              : 00:00:41
```

```
Instance: PIM.VPN-A
Tunnel direction: Incoming
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.1081344
```

show pim mdt instance incoming

```
user@host> show pim mdt instance VPN-A incoming
Instance: PIM.VPN-A
Tunnel direction: Incoming
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.1081344
```

show pim mdt instance outgoing

```
user@host> show pim mdt instance VPN-A outgoing
Instance: PIM.VPN-A
Tunnel direction: Outgoing
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.32768
Default tunnel source: 192.168.7.1

C-group address   C-source address   P-group address    Data tunnel interface
235.1.1.2         192.168.195.74     228.0.0.0          mt-1/1/0.32769
```

show pim mdt instance (SSM Mode)

```
user@host> show pim mdt instance vpn-a
Instance: PIM.vpn-a
Tunnel direction: Outgoing
Tunnel mode: PIM-SSM
Default group address: 232.1.1.1
Default source address: 10.255.14.216
Default tunnel interface: mt-1/3/0.32769
Default tunnel source: 192.168.7.1

Instance: PIM.vpn-a
Tunnel direction: Incoming
Tunnel mode: PIM-SSM
Default group address: 232.1.1.1
Default source address: 10.255.14.217
Default tunnel interface: mt-1/3/0.1081345

Instance: PIM.vpn-a
Tunnel direction: Incoming
Tunnel mode: PIM-SSM
Default group address: 232.1.1.1
Default source address: 10.255.14.218
Default tunnel interface: mt-1/3/0.1081345
```

show pim mdt data-mdt-joins

Syntax **show pim mdt data-mdt-joins**
<logical-system (all | *logical-system-name*)> instance *instance-name*

Release Information Command introduced in Junos OS Release 11.2.

Description In a draft-rosen Layer 3 multicast virtual private network (MVPN) configured with service provider tunnels, display the advertisements of new multicast distribution tree (MDT) group addresses cached by the provider edge (PE) routers in the specified VPN routing and forwarding (VRF) instance that is configured to use the Protocol Independent Multicast (PIM) protocol.

Options **instance *instance-name***—Display data MDT join packets cached by PE routers in a specific PIM instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

NOTE: **Draft-rosen multicast VPNs are not supported in a logical system environment even though the configuration statements can be configured under the logical-systems hierarchy.**

Required Privilege Level view

Related Documentation

- Understanding Data MDTs on page 499
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

List of Sample Output **show pim mdt data-mdt-joins on page 1707**

Output Fields Table 101 on page 1707 describes the output fields for the **show pim mdt data-mdt-joins** command. Output fields are listed in the approximate order in which they appear.

Table 101: show pim mdt data-mdt-joins Output Fields

Field Name	Field Description
C-Group	IPv4 group address in the address space of the customer's VPN-specific PIM-enabled routing instance of the multicast traffic destination. This 32-bit value is carried in the C-group field of the MDT join TLV packet.
C-Source	IPv4 address in the address space of the customer's VPN-specific PIM-enabled routing instance of the multicast traffic source. This 32-bit value is carried in the C-source field of the MDT join TLV packet.
P-Group	IPv4 group address in the service provider's address space of the new data MDT that the PE router will use to encapsulate the VPN multicast traffic flow (C-Source, C-Group). This 32-bit value is carried in the P-group field of the MDT join TLV packet.
P-Source	IPv4 address of the PE router.
Timeout	Timeout, in seconds, remaining for this cache entry. When the cache entry is created, this field is set to 180 seconds. After an entry times out, the PE router deletes the entry from its cache and prunes itself off the data MDT.

Sample Output

show pim mdt data-mdt-joins

```
user@host show pim mdt data-mdt-joins instance VPN-A
C-Source        C-Group         P-Source        P-Group         Timeout
20.2.15.9       225.1.1.2       20.0.0.5        239.10.10.0     172
20.2.15.9       225.1.1.3       20.0.0.5        239.10.10.1     172
```

show pim mdt data-mdt-limit

Syntax show pim mdt data-mdt-limit instance *instance-name*
<logical-system (all | *logical-system-name*)>

Release Information Command introduced in Junos OS Release 12.2.

Description Display the maximum number configured and the currently active data multicast distribution trees (MDTs) for a specific VPN routing and forwarding (VRF) instance.

Options **instance *instance-name***—Display data MDT information for the specified VRF instance.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

NOTE: Draft-rosen multicast VPNs are not supported in a logical system environment even though the configuration statements can be configured under the logical-systems hierarchy.

Required Privilege Level view

Related Documentation

- Understanding Data MDTs on page 499
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 502
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 512

List of Sample Output show pim mdt data-mdt-limit on page 1709

Output Fields Table 102 on page 1708 describes the output fields for the **show pim mdt data-mdt-limit** command. Output fields are listed in the approximate order in which they appear.

Table 102: show pim mdt data-mdt-limit Output Fields

Field Name	Field Description
Maximum Data Tunnels	Maximum number of data MDTs created in this VRF instance. If the number is 0, no data MDTs are created for this VRF instance.
Active Data Tunnels	Active number of data MDTs in this VRF instance.

Sample Output

show pim mdt data-mdt-limit

```
user@host show pim mdt data-mdt-limit instance VPN-A
Maximum Data Tunnels                  10
Active Data Tunnels                    2
```

show pim mvpn

Syntax show pim mvpn
<logical-system (all | logical-system-name)>

Release Information Command introduced in Junos OS Release 9.4.

Description Display information about multicast virtual private network (MVPN) instances.

Options **logical-system (all | *logical-system-name*)**—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level view

List of Sample Output show pim mvpn on page 1710

Output Fields Table 103 on page 1710 describes the output fields for the **show pim mvpn** command. Output fields are listed in the approximate order in which they appear.

Table 103: show pim mvpn Output Fields

Field Name	Field Description	Level of Output
Instance	Name of the routing instance.	All levels
VPN-Group	Multicast group address configured for the default multicast distribution tree.	All levels
Mode	Mode the tunnel is operating in: **PIM-MVPN**, **NGEN-MVPN**, **NGEN-TRANSITION** or **None**.	All levels
Tunnel	Type of tunnel: **PIM-SSM**, **PIM-SM**, **NGEN PMSI**, or **None** (VRF-only). If **NGEN-PMSI** is displayed, enter the **show mvpn instance** command for more information.	All levels

Sample Output

show pim mvpn

```
user@host> show pim mvpn
Instance              VPN-Group          Mode               Tunnel
PIM.ce1               232.1.1.1          PIM-MVPN           PIM-SSM
```

show route forwarding-table

List of Syntax

Syntax on page 1711
Syntax (MX Series Routers) on page 1711
Syntax (TX Matrix and TX Matrix Plus Routers) on page 1711

Syntax

show route forwarding-table
<detail | extensive | summary>
<all>
<ccc *interface-name*>
<destination *destination-prefix*>
<family *family* | matching *matching*>
<interface-name *interface-name*>
<label *name*>
<matching *matching*>
<multicast>
<table (default | *logical-system-name/routing-instance-name* | *routing-instance-name*)>
<vlan (all | *vlan-name*)>
<vpn *vpn*>

Syntax (MX Series Routers)

show route forwarding-table
<detail | extensive | summary>
<all>
<bridge-domain (all | *domain-name*)>
<ccc *interface-name*>
<destination *destination-prefix*>
<family *family* | matching *matching*>
<interface-name *interface-name*>
<label *name*>
<learning-vlan-id *learning-vlan-id*>
<matching *matching*>
<multicast>
<table (default | *logical-system-name/routing-instance-name* | *routing-instance-name*)>
<vlan (all | *vlan-name*)>
<vpn *vpn*>

Syntax (TX Matrix and TX Matrix Plus Routers)

show route forwarding-table
<detail | extensive | summary>
<all>
<ccc *interface-name*>
<destination *destination-prefix*>
<family *family* | matching *matching*>
<interface-name *interface-name*>
<matching *matching*>
<label *name*>
<lcc *number*>
<multicast>
<table *routing-instance-name*>
<vpn *vpn*>

Release Information

Command introduced before Junos OS Release 7.4.
Option **bridge-domain** introduced in Junos OS Release 7.5

Option **learning-vlan-id** introduced in Junos OS Release 8.4
Options **all** and **vlan** introduced in Junos OS Release 9.6.
Command introduced in Junos OS Release 11.3 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Display the Routing Engine's forwarding table, including the network-layer prefixes and their next hops. This command is used to help verify that the routing protocol process has relayed the correction information to the forwarding table. The Routing Engine constructs and maintains one or more routing tables. From the routing tables, the Routing Engine derives a table of active routes, called the forwarding table.

NOTE: The Routing Engine copies the forwarding table to the Packet Forwarding Engine, the part of the router that is responsible for forwarding packets. To display the entries in the Packet Forwarding Engine's forwarding table, use the **show pfe route** command.

Options

none—Display the routes in the forwarding tables. By default, the **show route forwarding-table** command does not display information about private, or internal, forwarding tables.

detail | extensive | summary—(Optional) Display the specified level of output.

all—(Optional) Display routing table entries for all forwarding tables, including private, or internal, tables.

bridge-domain (all | *bridge-domain-name*)—(MX Series routers only) (Optional) Display route entries for all bridge domains or the specified bridge domain.

ccc *interface-name*—(Optional) Display route entries for the specified circuit cross-connect interface.

destination *destination-prefix*—(Optional) Destination prefix.

family *family*—(Optional) Display routing table entries for the specified family: **bridge (ccc | destination | detail | extensive | interface-name | label | learning-vlan-id | matching | multicast | summary | table | vlan | vpn)**, **ethernet-switching**, **evpn**, **fibre-channel**, **fmembers**, **inet**, **inet6**, **iso**, **mcsnoop-inet**, **mcsnoop-inet6**, **mpls**, **satellite-inet**, **satellite-inet6**, **satellite-vpls**, **tnp**, **unix**, **vpls**, or **vlan-classification**.

interface-name *interface-name*—(Optional) Display routing table entries for the specified interface.

label *name*—(Optional) Display route entries for the specified label.

lcc *number*—(TX Matrix and TX matrix Plus routers only) (Optional) On a routing matrix composed of a TX Matrix router and T640 routers, display information for the specified T640 router (or line-card chassis) connected to the TX Matrix router. On a routing matrix composed of the TX Matrix Plus router and T1600 or T4000 routers,

display information for the specified router (line-card chassis) connected to the TX Matrix Plus router.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

learning-vlan-id *learning-vlan-id*—(MX Series routers only) (Optional) Display learned information for all VLANs or for the specified VLAN.

matching *matching*—(Optional) Display routing table entries matching the specified prefix or prefix length.

multicast—(Optional) Display routing table entries for multicast routes.

table —(Optional) Display route entries for all the routing tables in the main routing instance or for the specified routing instance. If your device supports logical systems, you can also display route entries for the specified logical system and routing instance. To view the routing instances on your device, use the **show route instance** command.

vlan (all | *vlan-name*)—(Optional) Display information for all VLANs or for the specified VLAN.

vpn *vpn*—(Optional) Display routing table entries for a specified VPN.

Required Privilege Level view

List of Sample Output

show route forwarding-table table default on page 1725
show route forwarding-table table
logical-system-name/routing-instance-name on page 1726
show route forwarding-table vpn on page 1727

Output Fields Table 104 on page 1714 lists the output fields for the **show route forwarding-table** command. Output fields are listed in the approximate order in which they appear. Field names might be abbreviated (as shown in parentheses) when no level of output is specified, or when the **detail** keyword is used instead of the **extensive** keyword.

Table 104: show route forwarding-table Output Fields

Field Name	Field Description	Level of Output
Logical system	Name of the logical system. This field is displayed if you specify the **table** ***logical-system-name/routing-instance-name*** option on a device that is configured for and supports logical systems.	All levels
Routing table	Name of the routing table (for example, inet, inet6, mpls).	All levels

Table 104: show route forwarding-table Output Fields (continued)

Field Name	Field Description	Level of Output
Enabled protocols	The features and protocols that have been enabled for a given routing table. This field can contain the following values: • BUM hashing—BUM hashing is enabled. • MAC Stats—Mac Statistics is enabled. • Bridging—Routing instance is a normal layer 2 bridge. • No VLAN—No VLANs are associated with the bridge domain. • All VLANs—The **vlan-id all** statement has been enabled for this bridge domain. • Single VLAN—Single VLAN ID is associated with the bridge domain. • MAC action drop—New MACs will be dropped when the MAC address limit is reached. • Dual VLAN—Dual VLAN tags are associated with the bridge domain • No local switching—No local switching is enabled for this routing instance.. • Learning disabled—Layer 2 learning is disabled for this routing instance. • MAC limit reached—The maximum number of MAC addresses that was configured for this routing instance has been reached. • VPLS—The VPLS protocol is enabled. • No IRB l2-copy—The no-irb-layer-2-copy feature is enabled for this routing instance. • ACKed by all peers—All peers have acknowledged this routing instance. • BUM Pruning—BUM pruning is enabled on the VPLS instance. • Def BD VXLAN—VXLAN is enabled for the default bridge domain. • EVPN—EVPN protocol is enabled for this routing instance. • Def BD OVSDB—Open vSwitch Database (OVSDB) is enabled on the default bridge domain. • Def BD Ingress replication—VXLAN ingress node replication is enabled on the default bridge domain. • L2 backhaul—Layer 2 backhaul is enabled. • FRR optimize—Fast reroute optimization • MAC pinning—MAC pinning is enabled for this bridge domain. • MAC Aging Timer—The MAC table aging time is set per routing instance. • EVPN VXLAN—This routing instance supports EVPN with VXLAN encapsulation. • PBBN—This routing instance is configured as a provider backbone bridged network. • PBN—This routing instance is configured as a provider bridge network. • ETREE—The ETREE protocol is enabled on this EVPN routing instance. • ARP/NDP suppression—EVPN ARP NDP suppression is enabled in this routing instance. • Def BD EVPN VXLAN—EVPN VXLAN is enabled for the default bridge domain. • MPLS control word—Control word is enabled for this MPLS routing instance.	All levels
Address family	Address family (for example, **IP**, **IPv6**, **ISO**, **MPLS**, and **VPLS**).	All levels
Destination	Destination of the route.	**detail extensive**

Table 104: show route forwarding-table Output Fields (continued)

Field Name	Field Description	Level of Output
Route Type (Type)	How the route was placed into the forwarding table. When the **detail** keyword is used, the route type might be abbreviated (as shown in parentheses): • **cloned** (**clon**)—(TCP or multicast only) Cloned route. • **destination** (**dest**)—Remote addresses directly reachable through an interface. • **destination down** (**iddn**)—Destination route for which the interface is unreachable. • **interface cloned** (**ifcl**)—Cloned route for which the interface is unreachable. • **route down** (**ifdn**)—Interface route for which the interface is unreachable. • **ignore** (**ignr**)—Ignore this route. • **interface** (**intf**)—Installed as a result of configuring an interface. • **permanent** (**perm**)—Routes installed by the kernel when the routing table is initialized. • **user**—Routes installed by the routing protocol process or as a result of the configuration.	All levels
Route Reference (RtRef)	Number of routes to reference.	**detail extensive**
Flags	Route type flags: • **none**—No flags are enabled. • **accounting**—Route has accounting enabled. • **cached**—Cache route. • **incoming-iface** ***interface-number***—Check against incoming interface. • **prefix load balance**—Load balancing is enabled for this prefix. • **rt nh decoupled**—Route has been decoupled from the next hop to the destination. • **sent to PFE**—Route has been sent to the Packet Forwarding Engine. • **static**—Static route.	**extensive**
Next hop	IP address of the next hop to the destination.	**detail extensive**

Table 104: show route forwarding-table Output Fields (continued)

Field Name	Field Description	Level of Output
Next hop Type (Type)	Next-hop type. When the **detail** keyword is used, the next-hop type might be abbreviated (as indicated in parentheses): • **broadcast (bcst)**—Broadcast. • **deny**—Deny. • **discard (dscd)** —Discard. • **hold**—Next hop is waiting to be resolved into a unicast or multicast type. • **indexed (idxd)**—Indexed next hop. • **indirect (indr)**—Indirect next hop. • **local (locl)**—Local address on an interface. • **routed multicast (mcrt)**—Regular multicast next hop. • **multicast (mcst)**—Wire multicast next hop (limited to the LAN). • **multicast discard (mdsc)**—Multicast discard. • **multicast group (mgrp)**—Multicast group member. • **receive (recv)**—Receive. • **reject (rjct)**—Discard. An ICMP unreachable message was sent. • **resolve (rslv)**—Resolving the next hop. • **unicast (ucst)**—Unicast. • **unilist (ulst)**—List of unicast next hops. A packet sent to this next hop goes to any next hop in the list.	**detail extensive**
Index	Software index of the next hop that is used to route the traffic for a given prefix.	**detail extensive** none
Route interface-index	Logical interface index from which the route is learned. For example, for interface routes, this is the logical interface index of the route itself. For static routes, this field is zero. For routes learned through routing protocols, this is the logical interface index from which the route is learned.	**extensive**
Reference (NhRef)	Number of routes that refer to this next hop.	**detail extensive** none
Next-hop interface (Netif)	Interface used to reach the next hop.	**detail extensive** none
Weight	Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible (see the **Balance** field description).	**extensive**
Balance	Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a router is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing.	**extensive**
RPF interface	List of interfaces from which the prefix can be accepted. Reverse path forwarding (RPF) information is displayed only when **rpf-check** is configured on the interface.	**extensive**

Sample Output

show route forwarding-table

```
user@host> show route forwarding-table
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    46     4
0.0.0.0/32         perm     0                    dscd    44     1
172.16.1.0/24         ifdn     0                    rslv   608     1 ge-2/0/1.0
172.16.1.0/32         iddn     0 172.16.1.0           recv   606     1 ge-2/0/1.0
172.16.1.1/32         user     0                    rjct    46     4
172.16.1.1/32         intf     0 172.16.1.1           locl   607     2
172.16.1.1/32         iddn     0 172.16.1.1           locl   607     2
172.16.1.255/32       iddn     0 ff:ff:ff:ff:ff:ff  bcst   605     1 ge-2/0/1.0
10.0.0.0/24        intf     0                    rslv   616     1 ge-2/0/0.0
10.0.0.0/32        dest     0 10.0.0.0           recv   614     1 ge-2/0/0.0
10.0.0.1/32        intf     0 10.0.0.1           locl   615     2
10.0.0.1/32        dest     0 10.0.0.1           locl   615     2
10.0.0.255/32      dest     0 10.0.0.255         bcst   613     1 ge-2/0/0.0
10.1.1.0/24        ifdn     0                    rslv   612     1 ge-2/0/1.0
10.1.1.0/32        iddn     0 10.1.1.0           recv   610     1 ge-2/0/1.0
10.1.1.1/32        user     0                    rjct    46     4
10.1.1.1/32        intf     0 10.1.1.1           locl   611     2
10.1.1.1/32        iddn     0 10.1.1.1           locl   611     2
10.1.1.255/32      iddn     0 ff:ff:ff:ff:ff:ff  bcst   609     1 ge-2/0/1.0
10.206.0.0/16      user     0 10.209.63.254      ucst   419    20 fxp0.0
10.209.0.0/16      user     1 0:12:1e:ca:98:0    ucst   419    20 fxp0.0
10.209.0.0/18      intf     0                    rslv   418     1 fxp0.0
10.209.0.0/32      dest     0 10.209.0.0         recv   416     1 fxp0.0
10.209.2.131/32    intf     0 10.209.2.131       locl   417     2
10.209.2.131/32    dest     0 10.209.2.131       locl   417     2
10.209.17.55/32    dest     0 0:30:48:5b:78:d2   ucst   435     1 fxp0.0
10.209.63.42/32    dest     0 0:23:7d:58:92:ca   ucst   434     1 fxp0.0
10.209.63.254/32   dest     0 0:12:1e:ca:98:0    ucst   419    20 fxp0.0
10.209.63.255/32   dest     0 10.209.63.255      bcst   415     1 fxp0.0
10.227.0.0/16      user     0 10.209.63.254      ucst   419    20 fxp0.0

...

Routing table: iso
ISO:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                   rjct    27     1
47.0005.80ff.f800.0000.0108.0003.0102.5524.5220.00
intf     0                   locl    28     1

Routing table: inet6
Internet6:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                   rjct     6     1
ff00::/8           perm     0                   mdsc     4     1
ff02::1/128        perm     0 ff02::1           mcst     3     1

Routing table: ccc
MPLS:
Interface.Label    Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct 16     1
100004(top)fe-0/0/1.0
```

show route forwarding-table detail

```
user@host> show route forwarding-table detail
Routing table: inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            user     2 0:90:69:8e:b1:1b   ucst   132     4 fxp0.0
default            perm     0                    rjct    14     1
10.1.1.0/24        intf     0 ff.3.0.21          ucst   322     1 so-5/3/0.0
10.1.1.0/32        dest     0 10.1.1.0           recv   324     1 so-5/3/0.0
10.1.1.1/32        intf     0 10.1.1.1           locl   321     1
10.1.1.255/32      dest     0 10.1.1.255         bcst   323     1 so-5/3/0.0
10.21.21.0/24      intf     0 ff.3.0.21          ucst   326     1 so-5/3/0.0
10.21.21.0/32      dest     0 10.21.21.0         recv   328     1 so-5/3/0.0
10.21.21.1/32      intf     0 10.21.21.1         locl   325     1
10.21.21.255/32    dest     0 10.21.21.255       bcst   327     1 so-5/3/0.0
127.0.0.1/32       intf     0 127.0.0.1          locl   320     1
172.17.28.19/32    clon     1 192.168.4.254      ucst   132     4 fxp0.0
172.17.28.44/32    clon     1 192.168.4.254      ucst   132     4 fxp0.0

...

Routing table: private1__.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    46     1
10.0.0.0/8         intf     0                    rslv   136     1 fxp1.0
10.0.0.0/32        dest     0 10.0.0.0           recv   134     1 fxp1.0
10.0.0.4/32        intf     0 10.0.0.4           locl   135     2
10.0.0.4/32        dest     0 10.0.0.4           locl   135     2

...

Routing table: iso
ISO:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    38     1

Routing table: inet6
Internet6:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    22     1
ff00::/8           perm     0                    mdsc    21     1
ff02::1/128        perm     0 ff02::1            mcst    17     1

...

Routing table: mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct 28     1
```

show route forwarding-table destination extensive (Weights and Balances)

```
user@host> show route forwarding-table destination 3.4.2.1 extensive
Routing table: inet [Index 0]
Internet:

Destination:  3.4.2.1/32
  Route type: user
```

```
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Next-hop type: unilist                Index: 262143   Reference: 1
  Nexthop: 172.16.4.4
  Next-hop type: unicast                Index: 335      Reference: 2
  Next-hop interface: so-1/1/0.0        Weight: 22      Balance: 3
  Nexthop: 145.12.1.2
  Next-hop type: unicast                Index: 337      Reference: 2
  Next-hop interface: so-0/1/2.0        Weight: 33      Balance: 33
```

show route forwarding-table extensive

```
user@host> show route forwarding-table extensive
Routing table: inet [Index 0]
Internet:

Destination:  default
  Route type: user
  Route reference: 2                    Route interface-index: 0
  Flags: sent to PFE
  Nexthop: 00:00:5E:00:53:1b
  Next-hop type: unicast                Index: 132      Reference: 4
  Next-hop interface: fxp0.0

Destination:  default
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: none
  Next-hop type: reject                 Index: 14       Reference: 1

Destination:  127.0.0.1/32
  Route type: interface
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Nexthop: 127.0.0.1
  Next-hop type: local                  Index: 320      Reference: 1

...

Routing table: private1__.inet [Index 1]
Internet:

Destination:  default
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Next-hop type: reject                 Index: 46       Reference: 1

Destination:  10.0.0.0/8
  Route type: interface
  Route reference: 0                    Route interface-index: 3
  Flags: sent to PFE
  Next-hop type: resolve                Index: 136      Reference: 1
  Next-hop interface: fxp1.0

...

Routing table: iso [Index 0]
ISO:

Destination:  default
```

```
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Next-hop type: reject                 Index: 38       Reference: 1

Routing table: inet6 [Index 0]
Internet6:

Destination:  default
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Next-hop type: reject                 Index: 22       Reference: 1

Destination:  ff00::/8
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Next-hop type: multicast discard      Index: 21       Reference: 1

...

Routing table: private1__.inet6 [Index 1]
Internet6:

Destination:  default
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Next-hop type: reject                 Index: 54       Reference: 1

Destination:  fe80::2a0:a5ff:fe3d:375/128
  Route type: interface
  Route reference: 0                    Route interface-index: 0
  Flags: sent to PFE
  Nexthop: fe80::2a0:a5ff:fe3d:375
  Next-hop type: local                  Index: 75       Reference: 1

...
```

show route forwarding-table extensive (RPF)

The next example is based on the following configuration, which enables an RPF check on all routes that are learned from this interface, including the interface route:

```
so-1/1/0 {
    unit 0 {
        family inet {
            rpf-check;
            address 192.0.2.2/30;
        }
    }
}
```

```
user@host> show route forwarding-table extensive
Routing table: inet [Index 0]
Internet:
...
...
Destination:  192.0.2.3/32
```

```
  Route type: destination
  Route reference: 0                    Route interface-index: 67
  Flags: sent to PFE
  Nexthop: 192.0.2.3
  Next-hop type: broadcast              Index: 328      Reference: 1
  Next-hop interface: so-1/1/0.0
  RPF interface: so-1/1/0.0
```

show route forwarding-table (dynamic list next hop)

The **show route forwarding table** output shows the two next hop elements for a multihomed EVPN destination.

```
user@host> show route forwarding-table label 299952 extensive
MPLS:

Destination:  299952
  Route type: user
  Route reference: 0                    Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE, rt nh decoupled
  Next-hop type: indirect               Index: 1048575  Reference: 2
  Nexthop:
  Next-hop type: composite              Index: 601      Reference: 2
  Next-hop type: indirect               Index: 1048574  Reference: 3
  Nexthop: 1.0.0.4
  Next-hop type: Push 301632, Push 299776(top) Index: 600 Reference: 2
  Load Balance Label: None
  Next-hop interface: ge-0/0/1.0
  Next-hop type: indirect               Index: 1048577  Reference: 3
  Nexthop: 1.0.0.4
  Next-hop type: Push 301344, Push 299792(top) Index: 619 Reference: 2
  Load Balance Label: None
  Next-hop interface: ge-0/0/1.0
```

After one of the PE router has been disabled in the EVPN multihomed network, the same **show route forwarding table** output command shows one next hop element and one empty next hop element.

```
user@host> show route forwarding-table label 299952 extensive
Routing table: default.mpls [Index 0]
MPLS:

Destination:  299952
  Route type: user
  Route reference: 0                    Route interface-index: 0
  Multicast RPF nh index: 0
  P2mpidx: 0
  Flags: sent to PFE, rt nh decoupled
  Next-hop type: indirect               Index: 1048575  Reference: 2
  Nexthop:
  Next-hop type: composite              Index: 601      Reference: 2
  Next-hop type: indirect               Index: 1048577  Reference: 3
  Nexthop: 1.0.0.4
  Next-hop type: Push 301344, Push 299792(top) Index: 619 Reference: 2
  Load Balance Label: None
  Next-hop interface: ge-0/0/1.0
```

show route forwarding-table family mpls

```
user@host> show route forwarding-table family mpls
Routing table: mpls
MPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            perm     0                   rjct    19     1
0                  user     0                   recv    18     3
1                  user     0                   recv    18     3
2                  user     0                   recv    18     3
100000             user     0 10.31.1.6         swap  100001     fe-1/1/0.0
800002             user     0                   Pop              vt-0/3/0.32770

vt-0/3/0.32770 (VPLS)
                   user     0                   indr   351     4
                                                Push 800000, Push 100002(top)
so-0/0/0.0
```

show route forwarding-table family mpls ccc ge-0/0/1.1004

```
user@host>show route forwarding-table mpls ccc ge-0/0/1.1004
Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop          Type  Index    NhRef Netif
ge-0/0/1.1004  (CCC) user     0                    ulst   1048577     2
                                                   comp      754        3
                                                  comp      755        3
                                                comp      756        3

Routing table: __mpls-oam__.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
default            perm     0                    dscd      556     1
```

show route forwarding-table family vpls

```
user@host> show route forwarding-table family vpls
Routing table: green.vpls
VPLS:
Destination        Type RtRef Next hop          Type Index NhRef Netif
default            dynm     0                   flood   353     1
default            perm     0                   rjct    298     1
fe-0/1/0.0         dynm     0                   flood   355     1
00:00:5E:00:53:1f/48             <<<<<Remote CE

                   dynm     0                   indr   351     4
                                                Push 800000, Push 100002(top)
so-0/0/0.0
00:00:5E:00:53:1f/48             <<<<<<Local CE

                   dynm     0                   ucst   354     2 fe-0/1/0.0
```

show route forwarding-table vpls (Broadcast, unknown unicast, and multicast (BUM) hashing is enabled)

```
user@host> show route forwarding-table vpls
Routing table: green.vpls
VPLS:
Enabled protocols: BUM hashing
Destination        Type RtRef Next hop           Type Index    NhRef Netif
```

```
default             perm     0                   dscd      519     1
lsi.1048832         intf     0                   indr  1048574     4
                               172.16.3.2          Push 262145      621     2
ge-3/0/0.0
00:00:5E:00:53:01/48 user     0                   ucst      590     5 ge-2/3/9.0
0x30003/51          user     0                   comp      627     2
ge-2/3/9.0          intf     0                   ucst      590     5 ge-2/3/9.0
ge-3/1/3.0          intf     0                   ucst      619     4 ge-3/1/3.0
0x30002/51          user     0                   comp      600     2
0x30001/51          user     0                   comp      597     2
```

show route forwarding-table vpls (Broadcast, unknown unicast, and multicast (BUM) hashing is enabled with MAC Statistics)

```
user@host> show route forwarding-table vpls
Routing table: green.vpls
VPLS:
Enabled protocols: BUM hashing, MAC Stats
Destination         Type RtRef Next hop           Type Index    NhRef Netif
default             perm     0                   dscd      519     1
lsi.1048834         intf     0                   indr  1048574     4
                               172.16.3.2          Push 262145      592     2
ge-3/0/0.0
00:19:e2:25:d0:01/48 user     0                   ucst      590     5 ge-2/3/9.0
0x30003/51          user     0                   comp      630     2
ge-2/3/9.0          intf     0                   ucst      590     5 ge-2/3/9.0
ge-3/1/3.0          intf     0                   ucst      591     4 ge-3/1/3.0
0x30002/51          user     0                   comp      627     2
0x30001/51          user     0                   comp      624     2
```

show route forwarding-table family vpls extensive

```
user@host> show route forwarding-table family vpls extensive
Routing table: green.vpls [Index 2]
VPLS:

Destination:  default
  Route type: dynamic
  Route reference: 0                    Route interface-index: 72
  Flags: sent to PFE
  Next-hop type: flood                  Index: 289      Reference: 1
  Next-hop type: unicast                Index: 291      Reference: 3
  Next-hop interface: fe-0/1/3.0
  Next-hop type: unicast                Index: 290      Reference: 3
  Next-hop interface: fe-0/1/2.0

Destination:  default
  Route type: permanent
  Route reference: 0                    Route interface-index: 0
  Flags: none
  Next-hop type: discard                Index: 341      Reference: 1

Destination:  fe-0/1/2.0
  Route type: dynamic
  Route reference: 0                    Route interface-index: 69
  Flags: sent to PFE
  Next-hop type: flood                  Index: 293      Reference: 1
  Next-hop type: indirect               Index: 363      Reference: 4
  Next-hop type: Push 800016
  Next-hop interface: at-1/0/1.0
```

```
  Next-hop type: indirect              Index: 301      Reference: 5
  Next hop: 10.31.3.2
  Next-hop type: Push 800000
  Next-hop interface: fe-0/1/1.0
  Next-hop type: unicast               Index: 291      Reference: 3
  Next-hop interface: fe-0/1/3.0

Destination:  fe-0/1/3.0
  Route type: dynamic
  Route reference: 0                   Route interface-index: 70
  Flags: sent to PFE
  Next-hop type: flood                 Index: 292      Reference: 1
  Next-hop type: indirect              Index: 363      Reference: 4
  Next-hop type: Push 800016
  Next-hop interface: at-1/0/1.0
  Next-hop type: indirect              Index: 301      Reference: 5
  Next hop: 10.31.3.2
  Next-hop type: Push 800000
  Next-hop interface: fe-0/1/1.0
  Next-hop type: unicast               Index: 290      Reference: 3
  Next-hop interface: fe-0/1/2.0

Destination:  00:00:5E:00:53:01/48
  Route type: dynamic
  Route reference: 0                   Route interface-index: 70
  Flags: sent to PFE, prefix load balance
  Next-hop type: unicast               Index: 291      Reference: 3
  Next-hop interface: fe-0/1/3.0
  Route used as destination:
    Packet count:         6640    Byte count:      675786
  Route used as source
    Packet count:         6894    Byte count:      696424

Destination:  00:00:5E:00:53:04/48
  Route type: dynamic
  Route reference: 0                   Route interface-index: 69
  Flags: sent to PFE, prefix load balance
  Next-hop type: unicast               Index: 290      Reference: 3
  Next-hop interface: fe-0/1/2.0
  Route used as destination:
    Packet count:           96    Byte count:        8079
  Route used as source:
    Packet count:          296    Byte count:       24955

Destination:  00:00:5E:00:53:05/48
  Route type: dynamic
  Route reference: 0                   Route interface-index: 74
  Flags: sent to PFE, prefix load balance
  Next-hop type: indirect              Index: 301      Reference: 5
  Next hop: 10.31.3.2
  Next-hop type: Push 800000
  Next-hop interface: fe-0/1/1.0
```

show route forwarding-table table default

```
user@host> show route forwarding-table table default
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    36     2
0.0.0.0/32         perm     0                    dscd    34     1
```

```
10.0.60.0/30       user     0 10.0.60.13         ucst   713     5 fe-0/1/3.0
10.0.60.12/30      intf     0                    rslv   688     1 fe-0/1/3.0
10.0.60.12/32      dest     0 10.0.60.12         recv   686     1 fe-0/1/3.0
10.0.60.13/32      dest     0 0:5:85:8b:bc:22    ucst   713     5 fe-0/1/3.0
10.0.60.14/32      intf     0 10.0.60.14         locl   687     2
10.0.60.14/32      dest     0 10.0.60.14         locl   687     2
10.0.60.15/32      dest     0 10.0.60.15         bcst   685     1 fe-0/1/3.0
10.0.67.12/30      user     0 10.0.60.13         ucst   713     5 fe-0/1/3.0
10.0.80.0/30       ifdn     0 ff.3.0.21          ucst   676     1 so-0/0/1.0
10.0.80.0/32       dest     0 10.0.80.0          recv   678     1 so-0/0/1.0
10.0.80.2/32       user     0                    rjct    36     2
10.0.80.2/32       intf     0 10.0.80.2          locl   675     1
10.0.80.3/32       dest     0 10.0.80.3          bcst   677     1 so-0/0/1.0
10.0.90.12/30      intf     0                    rslv   684     1 fe-0/1/0.0
10.0.90.12/32      dest     0 10.0.90.12         recv   682     1 fe-0/1/0.0
10.0.90.14/32      intf     0 10.0.90.14         locl   683     2
10.0.90.14/32      dest     0 10.0.90.14         locl   683     2
10.0.90.15/32      dest     0 10.0.90.15         bcst   681     1 fe-0/1/0.0
10.5.0.0/16        user     0 192.168.187.126    ucst   324    15 fxp0.0
10.10.0.0/16       user     0 192.168.187.126    ucst   324    15 fxp0.0
10.13.10.0/23      user     0 192.168.187.126    ucst   324    15 fxp0.0
10.84.0.0/16       user     0 192.168.187.126    ucst   324    15 fxp0.0
10.150.0.0/16      user     0 192.168.187.126    ucst   324    15 fxp0.0
10.157.64.0/19     user     0 192.168.187.126    ucst   324    15 fxp0.0
10.209.0.0/16      user     0 192.168.187.126    ucst   324    15 fxp0.0

...

Routing table: default.iso
ISO:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    60     1

Routing table: default.inet6
Internet6:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct    44     1
::/128             perm     0                    dscd    42     1
ff00::/8           perm     0                    mdsc    43     1
ff02::1/128        perm     0 ff02::1            mcst    39     1

Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd    50     1
```

show route forwarding-table table logical-system-name/routing-instance-name

```
user@host> show route forwarding-table table R4/vpn-red
Logical system: R4
Routing table: vpn-red.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   563     1
0.0.0.0/32         perm     0                    dscd   561     2
172.16.0.1/32         user     0                    dscd   561     2
172.16.2.0/24         intf     0                    rslv   771     1 ge-1/2/0.3
172.16.2.0/32        dest     0 172.16.2.0           recv   769     1 ge-1/2/0.3
172.16.2.1/32         intf     0 172.16.2.1           locl   770     2
172.16.2.1/32         dest     0 172.16.2.1           locl   770     2
172.16.2.2/32         dest     0 0.4.80.3.0.1b.c0.d5.e4.bd.0.1b.c0.d5.e4.bc.8.0
```

```
                                                ucst   789     1 ge-1/2/0.3
172.16.2.255/32       dest     0 172.16.2.255          bcst   768     1 ge-1/2/0.3
172.16.233.0/4         perm     1                     mdsc   562     1
172.16.233.1/32        perm     0 172.16.233.1           mcst   558     1
255.255.255.255/32 perm     0                     bcst   559     1

Logical system: R4
Routing table: vpn-red.iso
ISO:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   608     1

Logical system: R4
Routing table: vpn-red.inet6
Internet6:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    rjct   708     1
::/128             perm     0                    dscd   706     1
ff00::/8           perm     0                    mdsc   707     1
ff02::1/128        perm     0 ff02::1            mcst   704     1

Logical system: R4
Routing table: vpn-red.mpls
MPLS:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd   638
```

show route forwarding-table vpn

```
user@host> show route forwarding-table vpn VPN-A
      Routing table:: VPN-A.inet
Internet:
      Destination        Type RtRef Nexthop           Type Index NhRef Netif
      default            perm     0                   rjct     4     4
      10.39.10.20/30     intf     0 ff.3.0.21         ucst    40     1
      so-0/0/0.0
      10.39.10.21/32     intf     0 10.39.10.21       locl    36     1
      10.255.14.172/32   user     0                   ucst    69     2
      so-0/0/0.0
      10.255.14.175/32   user     0                   indr    81     3
                                                      Push 100004, Push
      100004(top) so-1/0/0.0
      172.16.233.0/4         perm     2                   mdsc     5     3
      172.16.233.1/32        perm     0 172.16.233.1         mcst     1     8
      172.16.233.5/32        user     1 172.16.233.5         mcst     1     8
      255.255.255.255/32 perm     0                   bcst     2     3
```

On QFX5200, the results for this command look like this:

```
show route forwarding-table family mpls
Routing table: default.mpls
MPLS:
Destination Type RtRef Next hop Type Index NhRef Netif
default perm 0 dscd 65 1
0 user 0 recv 64 4
1 user 0 recv 64 4
2 user 0 recv 64 4
13 user 0 recv 64 4
300384 user 0 9.1.1.1 Pop 1711 2 xe-0/0/34.0
300384(S=0) user 0 9.1.1.1 Pop 1712 2 xe-0/0/34.0
```

```
300400 user 0 ulst 131071 2
                              10.1.1.2 Pop 1713 1 xe-0/0/38.0
                              172.16.11.2 Pop 1714 1 xe-0/0/40.0
300400(S=0) user 0 ulst 131072 2
                              10.1.1.2 Pop 1715 1 xe-0/0/38.0
                              172.16.11.2 Pop 1716 1 xe-0/0/40.0

Routing table: __mpls-oam__.mpls
MPLS:
Destination Type RtRef Next hop Type Index NhRef Netif
default perm 0 dscd 1681 1
```

show route label

List of Syntax
Syntax on page 1729
Syntax (EX Series Switches) on page 1729

Syntax
show route label *label*
<brief | detail | extensive | terse>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switches)
show route label *label*
<brief | detail | extensive | terse>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.5 for EX Series switches.

Description
Display the routes based on a specified Multiprotocol Label Switching (MPLS) label value.

Options
label—Value of the MPLS label.

brief | detail | extensive | terse—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to brief.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

Related Documentation
- *Example: Configuring Multipoint LDP In-Band Signaling for Point-to-Multipoint LSPs*

List of Sample Output
show route label terse on page 1730
show route label on page 1730
show route label detail on page 1730
show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 1730
show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs) on page 1731
show route label detail (Multipoint LDP with Multicast-Only Fast Reroute) on page 1731
show route label detail (Dynamic List Next Hop) on page 1732
show route label extensive on page 1733

Output Fields
For information about output fields, see the output field table for the *show route* command, the *show route detail* command, the *show route extensive* command, or the *show route terse* command.

Sample Output

show route label terse

```
user@host> show route label 100016 terse

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* 100016             V 170                       >10.12.80.1
```

show route label

```
user@host> show route label 100016

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
+ = Active Route, - = Last Active, * = Both
100016             *[VPN/170] 03:25:41
                    > to 10.12.80.1 via ge-6/3/2.0, Pop
```

show route label detail

```
user@host> show route label 100016 detail

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
Restart Complete
100016 (1 entry, 1 announced)
        *VPN    Preference: 170
                Next-hop reference count: 2
                Source: 10.12.80.1
                Next hop: 10.12.80.1 via ge-6/3/2.0, selected
                Label operation: Pop
                State: <Active Int Ext>
                Local AS:     1
                Age: 3:23:31
                Task: BGP.0.0.0.0+179
                Announcement bits (1): 0-KRT
                AS path: 100 I
                Ref Cnt: 2
```

show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show route label 299872 detail
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
299872 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Next-hop reference count: 3
                Address: 0x9097d90
                Next hop: via vt-0/1/0.1
                Next-hop index: 661
                Label operation: Pop
                Address: 0x9172130
                Next hop: via so-0/0/3.0
                Next-hop index: 654
                Label operation: Swap 299872
```

```
                    State: **Active Int>
                    Local AS:  1001
                    Age: 8:20       Metric: 1
                    Task: LDP
                    Announcement bits (1): 0-KRT
                    AS path: I
                   FECs bound to route: P2MP root-addr 10.255.72.166, grp 232.1.1.1,
 src 192.168.142.2
```

show route label detail (Multipoint LDP Inband Signaling for Point-to-Multipoint LSPs)

```
user@host> show route label 299872 detail
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
299872 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Next-hop reference count: 3
                Address: 0x9097d90
                Next hop: via vt-0/1/0.1
                Next-hop index: 661
                Label operation: Pop
                Address: 0x9172130
                Next hop: via so-0/0/3.0
                Next-hop index: 654
                Label operation: Swap 299872
                State: **Active Int>
                Local AS:  1001
                Age: 8:20       Metric: 1
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
               FECs bound to route: P2MP root-addr 10.255.72.166, grp 232.1.1.1,
 src 192.168.142.2
```

show route label detail (Multipoint LDP with Multicast-Only Fast Reroute)

```
user@host> show route label 301568 detail

mpls.0: 18 destinations, 18 routes (18 active, 0 holddown, 0 hidden)
301568 (1 entry, 1 announced)
        *LDP    Preference: 9
                Next hop type: Flood
                Address: 0x2735208
                Next-hop reference count: 3
                Next hop type: Router, Next hop index: 1397
                Address: 0x2735d2c
                Next-hop reference count: 3
                Next hop: 1.3.8.2 via ge-1/2/22.0
                Label operation: Pop
                Load balance label:  None;
                Next hop type: Router, Next hop index: 1395
                Address: 0x2736290
                Next-hop reference count: 3
                Next hop: 1.3.4.2 via ge-1/2/18.0
                Label operation: Pop
                Load balance label:  None;
                State: <Active Int AckRequest MulticastRPF>
                Local AS:    10
                Age: 54:05      Metric: 1
                Validation State: unverified
```

```
                    Task: LDP
                    Announcement bits (1): 0-KRT
                    AS path: I
                   FECs bound to route: P2MP root-addr 1.1.1.1, grp: 232.1.1.1, src:
 192.168.219.11
                    Primary Upstream : 1.1.1.3:0--1.1.1.2:0
                       RPF Nexthops  :
                            ge-1/2/15.0, 1.2.94.1, Label: 301568, weight: 0x1
                            ge-1/2/14.0, 1.2.3.1, Label: 301568, weight: 0x1
                    Backup Upstream : 1.1.1.3:0--1.1.1.6:0
                       RPF Nexthops :
                            ge-1/2/20.0, 1.2.96.1, Label: 301584, weight: 0xfffe
                            ge-1/2/19.0, 1.3.6.1, Label: 301584, weight: 0xfffe
```

show route label detail (Dynamic List Next Hop)

The output for **show route label detail** shows the two indirect next hop for an ESI.

```
user@host> show route label 299952 detail
mpls.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
299952 (1 entry, 1 announced)
TSI:
KRT in-kernel 299952 /52 -> {Dyn list:indirect(1048577), indirect(1048574)}
        *EVPN   Preference: 7
                Next hop type: Dynamic List, Next hop index: 1048575
                Address: 0x13f497fc
                Next-hop reference count: 5
                Next hop: ELNH Address 0xb7a3d90 uflags EVPN data
                    Next hop type: Indirect, Next hop index: 0
                    Address: 0xb7a3d90
                    Next-hop reference count: 3
                    Protocol next hop: 10.255.255.2
                    Label operation: Push 301344
                    Indirect next hop: 0x135b5c00 1048577 INH Session ID: 0x181
                        Next hop type: Router, Next hop index: 619
                        Address: 0xb7a3d30
                        Next-hop reference count: 4
                        Next hop: 1.0.0.4 via ge-0/0/1.0
                        Label operation: Push 301344, Push 299792(top)
                        Label TTL action: no-prop-ttl, no-prop-ttl(top)
                        Load balance label: Label 301344: None; Label 299792:
 None;
                        Label element ptr: 0xb7a3cc0
                        Label parent element ptr: 0xb7a34e0
                        Label element references: 1
                        Label element child references: 0
                        Label element lsp id: 0
                Next hop: ELNH Address 0xb7a37f0 uflags EVPN data
                    Next hop type: Indirect, Next hop index: 0
                    Address: 0xb7a37f0
                    Next-hop reference count: 3
                    Protocol next hop: 10.255.255.3
                    Label operation: Push 301632
                    Indirect next hop: 0x135b5480 1048574 INH Session ID: 0x180
                        Next hop type: Router, Next hop index: 600
                        Address: 0xb7a3790
                        Next-hop reference count: 4
                        Next hop: 1.0.0.4 via ge-0/0/1.0
                        Label operation: Push 301632, Push 299776(top)
                        Label TTL action: no-prop-ttl, no-prop-ttl(top)
                        Load balance label: Label 301632: None; Label 299776:
```

```
None;
                        Label element ptr: 0xb7a3720
                        Label parent element ptr: 0xb7a3420
                        Label element references: 1
                        Label element child references: 0
                        Label element lsp id: 0
                State: <Active Int>
                Age: 1:18
                Validation State: unverified
                Task: evpn global task
                Announcement bits (2): 1-KRT 2-evpn global task
                AS path: I
                Routing Instance blue, Route Type Egress-MAC, ESI
00:11:22:33:44:55:66:77:88:99
```

show route label extensive

The output for the show route label extensive command is identical to that of the **show route label detail** command. For sample output, see show route label detail on page 1730.

show route snooping

Syntax

show route snooping
<brief | detail | extensive | terse>
<all>
<best *address/prefix*>
<exact *address*>
<logical-system *logical-system-name*>
<range *prefix-range*>
<summary>
<table *table-name*>

Release Information

Command introduced in Junos OS Release 8.5.
Command introduced in Junos OS Release 9.0 for EX Series switches.

Description

Display the entries in the routing table that were learned from snooping.

Options

none—Display the entries in the routing table that were learned from snooping.

brief | detail | extensive | terse—(Optional) Display the specified level of output. If you do not specify a level of output, the system defaults to **brief**.

all—(Optional) Display all entries, including hidden entries.

best *address/prefix*—(Optional) Display the longest match for the provided address and optional prefix.

exact *address/prefix*—(Optional) Display exact matches for the provided address and optional prefix.

logical-system *logical-system-name*—(Optional) Display information about a particular logical system, or type 'all'.

range *prefix-range*—(Optional) Display information for the provided address range.

summary—(Optional) Display route snooping summary statisitics.

table *table-name*—(Optional) Display information for the named table.

Required Privilege Level

view

List of Sample Output

show route snooping detail on page 1735
show route snooping logical-system all on page 1735

Output Fields

For information about output fields, see the output field tables for the *show route* command, the *show route detail* command, the *show route extensive* command, or the *show route terse* command.

Sample Output

show route snooping detail

```
user@host> show route snooping detail
__+domainAll__.inet.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

224.0.0.2/32 (1 entry, 1 announced)
        *IGMP   Preference: 0
                Next hop type: MultiRecv
                Next-hop reference count: 4
                State: <Active NoReadvrt Int>
                Age: 2:24
                Task: IGMP
                Announcement bits (1): 0-KRT
                AS path: I

224.0.0.22/32 (1 entry, 1 announced)
        *IGMP   Preference: 0
                Next hop type: MultiRecv
                Next-hop reference count: 4
                State: <Active NoReadvrt Int>
                Age: 2:24
                Task: IGMP
                Announcement bits (1): 0-KRT
                AS path: I

__+domainAll__.inet.1: 36 destinations, 36 routes (36 active, 0 holddown, 0 hidden)

224.0.0.0.0.0.0.0/24 (1 entry, 1 announced)
        *Multicast Preference: 180
                Next hop type: Multicast (IPv4), Next hop index: 1048584
                Next-hop reference count: 4
                State: <Active Int>
                Age: 2:24
                Task: MC
                Announcement bits (1): 0-KRT
                AS path: I

<snip>
```

show route snooping logical-system all

```
user@host> show route snooping logical-system all

logical-system: default

inet.1: 20 destinations, 20 routes (20 active, 0 holddown, 0 hidden)
Restart Unsupported
+ = Active Route, - = Last Active, * = Both

0.0,0.1,0.0,232.1.1.65,100.1.1.2/112*[Multicast/180] 00:07:36
                      Multicast (IPv4) Composite
0.0,0.1,0.0,232.1.1.66,100.1.1.2/112*[Multicast/180] 00:07:36
                      Multicast (IPv4) Composite
0.0,0.1,0.0,232.1.1.67,100.1.1.2/112*[Multicast/180] 00:07:36

<snip>

default-switch.inet.1: 237 dest, 237 rts (237 active, 0 holddown, 0 hidden)
```

```
Restart Complete
+ = Active Route, - = Last Active, * = Both

0.15,0.1,0.0,0.0.0.0,0.0.0.0,2/120*[Multicast/180] 00:08:21
                      Multicast (IPv4) Composite
0.15,0.1,0.0,0.0.0.0,0.0.0.0,2,17/128*[Multicast/180] 00:08:21
                      Multicast (IPv4) Composite

<snip>
```

show route table

List of Syntax
Syntax on page 1737
Syntax (EX Series Switches, QFX Series Switches) on page 1737

Syntax
show route table *routing-table-name*
<brief | detail | extensive | terse>
<logical-system (all | *logical-system-name*)>

Syntax (EX Series Switches, QFX Series Switches)
show route table *routing-table-name*
<brief | detail | extensive | terse>

Release Information
Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 14.1X53-D15 for QFX Series switches.
Show route table evpn statement introduced in Junos OS Release 15.1X53-D30 for QFX Series switches.

Description
Display the route entries in a particular routing table.

Options
brief | detail | extensive | terse—(Optional) Display the specified level of output.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

routing-table-name—Display route entries for all routing tables whose names begin with this string (for example, inet.0 and inet6.0 are both displayed when you run the **show route table inet** command).

Required Privilege Level
view

Related Documentation
- *show route summary*

List of Sample Output
show route table bgp.l2.vpn on page 1748
show route table bgp.l3vpn.0 on page 1748
show route table bgp.l3vpn.0 detail on page 1749
show route table bgp.rtarget.0 (When Proxy BGP Route Target Filtering Is Configured) on page 1750
show route table bgp.evpn.0 on page 1750
show route table evpna.evpn.0 on page 1751
show route table inet.0 on page 1751
show route table inet.3 on page 1752
show route table inet.3 protocol ospf on page 1752
show route table inet6.0 on page 1752
show route table inet6.3 on page 1752

show route table inetflow detail on page 1753
show route table lsdist.0 extensive on page 1753
show route table l2circuit.0 on page 1755
show route table mpls on page 1755
show route table mpls extensive on page 1755
show route table mpls.0 on page 1756
show route table mpls.0 detail (PTX Series) on page 1757
show route table mpls.0 ccc ge-0/0/1.1004 detail on page 1757
show route table mpls.0 protocol evpn on page 1758
show route table mpls.0 protocol ospf on page 1764
show route table mpls.0 extensive (PTX Series) on page 1765
show route table mpls.0 (RSVP Route—Transit LSP) on page 1765
show route table vpls_1 detail on page 1766
show route table vpn-a on page 1766
show route table vpn-a.mdt.0 on page 1766
show route table VPN-A detail on page 1767
show route table VPN-AB.inet.0 on page 1767
show route table VPN_blue.mvpn-inet6.0 on page 1768
show route table vrf1.mvpn.0 extensive on page 1768
show route table inetflow detail on page 1768
show route table bgp.evpn.0 extensive |no-more (EVPN) on page 1771

Output Fields Table 105 on page 1738 describes the output fields for the **show route table** command. Output fields are listed in the approximate order in which they appear.

Table 105: show route table Output Fields

Field Name	Field Description
routing-table-name	Name of the routing table (for example, inet.0).

Restart complete	All protocols have restarted for this routing table.
	
	Restart state:
	
	• **Pending:*protocol-name***—List of protocols that have not yet completed graceful restart for this routing table.
	• **Complete**—All protocols have restarted for this routing table.
	
	For example, if the output shows-
	
	• `LDP.inet.0 : 5 routes (4 active, 1 holddown, 0 hidden)`
	`Restart Pending: OSPF LDP VPN`
	
	This indicates that **OSPF**, **LDP**, and **VPN** protocols did not restart for the **LDP.inet.0** routing table.
	• `vpls_1.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)`
	`Restart Complete`
	
	This indicates that all protocols have restarted for the **vpls_1.l2vpn.0** routing table.

number destinations	Number of destinations for which there are routes in the routing table.

Table 105: show route table Output Fields (continued)

Field Name	Field Description
number routes	Number of routes in the routing table and total number of routes in the following states: • **active** (routes that are active) • **holddown** (routes that are in the pending state before being declared inactive) • **hidden** (routes that are not used because of a routing policy)

route-destination (entry, announced)	Route destination (for example:10.0.0.1/24). The **entry** value is the number of routes for this destination, and the **announced** value is the number of routes being announced for this destination. Sometimes the route destination is presented in another format, such as:
	• ***MPLS-label*** (for example, 80001).
	• ***interface-name*** (for example, ge-1/0/2).
	• ***neighbor-address*:*control-word-status*:*encapsulation type*:*vc-id*:*source*** (Layer 2 circuit only; for example, 10.1.1.195:NoCtrlWord:1:1:Local/96).
	  ◦ ***neighbor-address***—Address of the neighbor.
	  ◦ ***control-word-status***—Whether the use of the control word has been negotiated for this virtual circuit: **NoCtrlWord** or **CtrlWord**.
	  ◦ ***encapsulation type***—Type of encapsulation, represented by a number: (1) Frame Relay DLCI, (2) ATM AAL5 VCC transport, (3) ATM transparent cell transport, (4) Ethernet, (5) VLAN Ethernet, (6) HDLC, (7) PPP, (8) ATM VCC cell transport, (10) ATM VPC cell transport.
	  ◦ ***vc-id***—Virtual circuit identifier.
	  ◦ ***source***—Source of the advertisement: **Local** or **Remote**.
	• ***inclusive multicast Ethernet tag route***—Type of route destination represented by (for example, 3:100.100.100.10:100::0::10::100.100.100.10/384):
	  ◦ *route distinguisher*—(8 octets) Route distinguisher (RD) must be the RD of the EVPN instance (EVI) that is advertising the NLRI.
	  ◦ *Ethernet tag ID*—(4 octets) Identifier of the Ethernet tag. Can set to 0 or to a valid Ethernet tag value.
	  ◦ *IP address length*—(1 octet) Length of IP address in bits.
	  ◦ *originating router's IP address*—(4 or 16 octets) Must set to the provider edge (PE) device's IP address. This address should be common for all EVIs on the PE device, and may be the PE device's loopback address.

label stacking	**(**Next-to-the-last-hop routing device for MPLS only) Depth of the MPLS label stack, where the label-popping operation is needed to remove one or more labels from the top of the stack. A pair of routes is displayed, because the pop operation is performed only when the stack depth is two or more labels. • **S=0 route** indicates that a packet with an incoming label stack depth of 2 or more exits this routing device with one fewer label (the label-popping operation is performed). • If there is no **S=** information, the route is a normal MPLS route, which has a stack depth of 1 (the label-popping operation is not performed).

Table 105: show route table Output Fields (continued)

Field Name	Field Description
[*protocol*, *preference*]	Protocol from which the route was learned and the preference value for the route. • **+**—A plus sign indicates the active route, which is the route installed from the routing table into the forwarding table. • **-** —A hyphen indicates the last active route. • ***—An asterisk indicates that the route is both the active and the last active route. An asterisk before a **to** line indicates the best subpath to the route. In every routing metric except for the BGP **LocalPref** attribute, a lesser value is preferred. In order to use common comparison routines, Junos OS stores the 1's complement of the **LocalPref** value in the **Preference2** field. For example, if the **LocalPref** value for Route 1 is 100, the **Preference2** value is -101. If the **LocalPref** value for Route 2 is 155, the **Preference2** value is -156. Route 2 is preferred because it has a higher **LocalPref** value and a lower **Preference2** value.
Level	(IS-IS only). In IS-IS, a single AS can be divided into smaller groups called areas. Routing between areas is organized hierarchically, allowing a domain to be administratively divided into smaller areas. This organization is accomplished by configuring Level 1 and Level 2 intermediate systems. Level 1 systems route within an area. When the destination is outside an area, they route toward a Level 2 system. Level 2 intermediate systems route between areas and toward other ASs.
Route Distinguisher	IP subnet augmented with a 64-bit prefix.
PMSI	Provider multicast service interface (MVPN routing table).
Next-hop type	Type of next hop. For a description of possible values for this field, see Table 106 on page 1744.
Next-hop reference count	Number of references made to the next hop.
Flood nexthop branches exceed maximum message	Indicates that the number of flood next-hop branches exceeded the system limit of 32 branches, and only a subset of the flood next-hop branches were installed in the kernel.
Source	IP address of the route source.
Next hop	Network layer address of the directly reachable neighboring system.
via	Interface used to reach the next hop. If there is more than one interface available to the next hop, the name of the interface that is actually used is followed by the word **Selected**. This field can also contain the following information: • Weight—Value used to distinguish primary, secondary, and fast reroute backup routes. Weight information is available when MPLS label-switched path (LSP) link protection, node-link protection, or fast reroute is enabled, or when the standby state is enabled for secondary paths. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible. • Balance—Balance coefficient indicating how traffic of unequal cost is distributed among next hops when a routing device is performing unequal-cost load balancing. This information is available when you enable BGP multipath load balancing.
Label-switched-path *lsp-path-name*	Name of the LSP used to reach the next hop.

Table 105: show route table Output Fields (continued)

Field Name	Field Description
Label operation	MPLS label and operation occurring at this routing device. The operation can be **pop** (where a label is removed from the top of the stack), **push** (where another label is added to the label stack), or **swap** (where a label is replaced by another label).
Interface	(Local only) Local interface name.
Protocol next hop	Network layer address of the remote routing device that advertised the prefix. This address is used to derive a forwarding next hop.
Indirect next hop	Index designation used to specify the mapping between protocol next hops, tags, kernel export policy, and the forwarding next hops.
State	State of the route (a route can be in more than one state). See Table 107 on page 1745.
Local AS	AS number of the local routing devices.
Age	How long the route has been known.
AIGP	Accumulated interior gateway protocol (AIGP) BGP attribute.
Metric*n*	Cost value of the indicated route. For routes within an AS, the cost is determined by IGP and the individual protocol metrics. For external routes, destinations, or routing domains, the cost is determined by a preference value.
MED-plus-IGP	Metric value for BGP path selection to which the IGP cost to the next-hop destination has been added.
TTL-Action	For MPLS LSPs, state of the TTL propagation attribute. Can be enabled or disabled for all RSVP-signaled and LDP-signaled LSPs or for specific VRF routing instances.
Task	Name of the protocol that has added the route.
Announcement bits	The number of BGP peers or protocols to which Junos OS has announced this route, followed by the list of the recipients of the announcement. Junos OS can also announce the route to the kernel routing table (KRT) for installing the route into the Packet Forwarding Engine, to a resolve tree, a Layer 2 VC, or even a VPN. For example, ***n*-Resolve inet** indicates that the specified route is used for route resolution for next hops found in the routing table. • ***n***—An index used by Juniper Networks customer support only.

Table 105: show route table Output Fields (continued)

Field Name	Field Description
AS path	AS path through which the route was learned. The letters at the end of the AS path indicate the path origin, providing an indication of the state of the route at the point at which the AS path originated: • **I**—IGP. • **E**—EGP. • **Recorded**—The AS path is recorded by the sample process (sampled). • **?**—Incomplete; typically, the AS path was aggregated. When AS path numbers are included in the route, the format is as follows: • **[ ]**—Brackets enclose the number that precedes the AS path. This number represents the number of ASs present in the AS path, when calculated as defined in RFC 4271. This value is used in the AS-path merge process, as defined in RFC 4893. • **[ ]**—If more than one AS number is configured on the routing device, or if AS path prepending is configured, brackets enclose the local AS number associated with the AS path. • **{ }**—Braces enclose AS sets, which are groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order. • **( )**—Parentheses enclose a confederation. • **([ ])**—Parentheses and brackets enclose a confederation set. NOTE: In Junos OS Release 10.3 and later, the AS path field displays an unrecognized attribute and associated hexadecimal value if BGP receives attribute 128 (attribute set) and you have not configured an independent domain in any routing instance.
validation-state	(BGP-learned routes) Validation status of the route: • **Invalid**—Indicates that the prefix is found, but either the corresponding AS received from the EBGP peer is not the AS that appears in the database, or the prefix length in the BGP update message is longer than the maximum length permitted in the database. • **Unknown**—Indicates that the prefix is not among the prefixes or prefix ranges in the database. • **Unverified**—Indicates that the origin of the prefix is not verified against the database. This is because the database got populated and the validation is not called for in the BGP import policy, although origin validation is enabled, or the origin validation is not enabled for the BGP peers. • **Valid**—Indicates that the prefix and autonomous system pair are found in the database.
FECs bound to route	Indicates point-to-multipoint root address, multicast source address, and multicast group address when multipoint LDP (M-LDP) inband signaling is configured.
Primary Upstream	When multipoint LDP with multicast-only fast reroute (MoFRR) is configured, indicates the primary upstream path. MoFRR transmits a multicast join message from a receiver toward a source on a primary path, while also transmitting a secondary multicast join message from the receiver toward the source on a backup path.
RPF Nexthops	When multipoint LDP with MoFRR is configured, indicates the reverse-path forwarding (RPF) next-hop information. Data packets are received from both the primary path and the secondary paths. The redundant packets are discarded at topology merge points due to the RPF checks.
Label	Multiple MPLS labels are used to control MoFRR stream selection. Each label represents a separate route, but each references the same interface list check. Only the primary label is forwarded while all others are dropped. Multiple interfaces can receive packets using the same label.

Table 105: show route table Output Fields (continued)

Field Name	Field Description
weight	Value used to distinguish MoFRR primary and backup routes. A lower weight value is preferred. Among routes with the same weight value, load balancing is possible.
VC Label	MPLS label assigned to the Layer 2 circuit virtual connection.
MTU	Maximum transmission unit (MTU) of the Layer 2 circuit.
VLAN ID	VLAN identifier of the Layer 2 circuit.
Prefixes bound to route	Forwarding equivalent class (FEC) bound to this route. Applicable only to routes installed by LDP.
Communities	Community path attribute for the route. See Table 108 on page 1747 for all possible values for this field.
Layer2-info: encaps	Layer 2 encapsulation (for example, VPLS).
control flags	Control flags: **none** or **Site Down**.
mtu	Maximum transmission unit (MTU) information.
Label-Base, range	First label in a block of labels and label block size. A remote PE routing device uses this first label when sending traffic toward the advertising PE routing device.
status vector	Layer 2 VPN and VPLS network layer reachability information (NLRI).
Accepted Multipath	Current active path when BGP multipath is configured.
Accepted LongLivedStale	The LongLivedStale flag indicates that the route was marked LLGR-stale by this router, as part of the operation of LLGR receiver mode. Either this flag or the LongLivedStaleImport flag might be displayed for a route. Neither of these flags is displayed at the same time as the Stale (ordinary GR stale) flag.
Accepted LongLivedStaleImport	The LongLivedStaleImport flag indicates that the route was marked LLGR-stale when it was received from a peer, or by import policy. Either this flag or the LongLivedStale flag might be displayed for a route. Neither of these flags is displayed at the same time as the Stale (ordinary GR stale) flag. Accept all received BGP long-lived graceful restart (LLGR) and LLGR stale routes learned from configured neighbors and import into the inet.0 routing table
ImportAccepted LongLivedStaleImport	Accept all received BGP long-lived graceful restart (LLGR) and LLGR stale routes learned from configured neighbors and imported into the inet.0 routing table The LongLivedStaleImport flag indicates that the route was marked LLGR-stale when it was received from a peer, or by import policy.
Accepted MultipathContrib	Path currently contributing to BGP multipath.
Localpref	Local preference value included in the route.
Router ID	BGP router ID as advertised by the neighbor in the open message.

Table 105: show route table Output Fields (continued)

Field Name	Field Description
Primary Routing Table	In a routing table group, the name of the primary routing table in which the route resides.
Secondary Tables	In a routing table group, the name of one or more secondary tables in which the route resides.

Table 106 on page 1744 describes all possible values for the Next-hop Types output field.

Table 106: Next-hop Types Output Field Values

Next-Hop Type	Description
Broadcast (bcast)	Broadcast next hop.
Deny	Deny next hop.
Discard	Discard next hop.
Flood	Flood next hop. Consists of components called branches, up to a maximum of 32 branches. Each flood next-hop branch sends a copy of the traffic to the forwarding interface. Used by point-to-multipoint RSVP, point-to-multipoint LDP, point-to-multipoint CCC, and multicast.
Hold	Next hop is waiting to be resolved into a unicast or multicast type.
Indexed (idxd)	Indexed next hop.
Indirect (indr)	Used with applications that have a protocol next hop address that is remote. You are likely to see this next-hop type for internal BGP (IBGP) routes when the BGP next hop is a BGP neighbor that is not directly connected.
Interface	Used for a network address assigned to an interface. Unlike the router next hop, the interface next hop does not reference any specific node on the network.
Local (locl)	Local address on an interface. This next-hop type causes packets with this destination address to be received locally.
Multicast (mcst)	Wire multicast next hop (limited to the LAN).
Multicast discard (mdsc)	Multicast discard.
Multicast group (mgrp)	Multicast group member.
Receive (recv)	Receive.
Reject (rjct)	Discard. An ICMP unreachable message was sent.

Table 106: Next-hop Types Output Field Values (continued)

Next-Hop Type	Description
Resolve (rslv)	Resolving next hop.
Routed multicast (mcrt)	Regular multicast next hop.
Router	A specific node or set of nodes to which the routing device forwards packets that match the route prefix. To qualify as a next-hop type router, the route must meet the following criteria: • Must not be a direct or local subnet for the routing device. • Must have a next hop that is directly connected to the routing device.
Table	Routing table next hop.
Unicast (ucst)	Unicast.
Unilist (ulst)	List of unicast next hops. A packet sent to this next hop goes to any next hop in the list.

Table 107 on page 1745 describes all possible values for the State output field. A route can be in more than one state (for example, **<Active NoReadvrt Int Ext>**).

Table 107: State Output Field Values

Value	Description
Accounting	Route needs accounting.
Active	Route is active.
Always Compare MED	Path with a lower multiple exit discriminator (MED) is available.
AS path	Shorter AS path is available.
Cisco Non-deterministic MED selection	Cisco nondeterministic MED is enabled, and a path with a lower MED is available.
Clone	Route is a clone.
Cluster list length	Length of cluster list sent by the route reflector.
Delete	Route has been deleted.
Ex	Exterior route.

Table 107: State Output Field Values (continued)

Value	Description
Ext	BGP route received from an external BGP neighbor.
FlashAll	Forces all protocols to be notified of a change to any route, active or inactive, for a prefix. When not set, protocols are informed of a prefix only when the active route changes.
Hidden	Route not used because of routing policy.
IfCheck	Route needs forwarding RPF check.
IGP metric	Path through next hop with lower IGP metric is available.
Inactive reason	Flags for this route, which was not selected as best for a particular destination.
Initial	Route being added.
Int	Interior route.
Int Ext	BGP route received from an internal BGP peer or a BGP confederation peer.
Interior > Exterior > Exterior via Interior	Direct, static, IGP, or EBGP path is available.
Local Preference	Path with a higher local preference value is available.
Martian	Route is a martian (ignored because it is obviously invalid).
MartianOK	Route exempt from martian filtering.
Next hop address	Path with lower metric next hop is available.
No difference	Path from neighbor with lower IP address is available.
NoReadvrt	Route not to be advertised.
NotBest	Route not chosen because it does not have the lowest MED.
Not Best in its group	Incoming BGP AS is not the best of a group (only one AS can be the best).
NotInstall	Route not to be installed in the forwarding table.
Number of gateways	Path with a greater number of next hops is available.
Origin	Path with a lower origin code is available.

Table 107: State Output Field Values (continued)

Value	Description
Pending	Route pending because of a hold-down configured on another route.
Release	Route scheduled for release.
RIB preference	Route from a higher-numbered routing table is available.
Route Distinguisher	64-bit prefix added to IP subnets to make them unique.
Route Metric or MED comparison	Route with a lower metric or MED is available.
Route Preference	Route with lower preference value is available.
Router ID	Path through a neighbor with lower ID is available.
Secondary	Route not a primary route.
Unusable path	Path is not usable because of one of the following conditions: • The route is damped. • The route is rejected by an import policy. • The route is unresolved.
Update source	Last tiebreaker is the lowest IP address value.

Table 108 on page 1747 describes the possible values for the Communities output field.

Table 108: Communities Output Field Values

Value	Description
area-number	4 bytes, encoding a 32-bit area number. For AS-external routes, the value is **0**. A nonzero value identifies the route as internal to the OSPF domain, and as within the identified area. Area numbers are relative to a particular OSPF domain.
bandwidth: ***local AS number:link-bandwidth-number***	Link-bandwidth community value used for unequal-cost load balancing. When BGP has several candidate paths available for multipath purposes, it does not perform unequal-cost load balancing according to the link-bandwidth community unless all candidate paths have this attribute.
domain-id	Unique configurable number that identifies the OSPF domain.
domain-id-vendor	Unique configurable number that further identifies the OSPF domain.
link-bandwidth-number	Link-bandwidth number: from **0** through **4,294,967,295** (bytes per second).
local AS number	Local AS number: from **1** through **65,535**.

Table 108: Communities Output Field Values (continued)

Value	Description
options	1 byte. Currently this is only used if the route type is **5** or **7**. Setting the least significant bit in the field indicates that the route carries a type 2 metric.
origin	(Used with VPNs) Identifies where the route came from.
ospf-route-type	1 byte, encoded as **1** or **2** for intra-area routes (depending on whether the route came from a type 1 or a type 2 LSA); **3** for summary routes; **5** for external routes (area number must be**0**); **7** for NSSA routes; or **129** for sham link endpoint addresses.
route-type-vendor	Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute **0x8000**. The format is ***area-number:ospf-route-type:options***.
rte-type	Displays the area number, OSPF route type, and option of the route. This is configured using the BGP extended community attribute **0x0306**. The format is ***area-number:ospf-route-type:options***.
target	Defines which VPN the route participates in; **target** has the format ***32-bit IP address:16-bit number***. For example, 10.19.0.0:100.
unknown IANA	Incoming IANA codes with a value between **0x1** and **0x7fff**. This code of the BGP extended community attribute is accepted, but it is not recognized.
unknown OSPF vendor community	Incoming IANA codes with a value above **0x8000**. This code of the BGP extended community attribute is accepted, but it is not recognized.

Sample Output

show route table bgp.l2.vpn

```
user@host> show route table bgp.l2.vpn
bgp.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.24.1:1:4:1/96
                   *[BGP/170] 01:08:58, localpref 100, from 192.168.24.1
                      AS path: I
                    > to 10.0.16.2 via fe-0/0/1.0, label-switched-path am
```

show route table bgp.l3vpn.0

```
user@host> show route table bgp.l3vpn.0
      bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
      + = Active Route, - = Last Active, * = Both

      10.255.71.15:100:10.255.71.17/32
                         *[BGP/170] 00:03:59, MED 1, localpref 100, from
      10.255.71.15
                            AS path: I
                          > via so-2/1/0.0, Push 100020, Push 100011(top)
      10.255.71.15:200:10.255.71.18/32
```

```
                                *[BGP/170] 00:03:59, MED 1, localpref 100, from
      10.255.71.15
                                  AS path: I
                                > via so-2/1/0.0, Push 100021, Push 100011(top)
```

show route table bgp.l3vpn.0 detail

```
user@host> show route table bgp.l3vpn.0 detail
bgp.l3vpn.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

10.255.245.12:1:172.16.4.0/8 (1 entry, 1 announced)
  *BGP   Preference: 170/-101
         Route Distinguisher: 10.255.245.12:1
         Source: 10.255.245.12
         Next hop: 192.168.208.66 via fe-0/0/0.0, selected
         Label operation: Push 182449
         Protocol next hop: 10.255.245.12
         Push 182449
         Indirect next hop: 863a630 297
         State: <Active Int Ext>
         Local AS:    35 Peer AS:    35
         Age: 12:19      Metric2: 1
         Task: BGP_35.10.255.245.12+179
         Announcement bits (1): 0-BGP.0.0.0.0+179
         AS path: 30 10458 14203 2914 3356 I (Atomic) Aggregator: 3356 4.68.0.11

         Communities: 2914:420 target:11111:1 origin:56:78
         VPN Label: 182449
         Localpref: 100
          Router ID: 10.255.245.12

10.255.245.12:1:4.17.225.0/24 (1 entry, 1 announced)
  *BGP   Preference: 170/-101
         Route Distinguisher: 10.255.245.12:1
         Source: 10.255.245.12
         Next hop: 192.168.208.66 via fe-0/0/0.0, selected
         Label operation: Push 182465
         Protocol next hop: 10.255.245.12
         Push 182465
         Indirect next hop: 863a8f0 305
         State: <Active Int Ext>
         Local AS:    35 Peer AS:    35
         Age: 12:19      Metric2: 1
         Task: BGP_35.10.255.245.12+179
         Announcement bits (1): 0-BGP.0.0.0.0+179
AS path: 30 10458 14203 2914 11853 11853 11853 6496 6496 6496 6496 6496 6496 I
                Communities: 2914:410 target:12:34 target:11111:1 origin:12:34
                VPN Label: 182465
                Localpref: 100
                Router ID: 10.255.245.12

10.255.245.12:1:4.17.226.0/23 (1 entry, 1 announced)
  *BGP   Preference: 170/-101
         Route Distinguisher: 10.255.245.12:1
         Source: 10.255.245.12
         Next hop: 192.168.208.66 via fe-0/0/0.0, selected
         Label operation: Push 182465
         Protocol next hop: 10.255.245.12
         Push 182465
         Indirect next hop: 86bd210 330
         State: <Active Int Ext>
```

```
                Local AS:    35 Peer AS:    35
                Age: 12:19      Metric2: 1
                Task: BGP_35.10.255.245.12+179
                Announcement bits (1): 0-BGP.0.0.0.0+179
                AS path: 30 10458 14203 2914 11853 11853 11853 6496 6496 6496 6496 6496

                6496 I
                Communities: 2914:410 target:12:34 target:11111:1 origin:12:34
                VPN Label: 182465
                Localpref: 100
                Router ID: 10.255.245.12

10.255.245.12:1:4.17.251.0/24 (1 entry, 1 announced)
  *BGP   Preference: 170/-101
                Route Distinguisher: 10.255.245.12:1
                Source: 10.255.245.12
                Next hop: 192.168.208.66 via fe-0/0/0.0, selected
                Label operation: Push 182465
                Protocol next hop: 10.255.245.12
                Push 182465
                Indirect next hop: 86bd210 330
                State: <Active Int Ext>
                Local AS:    35 Peer AS:    35
                Age: 12:19      Metric2: 1
                Task: BGP_35.10.255.245.12+179
                Announcement bits (1): 0-BGP.0.0.0.0+179
                AS path: 30 10458 14203 2914 11853 11853 11853 6496 6496 6496 6496 6496

                6496 I
                Communities: 2914:410 target:12:34 target:11111:1 origin:12:34
                VPN Label: 182465
                Localpref: 100
```

show route table bgp.rtarget.0 (When Proxy BGP Route Target Filtering Is Configured)

```
user@host> show route table bgp.rtarget.o
bgp.rtarget.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

100:100:100/96
                   *[RTarget/5] 00:03:14
                      Type Proxy
                        for 10.255.165.103
                        for 10.255.166.124
                      Local
```

show route table bgp.evpn.0

```
user@host> show route table bgp.evpn.0
bgp.evpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

2:100.100.100.2:100::0::00:26:88:5f:67:b0/304
                   *[BGP/170] 11:00:05, localpref 100, from 100.100.100.2
                      AS path: I, validation-state: unverified
                    > to 100.64.12.2 via xe-2/2/0.0, label-switched-path R0toR1
2:100.100.100.2:100::0::00:51:51:51:51:51/304
                   *[BGP/170] 11:00:05, localpref 100, from 100.100.100.2
                      AS path: I, validation-state: unverified
                    > to 100.64.12.2 via xe-2/2/0.0, label-switched-path R0toR1
```

```
2:100.100.100.3:100::0::00:52:52:52:52:52/304
                   *[BGP/170] 10:59:58, localpref 100, from 100.100.100.3
                      AS path: I, validation-state: unverified
                    > to 100.64.13.3 via ge-2/0/8.0, label-switched-path R0toR2
2:100.100.100.3:100::0::a8:d0:e5:5b:01:c8/304
                   *[BGP/170] 10:59:58, localpref 100, from 100.100.100.3
                      AS path: I, validation-state: unverified
                    > to 100.64.13.3 via ge-2/0/8.0, label-switched-path R0toR2
3:100.100.100.2:100::1000::100.100.100.2/304
                   *[BGP/170] 11:00:16, localpref 100, from 100.100.100.2
                      AS path: I, validation-state: unverified
                    > to 100.64.12.2 via xe-2/2/0.0, label-switched-path R0toR1
3:100.100.100.2:100::2000::100.100.100.2/304
                   *[BGP/170] 11:00:16, localpref 100, from 100.100.100.2
                      AS path: I, validation-state: unverified
                    > to 100.64.12.2 via xe-2/2/0.0, label-switched-path R0toR1
```

show route table evpna.evpn.0

```
user@host> show route table evpna.evpn.0
evpna.evpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

3:100.100.100.10:100::0::10::100.100.100.10/384
                   *[EVPN/170] 01:37:09
                      Indirect
3:100.100.100.2:100::2000::100.100.100.2/304
                   *[EVPN/170] 01:37:12
                      Indirect
```

show route table inet.0

```
user@host> show route table inet.0
inet.0: 12 destinations, 12 routes (11 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:51:57
                    > to 172.16.5.254 via fxp0.0
10.0.0.1/32         *[Direct/0] 00:51:58
                    > via at-5/3/0.0
10.0.0.2/32         *[Local/0] 00:51:58
                      Local
10.12.12.21/32     *[Local/0] 00:51:57
                      Reject
10.13.13.13/32     *[Direct/0] 00:51:58
                    > via t3-5/2/1.0
10.13.13.14/32     *[Local/0] 00:51:58
                      Local
10.13.13.21/32     *[Local/0] 00:51:58
                      Local
10.13.13.22/32     *[Direct/0] 00:33:59
                    > via t3-5/2/0.0
127.0.0.1/32        [Direct/0] 00:51:58
                    > via lo0.0
10.222.5.0/24     *[Direct/0] 00:51:58
                    > via fxp0.0
10.222.5.81/32    *[Local/0] 00:51:58
                      Local
```

show route table inet.3

```
user@host> show route table inet.3
inet.3: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.5/32        *[LDP/9] 00:25:43, metric 10, tag 200
                      to 10.2.94.2 via lt-1/2/0.49
                    > to 10.2.3.2 via lt-1/2/0.23
```

show route table inet.3 protocol ospf

```
user@host> show route table inet.3 protocol ospf
inet.3: 9 destinations, 18 routes (9 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.20/32         [L-OSPF/10] 1d 00:00:56, metric 2
                    > to 10.0.10.70 via lt-1/2/0.14, Push 800020
                     to 10.0.6.60 via lt-1/2/0.12, Push 800020, Push 800030(top)
1.1.1.30/32         [L-OSPF/10] 1d 00:01:01, metric 3
                    > to 10.0.10.70 via lt-1/2/0.14, Push 800030
                      to 10.0.6.60 via lt-1/2/0.12, Push 800030
1.1.1.40/32         [L-OSPF/10] 1d 00:01:01, metric 4
                    > to 10.0.10.70 via lt-1/2/0.14, Push 800040
                      to 10.0.6.60 via lt-1/2/0.12, Push 800040
1.1.1.50/32         [L-OSPF/10] 1d 00:01:01, metric 5
                    > to 10.0.10.70 via lt-1/2/0.14, Push 800050
                      to 10.0.6.60 via lt-1/2/0.12, Push 800050
1.1.1.60/32         [L-OSPF/10] 1d 00:01:01, metric 6
                    > to 10.0.10.70 via lt-1/2/0.14, Push 800060
                      to 10.0.6.60 via lt-1/2/0.12, Pop
```

show route table inet6.0

```
user@host> show route table inet6.0
inet6.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Route, * = Both

fec0:0:0:3::/64 *[Direct/0] 00:01:34
>via fe-0/1/0.0

fec0:0:0:3::/128 *[Local/0] 00:01:34
>Local

fec0:0:0:4::/64 *[Static/5] 00:01:34
>to fec0:0:0:3::ffff via fe-0/1/0.0
```

show route table inet6.3

```
user@router> show route table inet6.3
inet6.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

::10.255.245.195/128
                   *[LDP/9] 00:00:22, metric 1
                    > via so-1/0/0.0
::10.255.245.196/128
                   *[LDP/9] 00:00:08, metric 1
                    > via so-1/0/0.0, Push 100008
```

show route table inetflow detail

```
user@host> show route table inetflow detail
inetflow.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.12.44.1,*/48 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next-hop reference count: 2
                State: <Active Ext>
                Local AS: 64502 Peer AS: 64500
                Age: 4
                Task: BGP_64500.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 64500 I
                Communities: traffic-rate:0:0
                Validation state: Accept, Originator: 10.12.99.5
                Via: 10.12.44.0/24, Active
                Localpref: 100
                Router ID: 10.255.71.161

10.12.56.1,*/48 (1 entry, 1 announced)
        *Flow   Preference: 5
                Next-hop reference count: 2
                State: <Active>
                Local AS: 64502
                Age: 6:30
                Task: RT Flow
                Announcement bits (2): 0-Flow 1-BGP.0.0.0.0+179
                AS path: I
                Communities: 1:1
```

show route table lsdist.0 extensive

```
user@host> show route table lsdist.0 extensive
lsdist.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
NODE { AS:4170512532 BGP-LS ID:4170512532 ISO:3245.3412.3456.00 ISIS-L1:0 }/1152
 (1 entry, 1 announced)
TSI:
Page 0 idx 0, (group ibgp type Internal) Type 1 val 0xa62f378 (adv_entry)
   Advertised metrics:
     Nexthop: Self
     Localpref: 100
     AS path: [4170512532] I
     Communities:
Path NODE { AS:4170512532 BGP-LS ID:4170512532 ISO:3245.3412.3456.00 ISIS-L1:0 }
 Vector len 4.  Val: 0
        *IS-IS  Preference: 15
                Level: 1
                Next hop type: Fictitious, Next hop index: 0
                Address: 0x95dfc64
                Next-hop reference count: 9
                State: <Active NotInstall>
                Local AS: 4170512532
                Age: 6:05
                Validation State: unverified
                Task: IS-IS
                Announcement bits (1): 0-BGP_RT_Background
                AS path: I
                IPv4 Router-ids:
                   128.220.11.197
                Area membership:
```

```
                47 00 05 80 ff f8 00 00 00 01 08 00 01
             SPRING-Capabilities:                 - SRGB block [Start: 800000,
 Range: 256, Flags: 0xc0]
              SPRING-Algorithms:
                 - Algo: 0
 LINK { Local { AS:4170512532 BGP-LS ID:4170512532 ISO:3245.3412.3456.00 }.{
IPv4:8.65.1.105 } Remote { AS:4170512532 BGP-LS ID:4170512532 ISO:4284.3300.5067)
TSI:
Page 0 idx 0, (group ibgp type Internal) Type 1 val 0xa62f3cc (adv_entry)
    Advertised metrics:
      Nexthop: Self
      Localpref: 100
      AS path: [4170512532] I
      Communities:
Path LINK { Local { AS:4170512532 BGP-LS ID:4170512532 ISO:3245.3412.3456.00 }.{
 IPv4:8.65.1.105 } Remote { AS:4170512532 BGP-LS ID:4170512532 ISO:4284.33000
      *IS-IS  Preference: 15
               Level: 1
               Next hop type: Fictitious, Next hop index: 0
               Address: 0x95dfc64
               Next-hop reference count: 9
               State: <Active NotInstall>
               Local AS: 4170512532
               Age: 6:05
               Validation State: unverified
               Task: IS-IS
               Announcement bits (1): 0-BGP_RT_Background
               AS path: I
               Color: 32768
               Maximum bandwidth: 1000Mbps
               Reservable bandwidth: 1000Mbps
               Unreserved bandwidth by priority:
                  0   1000Mbps
                  1   1000Mbps
                  2   1000Mbps
                  3   1000Mbps
                  4   1000Mbps
                  5   1000Mbps
                  6   1000Mbps
                  7   1000Mbps
                Metric: 10
                TE Metric: 10
                LAN IPV4 Adj-SID -  Label: 299776, Flags: 0x30,
Weight: 0, Nbr: 10.220.1.83

PREFIX { Node { AS:4170512532 BGP-LS ID:4170512532 ISO:3245.3412.3456.00 } {
IPv4:128.220.11.197/32 } ISIS-L1:0 }/1152 (1 entry, 1 announced) TSI: Page 0 idx
 0, (group ibgp type Internal) Type 1 val 0xa62f43c (adv_entry)
    Advertised metrics:
      Nexthop: Self
      Localpref: 100
      AS path: [4170512532] I
      Communities:
Path PREFIX { Node { AS:4170512532 BGP-LS ID:4170512532 ISO:3245.3412.3456.00 }
{ IPv4:128.220.11.197/32 } ISIS-L1:0 } Vector len 4.  Val: 0
        *IS-IS  Preference: 15
                Level: 1
                Next hop type: Fictitious, Next hop index: 0
                Address: 0x95dfc64
                Next-hop reference count: 9
                State:<Active NotInstall>
```

```
                    Local AS: 4170512532
                    Age: 6:05
                    Validation State: unverified
                    Task: IS-IS
                    Announcement bits (1): 0-BGP_RT_Background
                    AS path: I
                                 Prefix SID: 67, Flags: 0x40, Algo: 0
```

show route table l2circuit.0

```
user@host> show route table l2circuit.0
l2circuit.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.195:NoCtrlWord:1:1:Local/96
                   *[L2CKT/7] 00:50:47
                    > via so-0/1/2.0, Push 100049
                      via so-0/1/3.0, Push 100049
10.1.1.195:NoCtrlWord:1:1:Remote/96
                   *[LDP/9] 00:50:14
                      Discard
10.1.1.195:CtrlWord:1:2:Local/96
                   *[L2CKT/7] 00:50:47
                    > via so-0/1/2.0, Push 100049
                      via so-0/1/3.0, Push 100049
10.1.1.195:CtrlWord:1:2:Remote/96
                   *[LDP/9] 00:50:14
                      Discard
```

show route table mpls

```
user@host> show route table mpls
mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:13:55, metric 1
                      Receive
1                  *[MPLS/0] 00:13:55, metric 1
                      Receive
2                  *[MPLS/0] 00:13:55, metric 1
                      Receive
1024               *[VPN/0] 00:04:18
                      to table red.inet.0, Pop
```

show route table mpls extensive

```
user@host> show route table mpls extensive
100000 (1 entry, 1 announced)
TSI:
KRT in-kernel 100000 /36 -> {so-1/0/0.0}
        *LDP    Preference:   9
                Next hop:  via so-1/0/0.0, selected
                Pop
                State: <Active Int>
                Age: 29:50      Metric: 1
                Task: LDP
                Announcement bits (1): 0-KRT
                AS path: I
                Prefixes bound to route: 10.0.0.194/32
```

show route table mpls.0

```
user@host> show route table mpls.0
mpls.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 11:39:56, metric 1
                      to table inet.0
0(S=0)             *[MPLS/0] 11:39:56, metric 1
                      to table mpls.0
1                  *[MPLS/0] 11:39:56, metric 1
                      Receive
2                  *[MPLS/0] 11:39:56, metric 1
                      to table inet6.0
2(S=0)             *[MPLS/0] 11:39:56, metric 1
                      to table mpls.0
13                 *[MPLS/0] 11:39:56, metric 1
                      Receive
303168             *[EVPN/7] 11:00:49, routing-instance pbbn10, route-type
Ingress-MAC, ISID 0
                      to table pbbn10.evpn-mac.0
303184             *[EVPN/7] 11:00:53, routing-instance pbbn10, route-type
Ingress-IM, ISID 1000
                      to table pbbn10.evpn-mac.0
                    [EVPN/7] 11:00:53, routing-instance pbbn10, route-type
Ingress-IM, ISID 2000
                      to table pbbn10.evpn-mac.0
303264             *[EVPN/7] 11:00:53, remote-pe 100.100.100.2, routing-instance
 pbbn10, route-type Egress-IM, ISID 1000
                     > to 100.1.12.2 via xe-2/2/0.0, label-switched-path R0toR1
303280             *[EVPN/7] 11:00:53, remote-pe 100.100.100.2, routing-instance
 pbbn10, route-type Egress-IM, ISID 2000
                     > to 100.1.12.2 via xe-2/2/0.0, label-switched-path R0toR1
303328             *[EVPN/7] 11:00:49, remote-pe 100.100.100.2, routing-instance
 pbbn10, route-type Egress-MAC, ISID 0
                     > to 100.1.12.2 via xe-2/2/0.0, label-switched-path R0toR1
303344             *[EVPN/7] 11:00:49, remote-pe 100.100.100.2, routing-instance
 pbbn10, route-type Egress-MAC, ISID 0
                     > to 100.1.12.2 via xe-2/2/0.0, label-switched-path R0toR1
303360             *[EVPN/7] 11:00:47, routing-instance pbbn10, route-type
Egress-MAC, ISID 0, BMAC 00:26:88:5f:67:b0
                     > to 100.1.12.2 via xe-2/2/0.0, label-switched-path R0toR1
303376             *[EVPN/7] 11:00:47, routing-instance pbbn10, route-type
Egress-MAC, ISID 0, BMAC 00:51:51:51:51:51
                     > to 100.1.12.2 via xe-2/2/0.0, label-switched-path R0toR1
303392             *[EVPN/7] 11:00:35, remote-pe 100.100.100.3, routing-instance
 pbbn10, route-type Egress-MAC, ISID 0
                     > to 100.1.13.3 via ge-2/0/8.0, label-switched-path R0toR2
303408             *[EVPN/7] 11:00:35, remote-pe 100.100.100.3, routing-instance
 pbbn10, route-type Egress-MAC, ISID 0
                     > to 100.1.13.3 via ge-2/0/8.0, label-switched-path R0toR2
303424             *[EVPN/7] 11:00:33, routing-instance pbbn10, route-type
Egress-MAC, ISID 0, BMAC a8:d0:e5:5b:01:c8
                     > to 100.1.13.3 via ge-2/0/8.0, label-switched-path R0toR2
303440             *[EVPN/7] 11:00:33, routing-instance pbbn10, route-type
Egress-MAC, ISID 0, BMAC 00:52:52:52:52:52
                     > to 100.1.13.3 via ge-2/0/8.0, label-switched-path R0toR2
```

show route table mpls.0 detail (PTX Series)

```
user@host> show route table mpls.0 detail
ge-0/0/2.600 (1 entry, 1 announced)
        *L2VPN  Preference: 7
                Next hop type: Indirect
                Address: 0x9438f34
                Next-hop reference count: 2
                Next hop type: Router, Next hop index: 567
                Next hop: 10.0.0.1 via ge-0/0/1.0, selected
                Label operation: Push 299808
                Label TTL action: prop-ttl
                Load balance label: Label 299808:None;
                Session Id: 0x1
                Protocol next hop: 10.255.255.1
                Label operation: Push 299872 Offset: 252
                Label TTL action: no-prop-ttl
                Load balance label: Label 299872:Flow label PUSH;
                Composite next hop: 0x9438ed8 570 INH Session ID: 0x2
                Indirect next hop: 0x9448208 262142 INH Session ID: 0x2
                State: <Active Int>
                Age: 21         Metric2: 1
                Validation State: unverified
                Task: Common L2 VC
                Announcement bits (2): 0-KRT 2-Common L2 VC
                AS path: I
```

show route table mpls.0 ccc ge-0/0/1.1004 detail

```
user@host>show route table mpls.0 ccc ge-0/0/1.1004 detail
mpls.0: 121 destinations, 121 routes (121 active, 0 holddown, 0 hidden)
ge-0/0/1.1004 (1 entry, 1 announced)
        *EVPN   Preference: 7
                Next hop type: List, Next hop index: 1048577
                Address: 0xdc14770
                Next-hop reference count: 3
                Next hop: ELNH Address 0xd011e30
                    Next hop type: Indirect, Next hop index: 0
                    Address: 0xd011e30
                    Next-hop reference count: 3
                    Protocol next hop: 100.100.100.1
                    Label operation: Push 301952
                    Composite next hop: 0xd011dc0 754 INH Session ID: 0x146
                    Indirect next hop: 0xb69a890 1048615 INH Session ID: 0x146
                        Next hop type: Router, Next hop index: 735
                        Address: 0xd00e530
                        Next-hop reference count: 23
                        Next hop: 100.46.1.2 via ge-0/0/5.0
                        Label-switched-path pe4_to_pe1
                        Label operation: Push 300320
                        Label TTL action: prop-ttl
                        Load balance label: Label 300320: None;
                        Label element ptr: 0xd00e580
                        Label parent element ptr: 0x0
                        Label element references: 18
                        Label element child references: 16
                        Label element lsp id: 5
                Next hop: ELNH Address 0xd012070
                    Next hop type: Indirect, Next hop index: 0
                    Address: 0xd012070
```

```
                    Next-hop reference count: 3
                    Protocol next hop: 100.100.100.2
                    Label operation: Push 301888
                    Composite next hop: 0xd012000 755 INH Session ID: 0x143
                    Indirect next hop: 0xb69a9a0 1048641 INH Session ID: 0x143
                        Next hop type: Router, Next hop index: 716
                        Address: 0xd00e710
                        Next-hop reference count: 23
                        Next hop: 100.46.1.2 via ge-0/0/5.0
                        Label-switched-path pe4_to_pe2
                        Label operation: Push 300304
                        Label TTL action: prop-ttl
                        Load balance label: Label 300304: None;
                        Label element ptr: 0xd00e760
                        Label parent element ptr: 0x0
                        Label element references: 15
                        Label element child references: 13
                        Label element lsp id: 6
                Next hop: ELNH Address 0xd0121f0, selected
                    Next hop type: Indirect, Next hop index: 0
                    Address: 0xd0121f0
                    Next-hop reference count: 3
                    Protocol next hop: 100.100.100.3
                    Label operation: Push 301984
                    Composite next hop: 0xd012180 756 INH Session ID: 0x145
                    Indirect next hop: 0xb69aab0 1048642 INH Session ID: 0x145
                        Next hop type: Router, Next hop index: 801
                        Address: 0xd010ed0
                        Next-hop reference count: 32
                        Next hop: 100.46.1.2 via ge-0/0/5.0
                        Label-switched-path pe4_to_pe3
                        Label operation: Push 300336
                        Label TTL action: prop-ttl
                        Load balance label: Label 300336: None;
                        Label element ptr: 0xd0108c0
                        Label parent element ptr: 0x0
                        Label element references: 22
                        Label element child references: 20
                        Label element lsp id: 7
                State: < Active Int >
                Age: 2:06:50
                Validation State: unverified
                Task: evpn global task
                Announcement bits (1): 1-KRT
                AS path: I
```

show route table mpls.0 protocol evpn

```
user@host>show route table mpls.0 protocol evpn
mpls.0: 121 destinations, 121 routes (121 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299872             *[EVPN/7] 02:30:58, routing-instance mhevpn, route-type
Ingress-IM, vlan-id 10
                      to table mhevpn.evpn-mac.0
300016             *[EVPN/7] 02:30:38, routing-instance VS-1, route-type
Ingress-IM, vlan-id 110
                      to table VS-1.evpn-mac.0
300032             *[EVPN/7] 02:30:38, routing-instance VS-1, route-type
Ingress-IM, vlan-id 120
                      to table VS-1.evpn-mac.0
```

```
300048             *[EVPN/7] 02:30:38, routing-instance VS-1, route-type
Ingress-IM, vlan-id 130
                      to table VS-1.evpn-mac.0
300064             *[EVPN/7] 02:30:38, routing-instance VS-2, route-type
Ingress-IM, vlan-id 210
                      to table VS-2.evpn-mac.0
300080             *[EVPN/7] 02:30:38, routing-instance VS-2, route-type
Ingress-IM, vlan-id 220
                      to table VS-2.evpn-mac.0
300096             *[EVPN/7] 02:30:38, routing-instance VS-2, route-type
Ingress-IM, vlan-id 230
                      to table VS-2.evpn-mac.0
300112             *[EVPN/7] 02:27:06, routing-instance mhevpn, route-type
Egress-MAC, ESI 00:44:44:44:44:44:44:44:44:44
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
300128             *[EVPN/7] 02:29:22, routing-instance mhevpn, route-type
Ingress-Aliasing
                      to table mhevpn.evpn-mac.0
300144             *[EVPN/7] 02:27:06, routing-instance VS-1, route-type
Egress-MAC, ESI 00:44:44:44:44:44:44:44:44:44
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
300160             *[EVPN/7] 02:29:22, routing-instance VS-1, route-type
Ingress-Aliasing
                      to table VS-1.evpn-mac.0
300176             *[EVPN/7] 02:27:07, routing-instance VS-2, route-type
Egress-MAC, ESI 00:44:44:44:44:44:44:44:44:44
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
300192             *[EVPN/7] 02:29:22, routing-instance VS-2, route-type
Ingress-Aliasing
                      to table VS-2.evpn-mac.0
300208             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-1, route-type Egress-IM, vlan-id 120
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300224             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 mhevpn, route-type Egress-IM, vlan-id 10
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300240             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-1, route-type Egress-IM, vlan-id 110
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300256             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-1, route-type Egress-IM, vlan-id 130
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300272             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-2, route-type Egress-IM, vlan-id 210
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300288             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-2, route-type Egress-IM, vlan-id 220
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300304             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-2, route-type Egress-IM, vlan-id 230
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300320             *[EVPN/7] 02:27:06, routing-instance VS-1, route-type
Egress-MAC, ESI 00:11:11:11:11:11:11:11:11:11
                      to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                      to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2

                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
300336             *[EVPN/7] 02:27:06, routing-instance VS-1, route-type
Egress-MAC, ESI 00:33:33:33:33:33:33:33:33:33
                      to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
```

```
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300368             *[EVPN/7] 02:27:07, routing-instance VS-2, route-type
Egress-MAC, ESI 00:33:33:33:33:33:33:33:33:33
                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300384             *[EVPN/7] 02:27:07, routing-instance VS-2, route-type
Egress-MAC, ESI 00:11:11:11:11:11:11:11:11:11
                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2

                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
300416             *[EVPN/7] 02:27:06, routing-instance mhevpn, route-type
Egress-MAC, ESI 00:33:33:33:33:33:33:33:33:33
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300432             *[EVPN/7] 02:27:06, routing-instance mhevpn, route-type
Egress-MAC, ESI 00:11:11:11:11:11:11:11:11:11
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2

                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
300480             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-1, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300496             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-2, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300560             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-1, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300592             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 VS-2, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
300608             *[EVPN/7] 02:29:23
                    > via ge-0/0/1.1001, Pop
300624             *[EVPN/7] 02:29:23
                    > via ge-0/0/1.2001, Pop
301232             *[EVPN/7] 02:29:17
                    > via ge-0/0/1.1002, Pop
301296             *[EVPN/7] 02:29:10
                    > via ge-0/0/1.1003, Pop
301312             *[EVPN/7] 02:27:06
                    > via ae10.2003, Pop
                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301360             *[EVPN/7] 02:29:01
                    > via ge-0/0/1.1004, Pop
301408             *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 vpws1004, route-type Egress, vlan-id 2004
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
301456             *[EVPN/7] 02:27:06
                    > via ae10.1010, Pop
                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301552             *[EVPN/7] 02:27:07, routing-instance VS-1, route-type
Egress-MAC, ESI 00:22:22:22:22:22:22:22:22:22
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301568             *[EVPN/7] 02:27:07, routing-instance VS-2, route-type
```

```
Egress-MAC, ESI 00:22:22:22:22:22:22:22:22:22
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301648              *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 vpws1010, route-type Egress, vlan-id 2010
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
301664              *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 mhevpn, route-type Egress-MAC
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
301680              *[EVPN/7] 02:27:07, remote-pe 100.100.100.2, routing-instance
 mhevpn, route-type Egress-MAC
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
301696              *[EVPN/7] 02:27:07, routing-instance mhevpn, route-type
Egress-MAC, ESI 00:22:22:22:22:22:22:22:22:22
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301712              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-2, route-type Egress-MAC
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301728              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-1, route-type Egress-MAC
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301744              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-2, route-type Egress-IM, vlan-id 230
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301760              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 vpws1010, route-type Egress, vlan-id 2010
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301776              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 mhevpn, route-type Egress-MAC
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301792              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-1, route-type Egress-IM, vlan-id 130
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301808              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 vpws1004, route-type Egress, vlan-id 2004
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301824              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 mhevpn, route-type Egress-IM, vlan-id 10
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301840              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 vpws1002, route-type Egress, vlan-id 2002
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301856              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 vpws1003, route-type Egress, vlan-id 2003
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301872              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 vpws1003, route-type Egress Protection, vlan-id 2003
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301888              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 vpws1010, route-type Egress Protection, vlan-id 1010
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301904              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-2, route-type Egress-IM, vlan-id 220
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301920              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-2, route-type Egress-IM, vlan-id 210
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
301936              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-IM, vlan-id 230
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301952              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-SH, vlan-id 230
```

```
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301968              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-IM, vlan-id 220
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
301984              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-SH, vlan-id 220
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302000              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-IM, vlan-id 210
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302016              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-SH, vlan-id 210
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302032              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-2, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302048              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-2, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302064              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302080              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-2, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302096              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-1, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302112              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-1, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302128              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302144              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-MAC
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302160              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-1, route-type Egress-IM, vlan-id 120
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302176              *[EVPN/7] 02:27:07, remote-pe 100.100.100.1, routing-instance
 VS-1, route-type Egress-IM, vlan-id 110
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302192              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-IM, vlan-id 130
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302208              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-SH, vlan-id 130
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302224              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-IM, vlan-id 120
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302240              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-SH, vlan-id 120
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302256              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-IM, vlan-id 110
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302272              *[EVPN/7] 02:27:07, remote-pe 100.100.100.3, routing-instance
 VS-1, route-type Egress-SH, vlan-id 110
                    > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
```

```
302288             *[EVPN/7] 02:27:06, remote-pe 100.100.100.1, routing-instance
 mhevpn, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302304             *[EVPN/7] 02:27:06, remote-pe 100.100.100.1, routing-instance
 mhevpn, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302320             *[EVPN/7] 02:27:06, remote-pe 100.100.100.3, routing-instance
 mhevpn, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302336             *[EVPN/7] 02:27:06, remote-pe 100.100.100.3, routing-instance
 mhevpn, route-type Egress-MAC
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302352             *[EVPN/7] 02:27:06, remote-pe 100.100.100.3, routing-instance
 vpws1004, route-type Egress, vlan-id 2004
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302368             *[EVPN/7] 02:27:06, remote-pe 100.100.100.3, routing-instance
 mhevpn, route-type Egress-IM, vlan-id 10
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302384             *[EVPN/7] 02:27:06, remote-pe 100.100.100.3, routing-instance
 mhevpn, route-type Egress-SH, vlan-id 10
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302400             *[EVPN/7] 02:26:21
                    > via ge-0/0/1.3001, Pop
302432             *[EVPN/7] 02:26:21, remote-pe 100.100.100.3, routing-instance
 vpws3001, route-type Egress, vlan-id 40000
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302448             *[EVPN/7] 02:26:21, remote-pe 100.100.100.1, routing-instance
 vpws3001, route-type Egress, vlan-id 40000
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302464             *[EVPN/7] 02:26:20, remote-pe 100.100.100.2, routing-instance
 vpws3001, route-type Egress, vlan-id 40000
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
302480             *[EVPN/7] 02:26:14
                    > via ge-0/0/1.3016, Pop
302512             *[EVPN/7] 02:26:14, remote-pe 100.100.100.1, routing-instance
 vpws3016, route-type Egress, vlan-id 40016
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302528             *[EVPN/7] 02:26:14, remote-pe 100.100.100.2, routing-instance
 vpws3016, route-type Egress, vlan-id 40016
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
302560             *[EVPN/7] 02:26:06
                    > via ae10.3011, Pop
                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302592             *[EVPN/7] 02:26:07, remote-pe 100.100.100.1, routing-instance
 vpws3011, route-type Egress, vlan-id 401100
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
302608             *[EVPN/7] 02:26:07, remote-pe 100.100.100.2, routing-instance
 vpws3011, route-type Egress, vlan-id 401100
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
302624             *[EVPN/7] 02:26:07, remote-pe 100.100.100.3, routing-instance
 vpws3011, route-type Egress Protection, vlan-id 301100
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302656             *[EVPN/7] 02:25:59
                    > via ae10.3006, Pop
                    to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302688             *[EVPN/7] 02:26:00, remote-pe 100.100.100.2, routing-instance
 vpws3006, route-type Egress, vlan-id 400600
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2
302704             *[EVPN/7] 02:26:00, remote-pe 100.100.100.1, routing-instance
 vpws3006, route-type Egress, vlan-id 400600
                  > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
```

```
302720             *[EVPN/7] 02:25:59, remote-pe 100.100.100.3, routing-instance
 vpws3006, route-type Egress, vlan-id 400600
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
302736             *[EVPN/7] 02:25:59, remote-pe 100.100.100.3, routing-instance
 vpws3006, route-type Egress Protection, vlan-id 300600
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
ge-0/0/1.1001      *[EVPN/7] 02:29:23
                    > via ge-0/0/1.2001
ge-0/0/1.2001      *[EVPN/7] 02:29:23
                    > via ge-0/0/1.1001
ge-0/0/1.1002      *[EVPN/7] 02:27:06
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
ae10.2003          *[EVPN/7] 02:29:10
                    > via ge-0/0/1.1003
ge-0/0/1.1003      *[EVPN/7] 02:27:06
                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3

                     > via ae10.2003
ge-0/0/1.1004      *[EVPN/7] 02:27:06
                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2

                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
ae10.1010          *[EVPN/7] 02:27:06
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
ge-0/0/1.3001      *[EVPN/7] 02:26:20
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2

                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
ge-0/0/1.3016      *[EVPN/7] 02:26:13
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
ae10.3011          *[EVPN/7] 02:26:06
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1
ae10.3006          *[EVPN/7] 02:25:59
                   > to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe1

                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe2

                     to 100.46.1.2 via ge-0/0/5.0, label-switched-path pe4_to_pe3
```

show route table mpls.0 protocol ospf

```
user@host> show route table mpls.0 protocol ospf
mpls.0: 29 destinations, 29 routes (29 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

299952             *[L-OSPF/10] 23:59:42, metric 0
                    > to 10.0.10.70 via lt-1/2/0.14, Pop
                      to 10.0.6.60 via lt-1/2/0.12, Swap 800070, Push 800030(top)
299952(S=0)        *[L-OSPF/10] 23:59:42, metric 0
                    > to 10.0.10.70 via lt-1/2/0.14, Pop
                      to 10.0.6.60 via lt-1/2/0.12, Swap 800070, Push 800030(top)
299968             *[L-OSPF/10] 23:59:48, metric 0
                    > to 10.0.6.60 via lt-1/2/0.12, Pop
```

show route table mpls.0 extensive (PTX Series)

```
user@host> show route table mpls.0 extensive
ge-0/0/2.600 (1 entry, 1 announced)
TSI:
KRT in-kernel ge-0/0/2.600.0      /32 -> {composite(570)}
        *L2VPN  Preference: 7
                Next hop type: Indirect
                Address: 0x9438f34
                Next-hop reference count: 2
                Next hop type: Router, Next hop index: 567
                Next hop: 10.0.0.1 via ge-0/0/1.0, selected
                Label operation: Push 299808
                Label TTL action: prop-ttl
                Load balance label: Label 299808:None;
                Session Id: 0x1
                Protocol next hop: 10.255.255.1
                Label operation: Push 299872 Offset: 252
                Label TTL action: no-prop-ttl
                Load balance label: Label 299872:Flow label PUSH;
                Composite next hop: 0x9438ed8 570 INH Session ID: 0x2
                Indirect next hop: 0x9448208 262142 INH Session ID: 0x2
                State: <Active Int>
                Age: 47         Metric2: 1
                Validation State: unverified
                Task: Common L2 VC
                Announcement bits (2): 0-KRT 2-Common L2 VC
                AS path: I
                Composite next hops: 1
                        Protocol next hop: 10.255.255.1 Metric: 1
                        Label operation: Push 299872 Offset: 252
                        Label TTL action: no-prop-ttl
                        Load balance label: Label 299872:Flow label PUSH;
                        Composite next hop: 0x9438ed8 570 INH Session ID: 0x2
                        Indirect next hop: 0x9448208 262142 INH Session ID: 0x2
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.0.0.1 via ge-0/0/1.0
                                Session Id: 0x1
                        10.255.255.1/32 Originating RIB: inet.3
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 1
                                Nexthop: 10.0.0.1 via ge-0/0/1.0
```

show route table mpls.0 (RSVP Route—Transit LSP)

```
user@host> show route table mpls.0

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:37:31, metric 1
                      Receive
1                  *[MPLS/0] 00:37:31, metric 1
                      Receive
2                  *[MPLS/0] 00:37:31, metric 1
                      Receive
13                 *[MPLS/0] 00:37:31, metric 1
                      Receive
300352             *[RSVP/7/1] 00:08:00, metric 1
```

```
                    > to 10.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300352(S=0)         *[RSVP/7/1] 00:08:00, metric 1
                    > to 10.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300384              *[RSVP/7/2] 00:05:20, metric 1
                     > to 10.64.1.106 via ge-1/0/0.0, Pop
300384(S=0)         *[RSVP/7/2] 00:05:20, metric 1
                     > to 10.64.1.106 via ge-1/0/0.0, Pop
```

show route table vpls_1 detail

```
user@host> show route table vpls_1 detail
vpls_1.l2vpn.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
Restart Complete

172.16.1.11:1000:1:1/96 (1 entry, 1 announced)
*L2VPN Preference: 170/-1
Receive table: vpls_1.l2vpn.0
Next-hop reference count: 2
State: <Active Int Ext>
Age: 4:29:47 Metric2: 1
Task: vpls_1-l2vpn
Announcement bits (1): 1-BGP.0.0.0.0+179
AS path: I
Communities: Layer2-info: encaps:VPLS, control flags:Site-Down
Label-base: 800000, range: 8, status-vector: 0xFF
```

show route table vpn-a

```
user@host> show route table vpn-a
vpn-a.l2vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both
192.168.16.1:1:1:1/96
                    *[VPN/7] 05:48:27
                      Discard
192.168.24.1:1:2:1/96
                    *[BGP/170] 00:02:53, localpref 100, from 192.168.24.1
                       AS path: I
                     > to 10.0.16.2 via fe-0/0/1.0, label-switched-path am
192.168.24.1:1:3:1/96
                    *[BGP/170] 00:02:53, localpref 100, from 192.168.24.1
                       AS path: I
                     > to 10.0.16.2 via fe-0/0/1.0, label-switched-path am
```

show route table vpn-a.mdt.0

```
user@host> show route table vpn-a.mdt.0
vpn-a.mdt.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:1:0:10.255.14.216:232.1.1.1/144
                    *[MVPN/70] 01:23:05, metric2 1
                       Indirect
1:1:1:10.255.14.218:232.1.1.1/144
                    *[BGP/170] 00:57:49, localpref 100, from 10.255.14.218
                       AS path: I
                     > via so-0/0/0.0, label-switched-path r0e-to-r1
1:1:2:10.255.14.217:232.1.1.1/144
                    *[BGP/170] 00:57:49, localpref 100, from 10.255.14.217
```

```
                        AS path: I
> via so-0/0/1.0, label-switched-path r0-to-r2
```

show route table VPN-A detail

```
user@host> show route table VPN-A detail
VPN-AB.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
10.255.179.9/32 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 10.255.179.13:200
                Next hop type: Indirect
                Next-hop reference count: 5
                Source: 10.255.179.13
                Next hop type: Router, Next hop index: 732
                Next hop: 10.39.1.14 via fe-0/3/0.0, selected
                Label operation: Push 299824, Push 299824(top)
                Protocol next hop: 10.255.179.13
                Push 299824
                Indirect next hop: 8f275a0 1048574
                State: (Secondary Active Int Ext)
                Local AS: 1 Peer AS: 1
                Age: 3:41:06 Metric: 1 Metric2: 1
                Task: BGP_1.10.255.179.13+64309
                Announcement bits (2): 0-KRT 1-BGP RT Background
                AS path: I
                Communities: target:1:200 rte-type:0.0.0.0:1:0
                Import Accepted
                VPN Label: 299824 TTL Action: vrf-ttl-propagate
                Localpref: 100
                Router ID: 10.255.179.13
                Primary Routing Table bgp.l3vpn.0
```

show route table VPN-AB.inet.0

```
user@host> show route table VPN-AB.inet.0
      VPN-AB.inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
      + = Active Route, - = Last Active, * = Both

      10.39.1.0/30       *[OSPF/10] 00:07:24, metric 1
                          > via so-7/3/1.0
      10.39.1.4/30       *[Direct/0] 00:08:42
                          > via so-5/1/0.0
      10.39.1.6/32       *[Local/0] 00:08:46
                            Local
      10.255.71.16/32    *[Static/5] 00:07:24
                          > via so-2/0/0.0
      10.255.71.17/32    *[BGP/170] 00:07:24, MED 1, localpref 100, from
      10.255.71.15
                             AS path: I
                          > via so-2/1/0.0, Push 100020, Push 100011(top)
      10.255.71.18/32    *[BGP/170] 00:07:24, MED 1, localpref 100, from
      10.255.71.15
                             AS path: I
                          > via so-2/1/0.0, Push 100021, Push 100011(top)
      10.255.245.245/32  *[BGP/170] 00:08:35, localpref 100
                             AS path: 2 I
                          > to 10.39.1.5 via so-5/1/0.0
      10.255.245.246/32  *[OSPF/10] 00:07:24, metric 1
                          > via so-7/3/1.0
```

show route table VPN_blue.mvpn-inet6.0

```
user@host> show route table VPN_blue.mvpn-inet6.0
vpn_blue.mvpn-inet6.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:10.255.2.202:65536:10.255.2.202/432
                   *[BGP/170] 00:02:37, localpref 100, from 10.255.2.202
                      AS path: I
                    > via so-0/1/3.0
1:10.255.2.203:65536:10.255.2.203/432
                   *[BGP/170] 00:02:37, localpref 100, from 10.255.2.203
                      AS path: I
                    > via so-0/1/0.0
1:10.255.2.204:65536:10.255.2.204/432
                   *[MVPN/70] 00:57:23, metric2 1
                      Indirect
5:10.255.2.202:65536:128:::192.168.90.2:128:ffff::1/432
                   *[BGP/170] 00:02:37, localpref 100, from 10.255.2.202
                      AS path: I
                    > via so-0/1/3.0
6:10.255.2.203:65536:64500:128:::10.12.53.12:128:ffff::1/432
                   *[PIM/105] 00:02:37
                      Multicast (IPv6)
7:10.255.2.202:65536:64500:128:::192.168.90.2:128:ffff::1/432
                   *[MVPN/70] 00:02:37, metric2 1
                      Indirect
```

show route table vrf1.mvpn.0 extensive

```
user@host> show route table vrf1.mvpn.0 extensive
1:10.255.50.77:1:10.255.50.77/240 (1 entry, 1 announced)
        *MVPN   Preference: 70
                PMSI: Flags 0x0: Label 0: RSVP-TE:
Session_13[10.255.50.77:0:25624:10.255.50.77]
                Next hop type: Indirect
                Address: 0xbb2c944
                Next-hop reference count: 360
                Protocol next hop: 10.255.50.77
                Indirect next hop: 0x0 - INH Session ID: 0x0
                State: <Active Int Ext>
                Age: 53:03      Metric2: 1
                Validation State: unverified
                Task: mvpn global task
                Announcement bits (3): 0-PIM.vrf1 1-mvpn global task 2-rt-export

                AS path: I
```

show route table inetflow detail

```
user@host> show route table inetflow detail
inetflow.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
10.12.44.1,*/48 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next-hop reference count: 2
                State: <Active Ext>
                Local AS: 64502 Peer AS: 64500
                Age: 4
                Task: BGP_64500.10.12.99.5+3792
                Announcement bits (1): 0-Flow
```

```
                AS path: 64500 I
                Communities: traffic-rate:0:0
                Validation state: Accept, Originator: 10.12.99.5
                Via: 10.12.44.0/24, Active
                Localpref: 100
                Router ID: 10.255.71.161

10.12.56.1,*/48 (1 entry, 1 announced)
        *Flow   Preference: 5
                Next-hop reference count: 2
                State: <Active>
                Local AS: 64502
                Age: 6:30
                Task: RT Flow
                Announcement bits (2): 0-Flow 1-BGP.0.0.0.0+179
                AS path: I
                Communities: 1:1


user@host> show route table green.l2vpn.0 (VPLS Multihoming with FEC 129)
green.l2vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.1.1.2:100:10.1.1.2/96 AD
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
10.1.1.4:100:10.1.1.4/96 AD
                   *[BGP/170] 1d 03:11:02, localpref 100, from 10.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
10.1.1.2:100:1:0/96 MH
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
10.1.1.4:100:1:0/96 MH
                   *[BGP/170] 1d 03:11:02, localpref 100, from 10.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
10.1.1.4:NoCtrlWord:5:100:100:10.1.1.2:10.1.1.4/176
                   *[VPLS/7] 1d 03:11:02, metric2 1
                    >    via ge-1/2/1.5
10.1.1.4:NoCtrlWord:5:100:100:10.1.1.4:10.1.1.2/176
                   *[LDP/9] 1d 03:11:02
                         Discard


user@host> show route table red extensive
red.inet.0: 364481 destinations, 714087 routes (364480 active, 48448 holddown, 1
 hidden)
10.0.0.0/32 (3 entries, 1 announced)
        State: <OnList CalcForwarding>
TSI:
KRT in-kernel 10.0.0.0/32 -> {composite(1048575)} Page 0 idx 1 Type 1 val 0x934342c

    Nexthop: Self
    AS path: [2] I
    Communities: target:2:1
Path 10.0.0.0 from 10.3.0.0 Vector len 4.  Val: 1
        @BGP    Preference: 170/-1
                Route Distinguisher: 2:1
                Next hop type: Indirect
                Address: 0x258059e4
                Next-hop reference count: 2
```

```
                    Source: 2.2.0.0
                    Next hop type: Router
                    Next hop: 10.1.1.1 via ge-1/1/9.0, selected
                    Label operation: Push 707633
                    Label TTL action: prop-ttl
                    Session Id: 0x17d8
                    Protocol next hop: 10.2.0.0
                    Push 16
                    Composite next hop: 0x25805988 - INH Session ID: 0x193c
                    Indirect next hop: 0x23eea900 - INH Session ID: 0x193c
                    State: <Secondary Active Int Ext ProtectionPath ProtectionCand>
                    Local AS:     2 Peer AS:     2
                    Age: 23         Metric2: 35
                    Validation State: unverified
                    Task: BGP_172.16.2.0.0+34549
                    AS path: I
                    Communities: target:2:1
                    Import Accepted
                    VPN Label: 16
                    Localpref: 0
                    Router ID: 10.2.0.0
                    Primary Routing Table bgp.l3vpn.0
                    Composite next hops: 1
                            Protocol next hop: 10.2.0.0 Metric: 35
                            Push 16
                            Composite next hop: 0x25805988 - INH Session ID: 0x193c
                            Indirect next hop: 0x23eea900 - INH Session ID: 0x193c
                            Indirect path forwarding next hops: 1
                                    Next hop type: Router
                                    Next hop: 10.1.1.1 via ge-1/1/9.0
                                    Session Id: 0x17d8
                            2.2.0.0/32 Originating RIB: inet.3
                              Metric: 35                      Node path count: 1
                              Forwarding nexthops: 1
                                    Nexthop: 10.1.1.1 via ge-1/1/9.0
            BGP     Preference: 170/-1
                    Route Distinguisher: 2:1
                    Next hop type: Indirect
                    Address: 0x9347028
                    Next-hop reference count: 3
                    Source: 10.3.0.0
                    Next hop type: Router, Next hop index: 702
                    Next hop: 10.1.4.2 via ge-1/0/0.0, selected
                    Label operation: Push 634278
                    Label TTL action: prop-ttl
                    Session Id: 0x17d9
                    Protocol next hop: 10.3.0.0
                    Push 16
                    Composite next hop: 0x93463a0 1048575 INH Session ID: 0x17da
                    Indirect next hop: 0x91e8800 1048574 INH Session ID: 0x17da
                    State: <Secondary NotBest Int Ext ProtectionPath ProtectionCand>

                    Inactive reason: Not Best in its group - IGP metric
                    Local AS:     2 Peer AS:     2
                    Age: 3:34       Metric2: 70
                    Validation State: unverified
                    Task: BGP_172.16.3.0.0+32805
                    Announcement bits (2): 0-KRT 1-BGP_RT_Background
                    AS path: I
                    Communities: target:2:1
                    Import Accepted
```

```
                    VPN Label: 16
                    Localpref: 0
                    Router ID: 10.3.0.0
                    Primary Routing Table bgp.l3vpn.0
                    Composite next hops: 1
                            Protocol next hop: 10.3.0.0 Metric: 70
                            Push 16
                            Composite next hop: 0x93463a0 1048575 INH Session ID:
0x17da
                            Indirect next hop: 0x91e8800 1048574 INH Session ID:
0x17da
                            Indirect path forwarding next hops: 1
                                    Next hop type: Router
                                    Next hop: 10.1.4.2 via ge-1/0/0.0
                                    Session Id: 0x17d9
                            10.3.0.0/32 Originating RIB: inet.3
                              Metric: 70                    Node path count: 1
                              Forwarding nexthops: 1
                                    Nexthop: 10.1.4.2 via ge-1/0/0.0
            #Multipath Preference: 255
                    Next hop type: Indirect
                    Address: 0x24afca30
                    Next-hop reference count: 1
                    Next hop type: Router
                    Next hop: 10.1.1.1 via ge-1/1/9.0, selected
                    Label operation: Push 707633
                    Label TTL action: prop-ttl
                    Session Id: 0x17d8
                    Next hop type: Router, Next hop index: 702
                    Next hop: 10.1.4.2 via ge-1/0/0.0
                    Label operation: Push 634278
                    Label TTL action: prop-ttl
                    Session Id: 0x17d9
                    Protocol next hop: 10.2.0.0
                    Push 16
                    Composite next hop: 0x25805988 - INH Session ID: 0x193c
                   Indirect next hop: 0x23eea900 - INH Session ID: 0x193c Weight 0x1

                    Protocol next hop: 10.3.0.0
                    Push 16
                    Composite next hop: 0x93463a0 1048575 INH Session ID: 0x17da
                  Indirect next hop: 0x91e8800 1048574 INH Session ID: 0x17da Weight
 0x4000
                    State: <ForwardingOnly Int Ext>
                    Inactive reason: Forwarding use only
                    Age: 23         Metric2: 35
                    Validation State: unverified
                    Task: RT
                    AS path: I
                    Communities: target:2:1
```

show route table bgp.evpn.0 extensive |no-more (EVPN)

```
show route table bgp.evpn.0 extensive | no-more
bgp.evpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
2:1000:10::100::00:aa:aa:aa:aa:aa/304 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 1000:10
                Next hop type: Indirect
                Address: 0x9420fd0
                Next-hop reference count: 12
```

```
                Source: 10.2.3.4
                Protocol next hop: 10.2.3.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
         State: Local AS: 17  Peer AS:17  Age:21:12  Metric2:1  Validation State:
 unverified
                Task: BGP_17.1.2.3.4+50756
                AS path: I
                Communities: target:1111:8388708 encapsulation0:0:0:0:3
                Import Accepted
                Route Label: 100
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 10.2.3.4
                Secondary Tables: default-switch.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.2.3.4 Metric: 1
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.1 via xe-0/0/1.0
                                Session Id: 0x2
                        1.2.3.4/32 Originating RIB: inet.0
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 10.92.78.102 via em0.0

2:1000:10::200::00:bb:bb:bb:bb:bb/304 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 1000:10
                Next hop type: Indirect
                Address: 0x9420fd0
                Next-hop reference count: 12
                Source: 10.2.3.4
                Protocol next hop: 10.2.3.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
          State: Local AS:17  Peer AS:17  Age:19:43  Metric2:1  Validation
State:unverified
                Task: BGP_17.1.2.3.4+50756
                AS path: I
                Communities: target:2222:22 encapsulation0:0:0:0:3
                Import Accepted
                Route Label: 200
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 10.2.3.4
                Secondary Tables: default-switch.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.2.3.4 Metric: 1
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.1 via xe-0/0/1.0
                                Session Id: 0x2
                        10.2.3.4/32 Originating RIB: inet.0
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 10.92.78.102 via em0.0

2:1000:10::300::00:cc:cc:cc:cc:cc/304 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 1000:10
```

```
                Next hop type: Indirect
                Address: 0x9420fd0
                Next-hop reference count: 12
                Source: 10.2.3.4
                Protocol next hop: 10.2.3.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
          State: Local AS:17  Peer AS:17  Age:17:21  Metric2:1  Validation State:
 unverified  Task: BGP 17,1,2,3,4+50756
           AS path: I
                Communities: target:3333:33 encapsulation0:0:0:0:3
                Import Accepted
                Route Label: 300
                ESI: 00:00:00:00:00:00:00:00:00:00
                Localpref: 100
                Router ID: 10.2.3.4
                Secondary Tables: default-switch.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.2.3.4 Metric: 1
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.1 via xe-0/0/1.0
                                Session Id: 0x2
                        10.2.3.4/32 Originating RIB: inet.0
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 10.92.78.102 via em0.0

3:1000:10::100::1.2.3.4/304 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 1000:10
                PMSI: Flags 0x0: Label 100: Type INGRESS-REPLICATION 1.2.3.4
                Next hop type: Indirect
                Address: 0x9420fd0
                Next-hop reference count: 12
                Source: 10.2.3.4
                Protocol next hop: 10.2.3.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
          State: Local AS:17  Peer AS:17 Age:37:01  Metric2:1  Validation State:
 unverified  Task: BGP 17.1.2.3.4+50756
                AS path: I
                Communities: target:1111:8388708 encapsulation0:0:0:0:3
                Import Accepted
                Localpref: 100
                Router ID: 10.2.3.4
                Secondary Tables: default-switch.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.2.3.4 Metric: 1
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.1 via xe-0/0/1.0
                                Session Id: 0x2
                        10.2.3.4/32 Originating RIB: inet.0
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 10.92.78.102 via em0.0

3:1000:10::200::1.2.3.4/304 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 1000:10
```

```
                PMSI: Flags 0x0: Label 200: Type INGRESS-REPLICATION 1.2.3.4
                Next hop type: Indirect
                Address: 0x9420fd0
                Next-hop reference count: 12
                Source: 10.2.3.4
                Protocol next hop: 10.2.3.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
          State: Local AS: 17 Peer AS: 17   Age:35:22  Metric2:1  Validation
State:unverified  Task: BGP 17.1.2.3.4+50756
           AS path:I  Communities: target:2222:22 encapsulation):0:0:0:0:3

Import Accepted
                Localpref: 100
                Router ID: 10.2.3.4
                Secondary Tables: default-switch.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.2.3.4 Metric: 1
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.1 via xe-0/0/1.0
                                Session Id: 0x2
                        10.2.3.4/32 Originating RIB: inet.0
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 10.92.78.102 via em0.0

3:1000:10::300::1.2.3.4/304 (1 entry, 0 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 1000:10
                PMSI: Flags 0x0: Label 300: Type INGRESS-REPLICATION 1.2.3.4
                Next hop type: Indirect
                Address: 0x9420fd0
                Next-hop reference count: 12
                Source: 10.2.3.4
                Protocol next hop: 10.2.3.4
                Indirect next hop: 0x2 no-forward INH Session ID: 0x0
         State: Local AS: 17 Peer AS:  17  Age 35:22 Metric2:1  Validation State:
 unverified   Task: BGP 17.1.2.3.4+5075
          6  AS path: I   Communities: target:3333:33  encapsulation0:0:0:0:3
Import Accepted   Localpref:100
          Router ID: 10.2.3.4
                Secondary Tables: default-switch.evpn.0
                Indirect next hops: 1
                        Protocol next hop: 10.2.3.4 Metric: 1
                        Indirect next hop: 0x2 no-forward INH Session ID: 0x0
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 10.10.10.1 via xe-0/0/1.0
                                Session Id: 0x2
                        10.2.3.4/32 Originating RIB: inet.0
                          Metric: 1                     Node path count: 1
                          Forwarding nexthops: 2
                                Nexthop: 10.92.78.102 via em0.0
```

show sap listen

Syntax
show sap listen
<brief | detail>
<logical-system (all | *logical-system-name*)>

Release Information
Command introduced before Junos OS Release 7.4.

Description
Display the addresses that the router is listening to in order to receive multicast Session Announcement Protocol (SAP) session announcements.

Options
none—Display standard information about the addresses that the router is listening to in order to receive multicast SAP session announcements.

brief | detail—(Optional) Display the specified level of output.

logical-system (all | *logical-system-name*)—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level
view

List of Sample Output
show sap listen on page 1775
show sap listen brief on page 1776
show sap listen detail on page 1776

Output Fields
Table 109 on page 1775 describes the output fields for the **show sap listen** command. Output fields are listed in the approximate order in which they appear.

Table 109: show sap listen Output Fields

Field Name	Field Description
Group address	Address of the group that the local router is listening to for SAP messages.
Port	UDP port number used for SAP.

Sample Output

show sap listen

```
user@host> show sap listen
Group address   Port
224.2.127.254   9875
239.255.255.255 9875
```

show sap listen brief

The output for the **show sap listen brief** command is identical to that for the **show sap listen** command. For sample output, see show sap listen on page 1775.

show sap listen detail

The output for the **show sap listen detail** command is identical to that for the **show sap listen** command. For sample output, see show sap listen on page 1775.

test msdp

Syntax test msdp (dependent-peers *prefix* | rpf-peer *originator*)
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

Release Information Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 12.1 for the QFX Series.
Command introduced in Junos OS Release 14.1X53-D20 for the OCX Series.

Description Find Multicast Source Discovery Protocol (MSDP) peers.

Options

dependent-peers *prefix*—Find downstream dependent MSDP peers.

rpf-peer *originator*—Find the MSDP reverse-path-forwarding (RPF) peer for the originator.

instance *instance-name*—(Optional) Find MDSP peers for the specified routing instance.

logical-system (all | *logical-system-name)*—(Optional) Perform this operation on all logical systems or on a particular logical system.

Required Privilege Level view

List of Sample Output test msdp dependent-peers on page 1777

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

test msdp dependent-peers

```
user@host> test msdp dependent-peers 10.0.0.1/24
```