# Junos® OS

## Multicast Protocols Configuration Guide

Release
13.1

Published: 2013-02-12

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net



*Junos® OS Multicast Protocols Configuration Guide*
13.1


The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

## END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# 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 http://www.juniper.net/techpubs/.

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 http://www.juniper.net/books.

## Supported Platforms

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

- T Series
- MX Series
- M Series
- J Series
- SRX Series
- PTX 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 the CLI User Guide.

## Documentation Conventions

Table 1 on page xxxiii 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. |

Table 2 on page xxxiv 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:<br><br>user@host> **configure** |
| `Fixed-width text like this` | Represents output that appears on the terminal screen. | `user@host>` **show chassis alarms**<br><br>`No alarms currently active` |
| *Italic text like this* | • Introduces or emphasizes important new terms.<br>• Identifies book names.<br>• Identifies RFC and Internet draft titles. | • A policy *term* is a named structure that defines match conditions and actions.<br>• *Junos OS System Basics Configuration Guide*<br>• 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:<br><br>[edit]<br>root@# **set system domain-name** *domain-name* |
| **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.<br>• The console port is labeled **CONSOLE**. |
| < > (angle brackets) | Enclose 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**<br><br>**(*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) | Enclose a variable for which you can substitute one or more values. | **community name members [** ***community-ids*** **]** |
| Indention and braces ( { } ) | Identify a level in the configuration hierarchy. | [edit]<br>routing-options {<br>  static {<br>  route default {<br>  nexthop *address*;<br>  retain;<br>  }<br>  }<br>} |
| ; (semicolon) | Identifies a leaf statement at a configuration hierarchy level. | |

### J-Web GUI Conventions

Table 2: Text and Syntax Conventions *(continued)*

| Convention | Description | Examples |
|---|---|---|
| **Bold text like this** | Represents J-Web graphical user interface (GUI) items you click or select. | • In the Logical Interfaces box, select **All Interfaces**.<br>• To cancel the configuration, click **Cancel**. |
| > (bold right angle bracket) | Separates levels in a hierarchy of J-Web 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 send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/ . If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release 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 JNASC 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 http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://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: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/

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

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

## 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 http://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 http://www.juniper.net/support/requesting-support.html.

PART 1

# Overview

CHAPTER 1

# Multicast Overview



## 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.

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 routers 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 routers 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 routers 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 routers. Multicast routers distribute the multicast traffic across the network from source to destinations. The multicast router 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 routers 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.

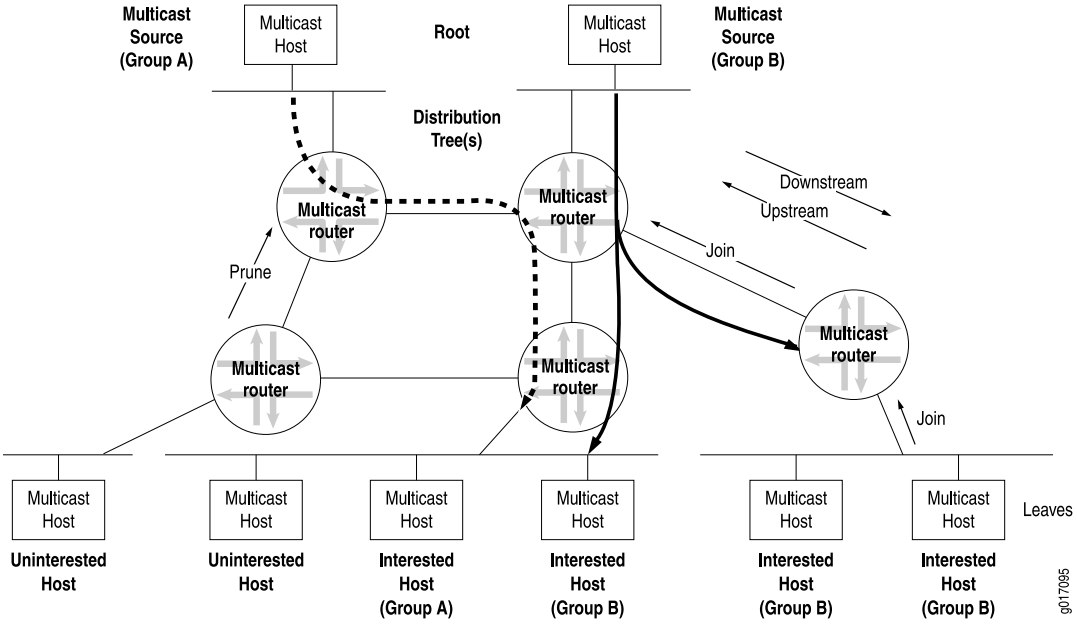
For radio station or news ticker traffic, multicast provides the most efficient and effective outcome, with none of the drawbacks and all of the advantages of the other methods. A single source of multicast packets finds its way to every *interested* receiver. As with broadcast, the transmitting host generates only a single stream of IP packets, so the load remains constant whether there is one receiver or one million. The network routers replicate the packets and deliver the packets to the proper receivers, but only the replication role is a new one for routers. 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 routers 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 *router*, which is able to replicate packets and is therefore multicast-capable. The routers 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 router 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 router to receive multicast packets. The interface on the router 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 router, where *N* is the number of logical interfaces on the router. 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. Three multicast strategies—reverse-path forwarding (RPF), shortest-path tree (SPT), and administrative scoping—help prevent routing loops by defining routing paths in different ways.

## Reverse-Path Forwarding for Loop Prevention

The router'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 router 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. Routers 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 router operating as a rendezvous point (RP), a feature of sparse mode multicast protocols.

### Administrative Scoping for Loop Prevention

Scoping limits the routers 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*. Routers 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 router that has at least one interested receiver is a *leaf* on the distribution tree. Routers 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 router. 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 router 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 router, 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 routers, 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.

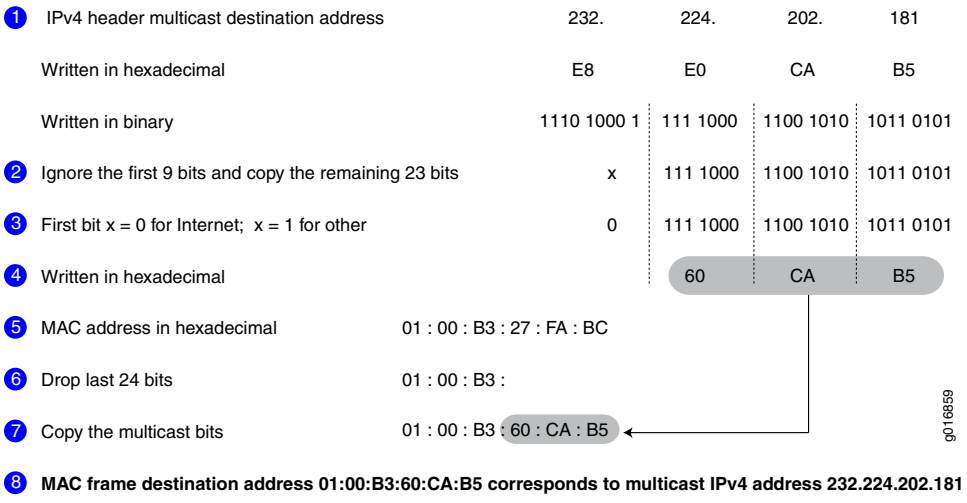
However, there is an easy way to effectively use Layer 2 frames for multicast purposes. The MAC address has a bit that is set to 0 for unicast (the LAN term is *individual address*) and set to 1 to indicate that this is a multicast address. Some of these addresses are reserved for multicast groups of 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 10.

### 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 router 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.

Routers closely monitor the status of the incoming and outgoing interfaces, a process that determines the *multicast forwarding state*. A router with a multicast forwarding state for a particular multicast group is essentially “turned on” for that group's content. Interfaces on the router'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 router 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 router could use (*,224.1.1.2) to represent the state of a router forwarding traffic from both sources to the group.

## Multicast Routing Protocols

Multicast routing protocols enable a collection of multicast routers 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 routers 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 routers 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 router to use any unicast routing protocol and performs RPF checks using the unicast routing table. PIM dense mode has an implicit join message, so routers 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 router 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 routers 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 router 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 router, 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 routers.
- 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 router 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 |
| 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 router 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.

## 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.

PIM is one of the most rapidly evolving specifications on the Internet today. Since its introduction in 1995, PIM has already seen two major revisions to its packet structure (PIM version 1 [PIMv1] and PIM version 2 [PIMv2]), two major RFCs (RFC 2362 obsoleted RFC 2117), and numerous drafts describing major components of PIM, such as many-to-many trees and source-specific multicast (SSM). Long-lasting RFCs are not a feature of PIM, and virtually all of PIM must be researched, understood, and implemented directly from Internet drafts. In fact, no current RFC describes PIMv1 at all. The drafts have all expired, and PIMv1 was never issued as an official RFC.

PIM itself is not nonstandard or unstable, however. PIM has been a promising multicast routing protocol since its inception, especially PIM sparse mode, the first real sparse-mode multicast routing protocol. Work continues on PIM in a number of areas, from bidirectional trees to network management, and the rapid pace of development makes drafts essential for PIM.

PIMv1 and PIMv2 can coexist on the same router 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 routers 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 router interface to PIMv1. Because the difference between PIMv1 and PIMv2 involves the message format, but not the meaning of the message or how the router processes the PIM message, a router 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.

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, routers must join and leave multicast groups explicitly. Upstream routers do not forward multicast traffic to a downstream router unless the downstream router has sent an explicit request (by means of a join message) to the rendezvous point (RP) router 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.

- 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, routers 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 routers 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 router 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 router has no interested receivers for the data, and the outgoing interface list becomes empty, the router 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.

## Basic PIM Network Components

PIM dense mode requires only a multicast source and series of multicast-enabled routers 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 routers called *rendezvous points* (*RPs*) in the network core. These routers 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 routers 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 router, as long as they are covered by a subnet that is connected to a bidirectional PIM-capable router and advertised to the network.

**Related Documentation**

- Supported IP Multicast Protocol Standards on page 31 in the Multicast Protocols Configuration Guide

## 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.
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.
7. (Optional) Configure the SAP and SDP protocols to listen for multicast session announcements. See "Configuring the Session Announcement Protocol" on page 563.
8. Configure IGMP. See "Configuring IGMP" on page 305.
9. (Optional) Configure the PIM static RP. See "Configuring Static RP" on page 91.
10. (Optional) Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 126 and "Example: Stopping Outgoing PIM Register Messages on a Designated Router" on page 122.
11. (Optional) Configure a PIM RPF routing table. See "Example: Configuring a PIM RPF Routing Table" on page 244.

**Related Documentation**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 3
- Verifying a Multicast Configuration on page 45

## IPv6 Multicast Flow

- IPv6 Multicast Flow Overview on page 18
- Multicast Listener Discovery (MLD) Overview on page 20

### 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.

## Multicast Listener Discovery (MLD) Overview

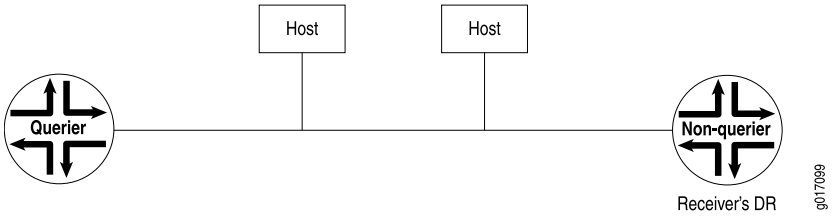
The Multicast Listener Discovery (MLD) protocol manages the membership of hosts and routers in multicast groups. IP version 6 (IPv6) multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. Each router maintains a list of host multicast addresses that have listeners for each subnetwork, as well as a timer for each address. However, the router does not need to know the address of the listeners—just the address of the hosts. The router provides addresses to the multicast routing protocol it uses; this 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 routers and hosts that need to receive IP multicast traffic. Junos OS supports MLD versions 1 and 2. Version 2 is supported for the source-specific multicast (SSM) include and exclude modes.

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

For each attached network, a multicast router can be either a querier or a nonquerier. A querier router, usually one per subnet, solicits group membership information by transmitting MLD queries. When a host reports to the querier router that it has interested listeners, the querier router forwards the membership information to the rendezvous point (RP) router 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 router. The RPT is the initial path used by the sender to transmit information to the interested listeners. Non-querier routers do not transmit MLD queries on a subnet but can trasmit them if the querier router goes down.

All MLD-configured routers start up as querier routers on each attached subnet. The non-querier router on the right is the receiver's DR.

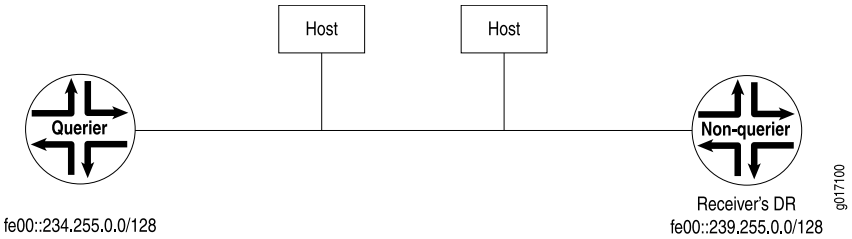



To elect the querier router, the routers exchange query messages containing their IPv6 source addresses. If a router hears a query message whose IPv6 source address is numerically lower than its own selected address, it becomes a non-querier. The router on the left has a source address numerically lower than the one on the right and therefore becomes the querier router.

In the practical application of MLD, several routers on a subnet are nonqueriers. If the elected querier router goes down, query messages are exchanged among the remaining routers. The router with the lowest IPv6 source address then becomes the new querier router. Note that 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.

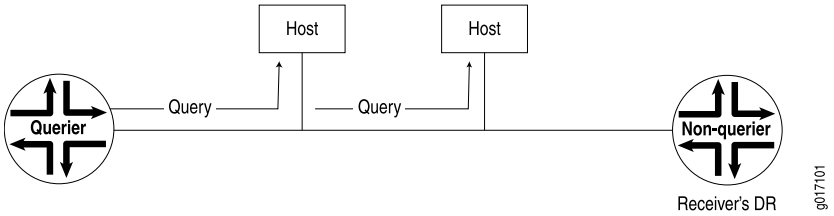
Configuration option for IPv6 Neighbor Discovery Protocol (NDP) is available. The configuration option is **set protocol neighbor-discovery onlink-subnet-only** command. This option will prevent the device from responding to a Neighbor Solicitation (NS) from a prefix which was not included as one of the device interface prefixes.

NOTE: **The Routing Engine needs to be rebooted after setting this option to remove any possibility of a previous IPv6 entry from remaining in the forwarding-table.**




The querier router 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. Within the query message is the *maximum response delay* value, specifying the maximum allowed delay for the host to respond with a report message.




**Related Documentation**

- Multicast Overview on page 3

CHAPTER 2

# Multicast VPNs Overview



## 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**

- Multicast over Layer 3 VPNs
- Junos OS VPNs Configuration Guide

## 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.

## 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.

## 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.

## 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 3 on page 28.

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.

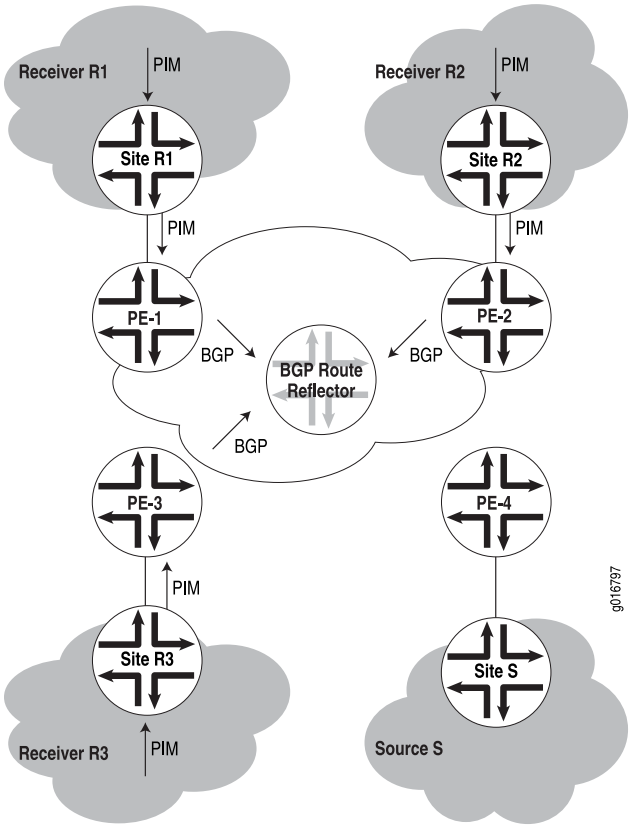
 

When router PE-1 receives a PIM join message for (S,G) from site router R1, this means that site R1 has one or more receivers for a given source and multicast group (S,G) combination. In that case, router PE-1 constructs and originates a customer multicast route after doing three things:

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 3: Source and Receiver Sites in an MVPN



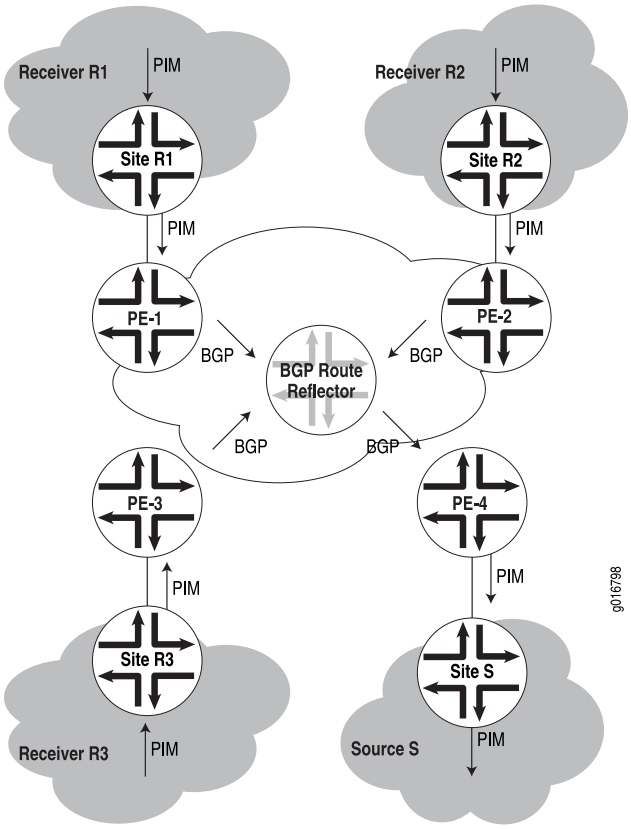

What happens when the source site S receives the MBGP information is shown in Figure 4 on page 29. In the figure, the customer multicast route information is distributed by the BGP route reflector as an MBGP update.

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 4: Adding a Receiver to an MVPN Source Site Using MBGP




**Related Documentation**

- Example: Configuring MBGP MVPN Extranets on page 442
- Examples: Configuring Multiprotocol BGP Multicast VPNs on page 383

CHAPTER 3

# Multicast Supported Standards



## 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 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*
- RFC 4601, *Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)*
- RFC 4607, *Source-Specific Multicast for IP*
- RFC 5015, *Bidirectional Protocol Independent Multicast (BIDIR-PIM)*
- *Using IGMPv3 and MLDv2 for Source-Specific Multicast*
- Internet draft draft-ietf-l3vpn-2547bis-mcast-10.txt, *Multicast in MPLS/BGP IP VPNs*
- Internet draft draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt, *BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs*
- Internet draft draft-ietf-pim-sm-bsr-05.txt, *Bootstrap Router (BSR) Mechanism for PIM*

The scoping mechanism is not supported.

- Internet draft draft-raggarwa-l3vpn-2547-mvpn-00.txt, *Base Specification for Multicast in BGP/MPLS VPNs* (expires December 2004)

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

# Configuration

CHAPTER 4

# Protocol-Independent Multicast



## Configuring Basic PIM Settings

## PIM Configuration Statements

To configure Protocol Independent Multicast (PIM), include the **pim** statement:

```
pim {
  disable;
  default-vpn-source {
    interface-name interface-name;
  }
  assert-timeout seconds;
  dense-groups {
    addresses;
  }
  dr-election-on-p2p;
  export;
  graceful-restart {
    disable;
    no-bidirectional-mode;
    restart-duration seconds;
  }
  idle-standby-path-switchover-delay seconds;
  import [ policy-names ];
  interface interface-name {
    bidirectional {
      df-election {
        backoff-period milliseconds;
        offer-period milliseconds;
        robustness-count number;
      }
    }
    import;
    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 {
    automatic;
  }
  join-prune-timeout;
  nonstop-routing {
    disable;
  }
```

```
override-interval milliseconds;
propagation-delay milliseconds;
reset-tracking-bit;
rib-group {
  inet group-name;
  inet6 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-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;
  }
  local {
    family (inet | inet6) {
      address address;
      anycast-pim {
        rp-set {
          address address <forward-msdp-sa>;
        }
        local-address address;
      }
      disable;
      group-ranges {
        destination-ip-prefix</prefix-length>;
      }
      hold-time seconds;
      override;
      priority number;
    }
  }
```

```
        rp-register-policy [ policy-names ];
        standby-path-creation-delay seconds;
        static {
          address address {
            override;
            version version;
            group-ranges {
              destination-ip-prefix</prefix-length>;
            }
            spt-threshold {
              infinity [ policy-names ];
            }
            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, PIM is disabled.

NOTE: You cannot configure PIM within a nonforwarding instance. If you try to do so, the router displays a commit check error and does not complete the configuration commit process.

## Changing the PIM Version

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 for rendezvous point (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.

To configure the PIM version, include the **version** statement:

```
version (1 | 2);
```

## 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
   ```

## 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
   ```

## 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).

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.

## 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. |
| **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. |

| Flag | Description |
|---|---|
| **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
   ```

## Disabling PIM

By default, when configured, the PIM protocol is enabled on all interfaces for all families. If desired, 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 43
- Disabling PIM On an Interface on page 44
- Disabling PIM for a Family on page 44
- Disabling PIM for a Rendezvous Point on page 45


## 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
   ```

## 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
   ```

## 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
   ```

### 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
   ```

## Verifying a Multicast Configuration

To verify a multicast configuration, perform these tasks:

- Verifying SAP and SDP Addresses and Ports on page 45
- Verifying the IGMP Version on page 46
- Verifying the PIM Mode and Interface Configuration on page 46
- Verifying the PIM RP Configuration on page 47
- Verifying the RPF Routing Table Configuration on page 47

### 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.

### 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.

### 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.

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring PIM Dense Mode on page 171
- Configuring a Designated Router for PIM on page 49
- Configuring PIM Filtering on page 118
- Configuring PIM Sparse-Dense Mode on page 174
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146
- Example: Configuring Nonstop Active Routing for PIM on page 158
- Examples: Configuring PIM RPT and SPT Cutover on page 131
- Examples: Configuring PIM Sparse Mode on page 51
- Configuring PIM Sparse-Dense Mode on page 174
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146

## 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 Configuration Guide
- Junos OS VPNs Configuration Guide

## 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 the lowest probability (priority 0) 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
```

## 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**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring PIM Dense Mode on page 171
- Configuring PIM Filtering on page 118
- Example: Configuring Nonstop Active Routing for PIM on page 158
- Examples: Configuring PIM RPT and SPT Cutover on page 131
- Examples: Configuring PIM Sparse Mode on page 51
- Configuring PIM Sparse-Dense Mode on page 174
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146
- Configuring Basic PIM Settings on page 35

## Examples: Configuring 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)`.

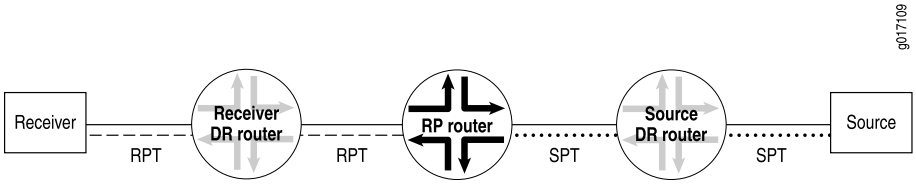
PIM sparse mode has standard features for all of these issues.

### 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 5 on page 53, the RP router is upstream from the receiver and thus forms one end of the rendezvous-point tree.

**Figure 5: 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.

## Designated Router

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 availble 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 4 on page 55 shows the complete matrix of IPv4 and IPv6 PIM Tunnel PIC requirements.

**Table 4: 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.

By default, PIM is disabled. 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.

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 for rendezvous point (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. The following example explicitly configures PIMv2 on the interfaces.

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 all version 2
   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 198.58.3.253 version 2
   user@host# set interface all mode sparse
   user@host# set interface all version 2
   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

## 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 join load balancing.

   ```
   [edit protocols pim rp]
   user@host# set static address 10.10.10.1
   user@host# set interface all mode sparse version 2
   user@host# set join-load-balance
   ```

   The static address is the address of the RP.

3. 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.

## 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 240 seconds.

## Example: Enabling Join Suppression

This example describes how to enable PIM join suppression.



## Requirements

Before you begin:

- Configure the router interfaces. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 55.

## 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:

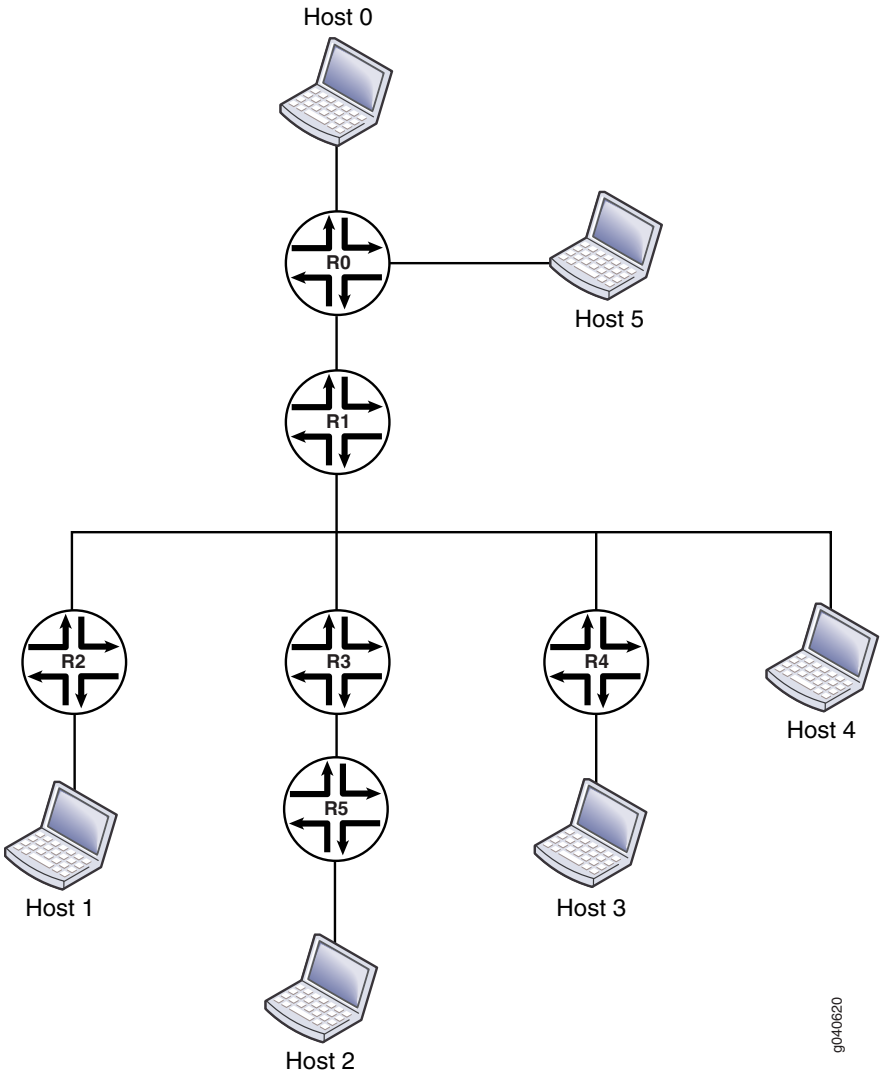
- **override-interval**—Sets the maximum time in milliseconds to delay sending override join messages. When a router sees a prune message for a join it is currently suppressing, it waits before it sends an override join message. Waiting helps avoid multiple downstream routers sending override join messages 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 6 on page 61 shows the topology used in this example.

Figure 6: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
[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**

## 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.

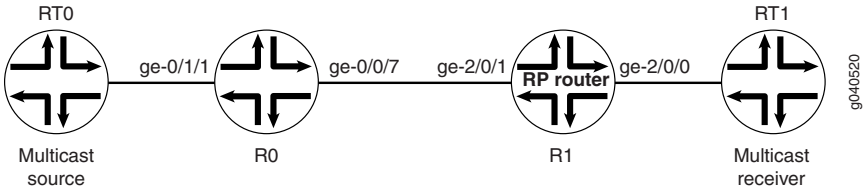
The configuration shown in this example works on the following platforms:

- 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 7 on page 64, 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 7: 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 interface sp-0/2/0.1
user@host# set interface ge-0/1/1.0
user@host# set interface 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 proposals 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 encryption-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
  "$9$nuDo6CuREyvWxO1LNbsZGFn/AOR8LNws4"
```

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 interfaces]
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 proposals 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 encryption-algorithm 3des-cbc
```

24. 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
  "$9$twR6pORrlMxNbhSds4aHkCtuBhr-dsoaU"
```

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
```

## 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 Configuration Guide.

### 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:

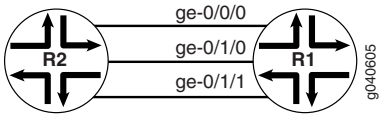
1. On R1, configure a virtual router instance with three interfaces (**ge-0/0/0.0**, **ge-0/1/0.0**, and **ge-0/1/1.0**).
2. Configure PIM and the RP.
3. Configure an MLD static group containing interfaces **ge-0/1/0.0** and **ge-0/1/1.0**.

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 8 on page 69 shows the topology for this example.

Figure 8: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
[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

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring PIM Dense Mode on page 171
- Configuring a Designated Router for PIM on page 49
- Configuring PIM Filtering on page 118
- Example: Configuring Nonstop Active Routing for PIM on page 158
- Examples: Configuring PIM RPT and SPT Cutover on page 131
- Configuring PIM Sparse-Dense Mode on page 174
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146
- Configuring Basic PIM Settings on page 35

## Example: Configuring Bidirectional PIM

- Understanding Bidirectional PIM on page 72
- Example: Configuring Bidirectional PIM on page 78

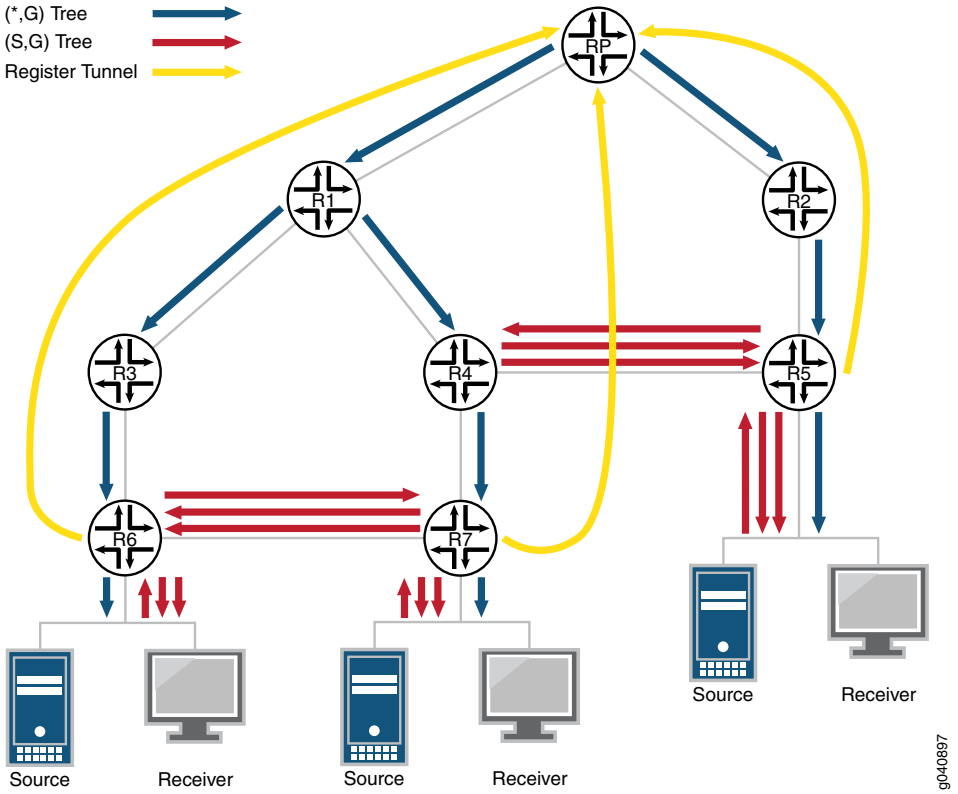
### Understanding Bidirectional PIM

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 9 on page 73 shows the traffic flows generated to deliver traffic for one group to and from three stations in a PIM-SM network.

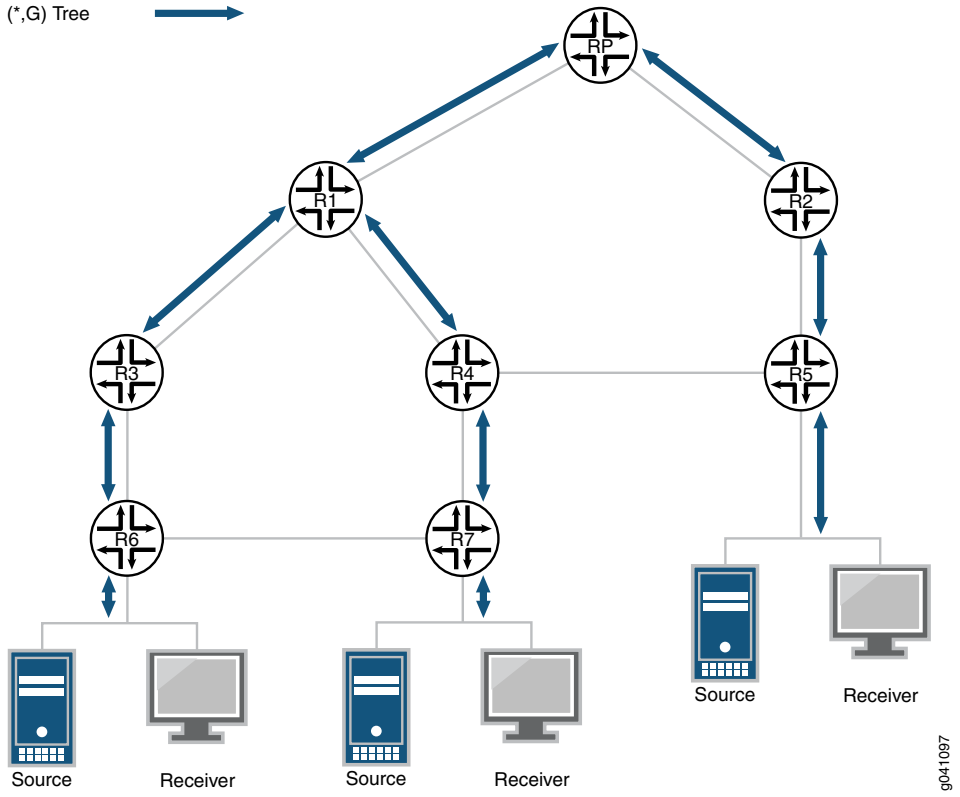
Figure 9: Example PIM Sparse-Mode Tree




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 10 on page 74 shows the traffic flows generated to deliver traffic for one group to and from three stations in a bidirectional PIM network.

Figure 10: 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

### Limitations of Bidirectional PIM

The Junos OS implementation of bidirectional PIM does not support the following functionality:

- 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).

NOTE: Starting with Release 12.2, Junos OS extends the nonstop active routing PIM support to draft-rosen MVPNs. 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. In releases earlier than Release 12.2, nonstop active routing PIM configuration was incompatible with draft-rosen MVPN configuration.

The bidirectional PIM protocol does not support the following functionality:

- Embedded RP
- Anycast RP

## 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.

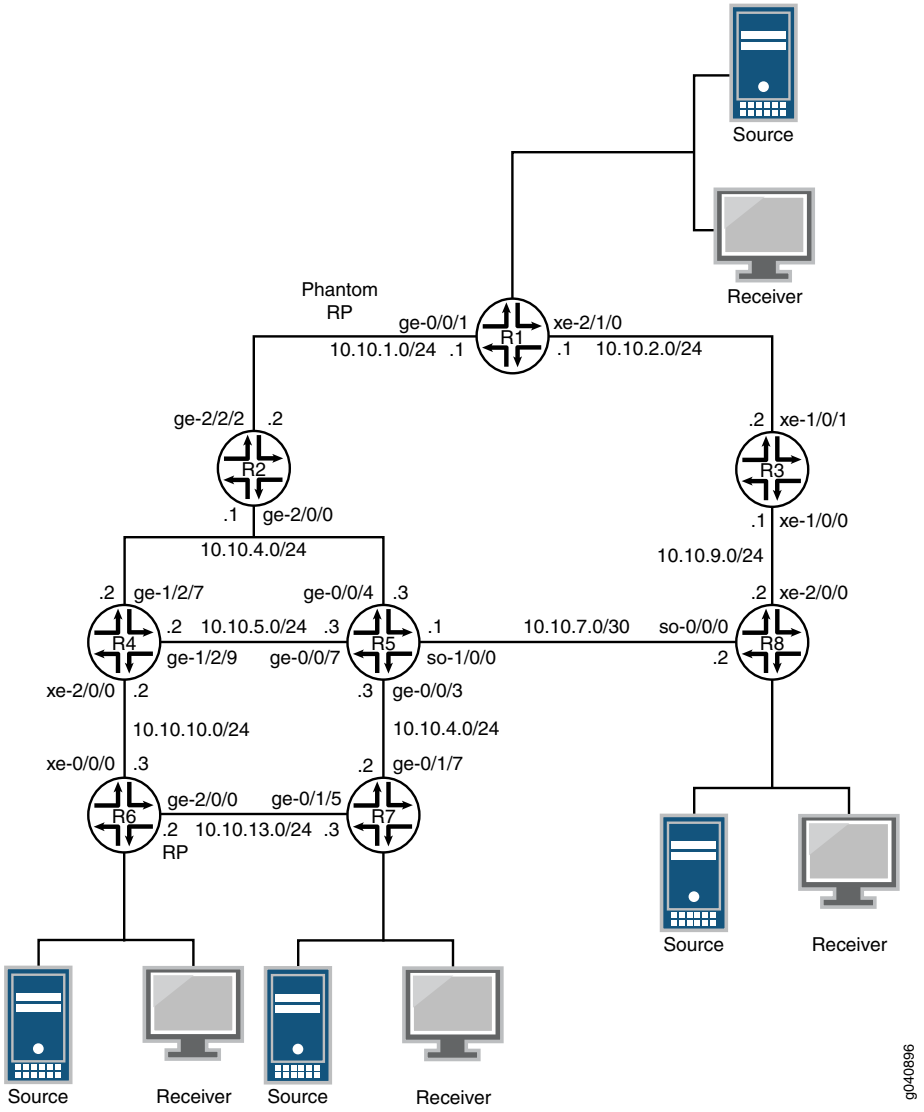
This example uses bidirectional-sparse-dense mode on the interfaces. The choice of PIM mode is closely tied to controlling how groups are mapped to PIM modes, as follows:

- **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 11 on page 80 shows the topology used in this example.

Figure 11: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

**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.

## 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.

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.

## 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.

## 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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 96
- Verifying the IGMP Version on page 96
- Verifying the PIM Mode and Interface Configuration on page 96

*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.

## 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.

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. 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
   ```

   NOTE: 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*]). Explicitly configured versions override the defaults.

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.

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring a Designated Router for PIM on page 49

## Example: Configuring Anycast RP



### 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 Internet draft draft-ietf-mboned-anycast-rp-08.txt, *Anycast RP Mechanism Using PIM and MSDP*. To access Internet RFCs and drafts, go to the IETF website at http://www.ietf.org .

### 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 55.

### 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

**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.

## 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.

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 **198.58.3.254** and the shared RP address is **198.58.3.253**. 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 198.58.3.254/32;
        primary;
        address 198.58.3.253/32;
      }
    }
  }
}
```

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 198.58.3.253;
      }
      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 198.58.3.250 {
      local-address address 198.58.3.254;
    }
  }
}
```

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 **198.58.3.254** and the shared RP address is **198.58.3.253**. 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 198.58.3.254/32 {
          primary;
        }
        address 198.58.3.253/32;
      }
    }
  }
}
```

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 198.58.3.253;
          anycast-pim {
            rp-set {
              address 198.58.3.240;
              address 198.58.3.241 forward-msdp-sa;
            }
            local-address 198.58.3.254; #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 198.58.3.253 {
          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 198.58.3.254/32 and the shared RP address is 198.58.3.253/32. 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 198.58.3.254/32 {
               primary;
            }
            address 198.58.3.253/32;
         }
      }
   }
}
```

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 198.58.3.253;
               anycast-pim {
                  rp-set {
                     address 198.58.3.240;
                     address 198.58.3.241 forward-msdp-sa;
                  }
                  local-address 198.58.3.254; #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.

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring a Designated Router for PIM on page 49
- Examples: Configuring PIM Sparse Mode on page 51
- Configuring Basic PIM Settings on page 35

## 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.

### 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 108 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.

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. See Configuring the Loopback Interface for information about configuring a loopback interface.

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 routers 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 routers by running the **show pim bootstrap** command.

## 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.

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. See Configuring the Loopback Interface for information about configuring a loopback interface.

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 routers 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 routers by running the **show pim bootstrap** command.

## 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 110
- Configuring a Designated Router for PIM on page 49
- Examples: Configuring PIM Sparse Mode on page 51
- Configuring Basic PIM Settings on page 35

## 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.

## 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.

Auto-RP operates in PIM version 1 and version 2.

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 5 on page 111 shows how the routing device behaves depending on the local RP configuration.

### Table 5: 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. |

Table 5: Local RP and Auto-RP Message Types *(continued)*

| Auto-RP Message Type | Local RP? | Routing Device Behavior |
|---|---|---|
| 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. The router ID is used as the address for auto-RP updates. 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**.

**Related Documentation**

- Configuring PIM Bootstrap Router on page 106
- Configuring a Designated Router for PIM on page 49
- Examples: Configuring PIM Sparse Mode on page 51
- Configuring Basic PIM Settings on page 35

## Configuring Embedded RP

- Understanding Embedded RP for IPv6 Multicast on page 115
- Configuring PIM Embedded RP for IPv6 on page 117

### 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).

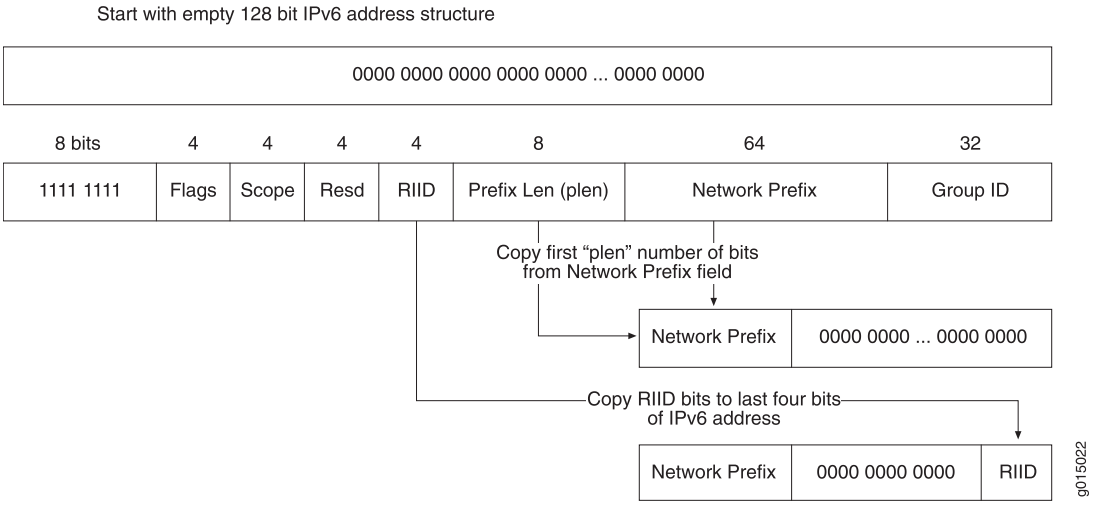
When the IPv6 address of the RP is embedded in a unicast-prefix-based any-source multicast (ASM) address, all of the following conditions must be true:

- 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 12 on page 116 for an illustration of this process.

### Figure 12: 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.

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring a Designated Router for PIM on page 49
- Examples: Configuring PIM Sparse Mode on page 51
- Configuring Basic PIM Settings on page 35

## 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.

## 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.

## 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.

## 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.

## 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.

## 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 126.

### 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 on page 124
- Verifying the IGMP Version on page 125
- Verifying the PIM Mode and Interface Configuration on page 125
- Verifying the PIM RP Configuration on page 125


### *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.

## 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 6 on page 125 for a list of match conditions.

### Table 6: 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.

## 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 563.
8. Configure IGMP. See "Configuring IGMP" on page 305.
9. Configure the PIM static RP. See "Configuring Static RP" on page 91.

### 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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 128
- Verifying the IGMP Version on page 128
- Verifying the PIM Mode and Interface Configuration on page 129
- Verifying the PIM Register Messages on page 129

### *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.

## 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**

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring PIM Dense Mode on page 171
- Configuring a Designated Router for PIM on page 49
- Example: Configuring Nonstop Active Routing for PIM on page 158
- Examples: Configuring PIM RPT and SPT Cutover on page 131
- Configuring PIM Sparse-Dense Mode on page 174
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146
- Configuring Basic PIM Settings on page 35

## 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.

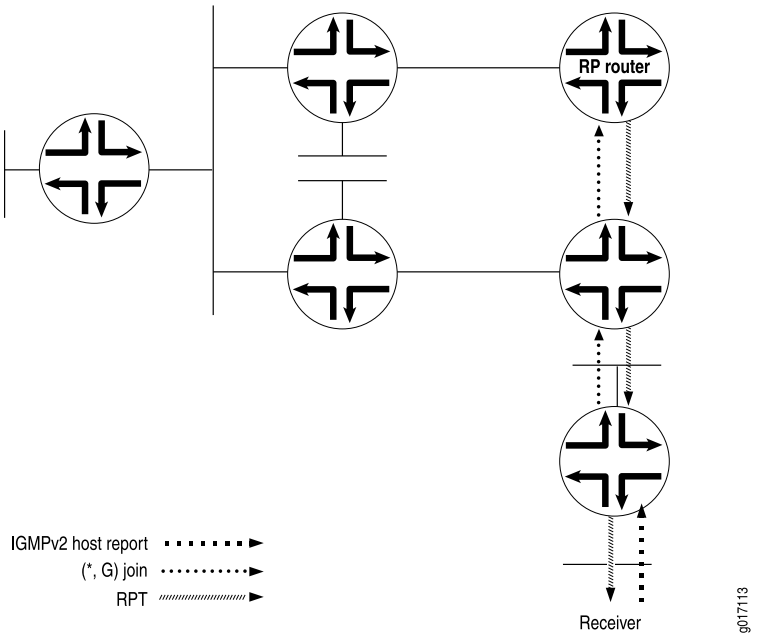
 

## Building an RPT Between the RP and Receivers

The RPT is the path between the RP and receivers (hosts) in a multicast group (see Figure 13 on page 133). 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 13: Building an RPT Between the RP and the Receiver



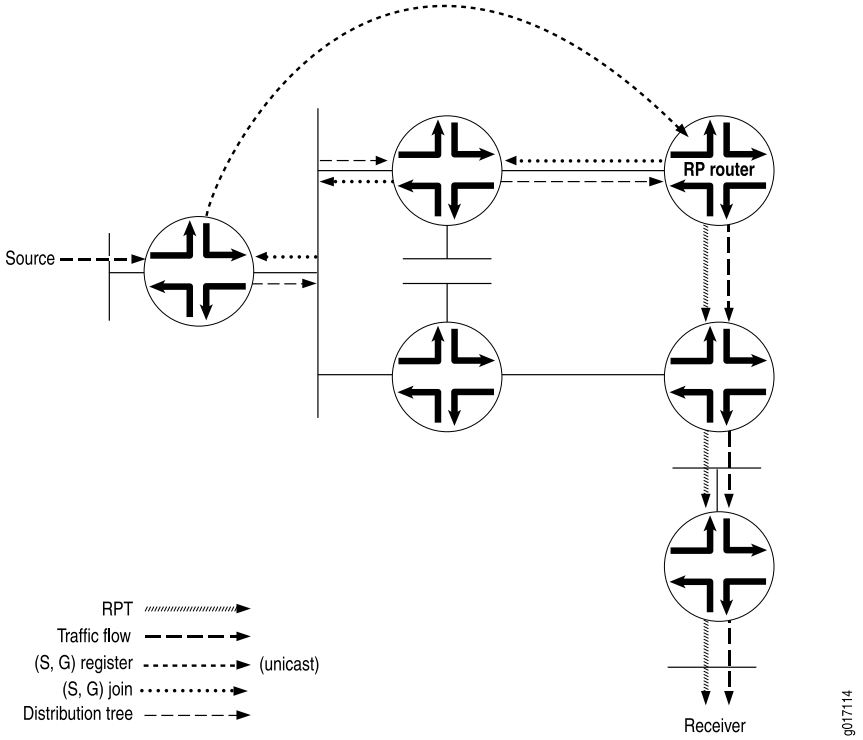

## PIM Sparse Mode Source Registration

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:

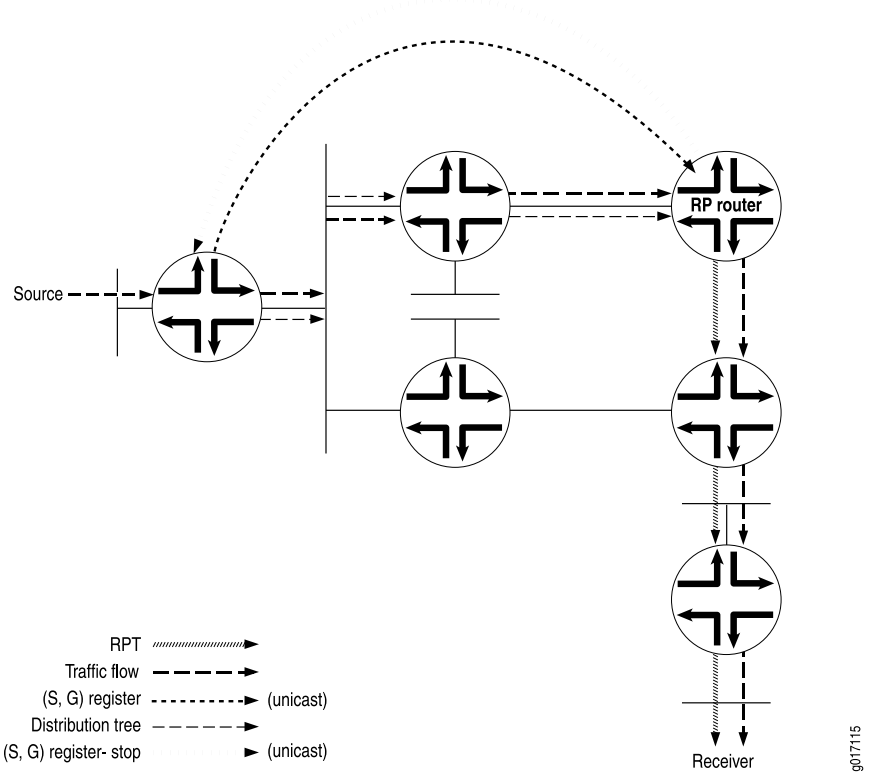
1. The source becomes active, sending out multicast packets on the LAN to which it is attached. The source's DR receives the packets and encapsulates them in a PIM register message, which it sends to the RP router (see Figure 14 on page 134).
2. When the RP router receives the PIM register message from the source, it sends a PIM join message back to the source.

Figure 14: PIM Register Message and PIM Join Message Exchanged



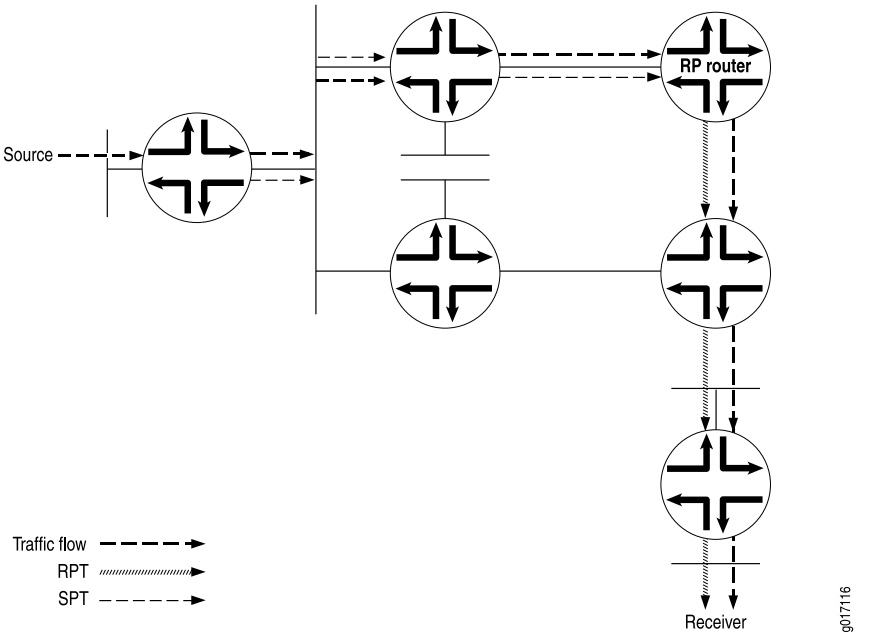

3. The source's DR receives the PIM join message and begins sending traffic down the SPT toward the RP router (see Figure 15 on page 135).
4. Once traffic is received by the RP router, it sends a register stop message to the source's DR to stop the register process.

Figure 15: 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 16 on page 135).

Figure 16: 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.

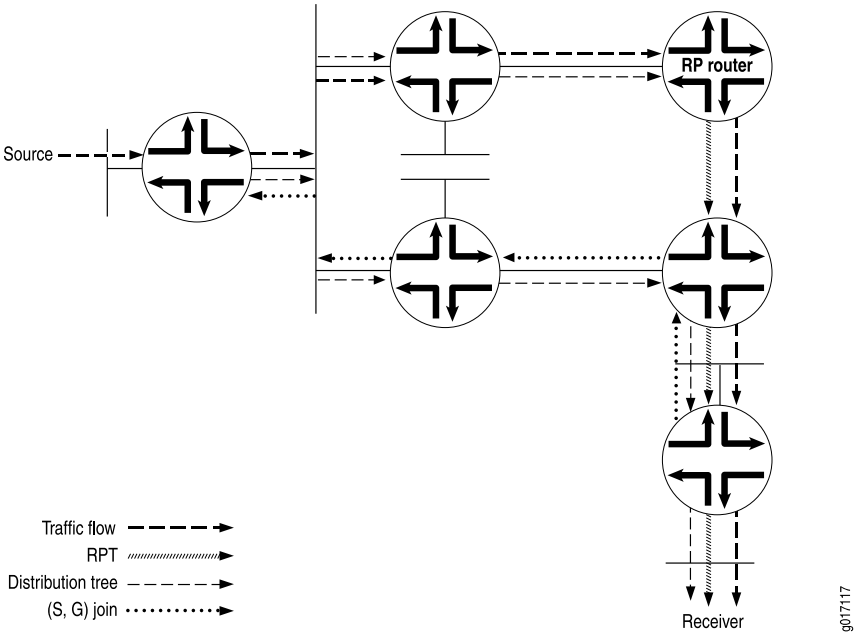
SPTs are always the shortest path, but they are not necessarily short. That is, sources and receivers tend to be on the periphery of a router network, not on the backbone, and multicast distribution trees have a tendency to sprawl across almost every router in the network. Because multicast traffic can 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.

## 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:

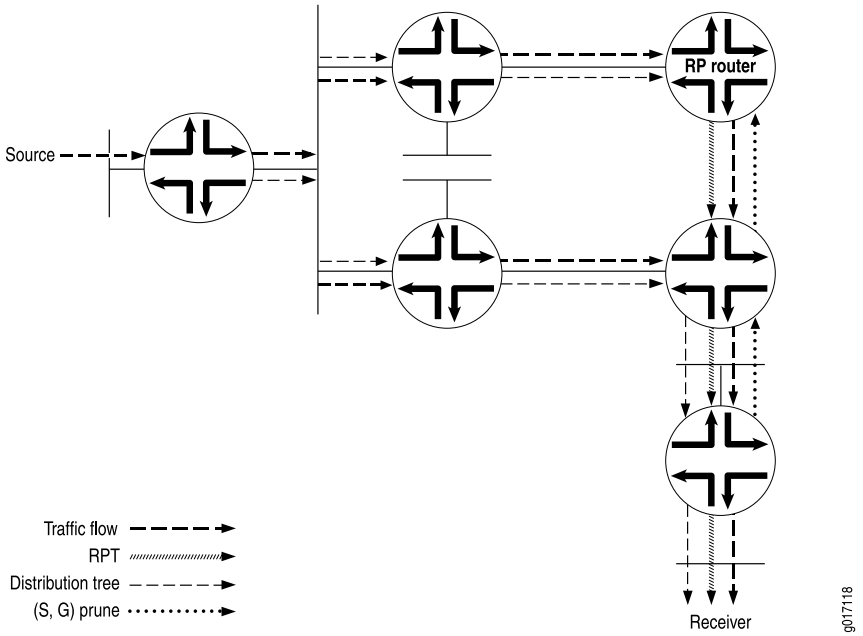
1. Once the receiver's DR receives the first multicast packet from the source, the DR sends a PIM join message to its RPF neighbor (see Figure 17 on page 137).
2. The source's DR receives the PIM join message, and an additional (S,G) state is created to form the SPT.
3. Multicast packets from that particular 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 17: Receiver DR Sends a PIM Join Message to the Source



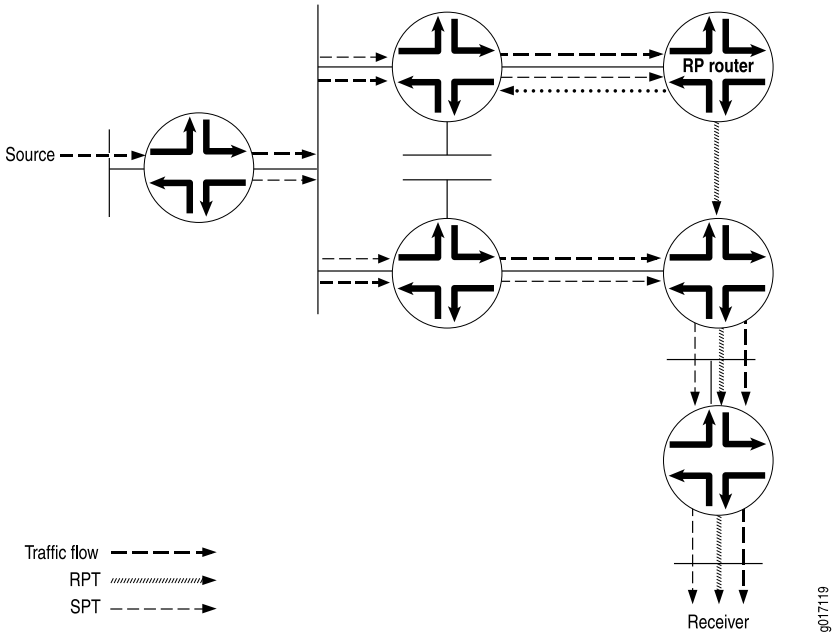

4. To stop duplicate multicast packets, the receiver's DR sends a PIM prune message toward the RP router, letting it know that the multicast packets from this particular source coming in from the RPT are no longer needed (see Figure 18 on page 138).

Figure 18: PIM Prune Message Is Sent from the Receiver's DR Toward the RP Router



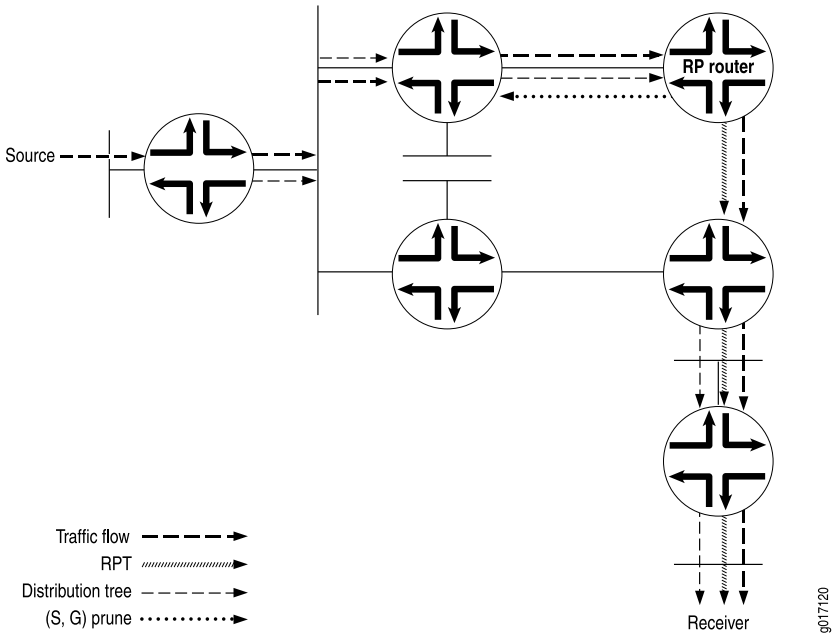

5. The PIM prune message is received by the RP router, and it stops sending multicast packets down to the receiver's DR. The receiver's DR is getting multicast packets only for this particular source over the new SPT. However, multicast packets from the source are still arriving from the source's DR toward the RP router (see Figure 19 on page 138).

Figure 19: RP Router Receives PIM Prune Message



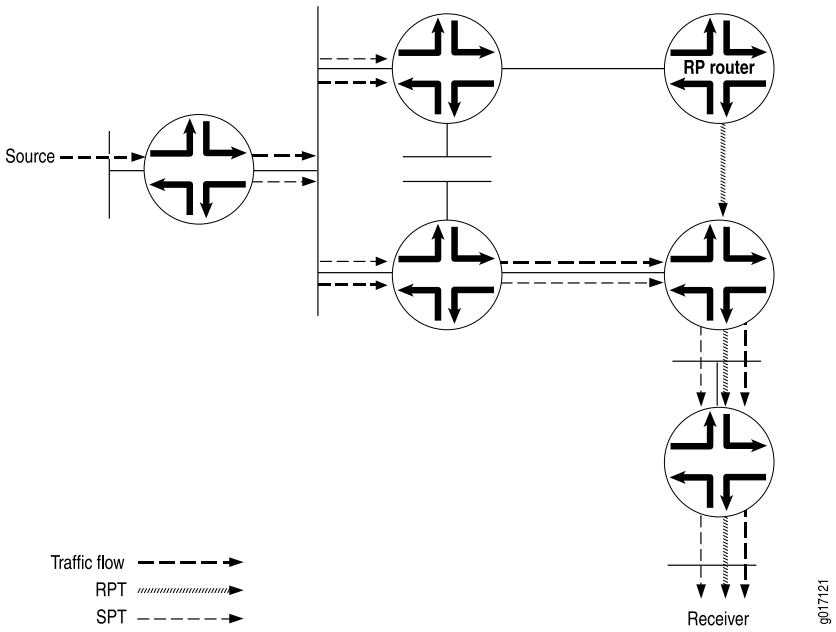
6. To stop the unneeded multicast packets from this particular source, the RP router sends a PIM prune message to the source's DR (see Figure 20 on page 139).

Figure 20: 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 21 on page 139).

Figure 21: 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 55.

### 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 22 on page 141, 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.

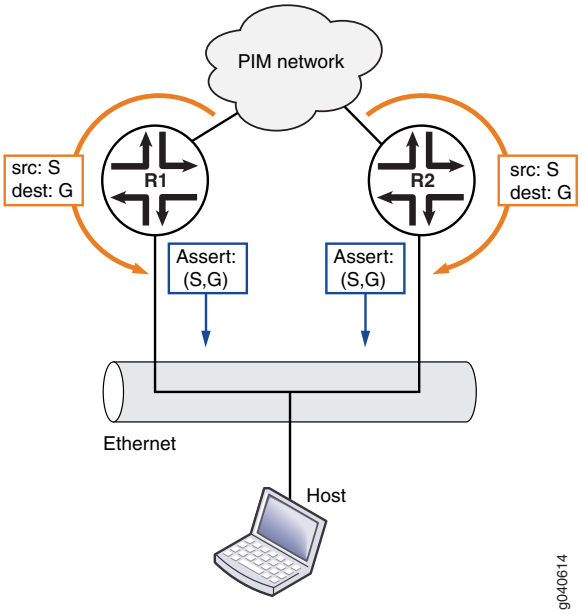
When an assert message is received and the RPF neighbor is changed to the assert winner, the assert timer is set to an assert timeout period. The assert timeout period is restarted every time a subsequent assert message for the route entry is received on the incoming interface. When the assert timer 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 22 on page 141 shows the topology for this example.

### Figure 22: 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

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 55.

## 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 SPT, 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
[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.

**Related Documentation**

- Configuring PIM Auto-RP on page 110
- Configuring PIM Bootstrap Router on page 106
- Configuring PIM Dense Mode on page 171
- Configuring a Designated Router for PIM on page 49
- Configuring PIM Filtering on page 118
- Example: Configuring Nonstop Active Routing for PIM on page 158
- Configuring PIM Sparse-Dense Mode on page 174
- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146
- Configuring Basic PIM Settings on page 35

## Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol

- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 146
- Configuring BFD for PIM on page 148
- Configuring BFD Authentication for PIM on page 150
- Example: Configuring BFD Liveness Detection for PIM IPv6 on page 152

### 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.

NOTE: **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.

### 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.

## 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 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 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 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 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 bfd-liveness-detection]
   user@host# set multiplier 50
   ```

7. Configure the BFD version.

   ```
   [edit protocols pim interface fe-1/0/0.0 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 bfd-liveness-detection]
   user@host# set no-adaptation
   ```

9. Verify the configuration by checking the output of the **show bfd session** command.

## Configuring BFD Authentication for PIM

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 steps are needed to configure authentication on a BFD session:

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.

The following sections provide instructions for configuring and viewing BFD authentication on PIM:


- Configuring BFD Authentication Parameters on page 150
- Viewing Authentication Information for BFD Sessions on page 151


### 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 if3-pim 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 if3-pim 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
     $9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm start-time 2009-06-14.10:00:00
   ```

4. (Optional) Specify loose authentication checking if you are transitioning from nonauthenticated sessions to authenticated sessions.

   ```
   [edit protocols pim]
   user@host# set interface if3-pim 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 **if3-pim** BGP group. 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 "**$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm**" and a start time of June 1, 2009, at 9:46:02 AM PST. Key **2** contains the secret data "**$9$a5jiKW9l.reP38ny.TszF2/9**" and a start time of June 1, 2009, at 3:29:20 PM PST.

```
[edit protocols pim]
interface if3–pim {
   bfd-liveness-detection {
      authentication {
         algorithm keyed-sha-1;
         key-chain bfd-pim;
      }
   }
}
[edit security]
authentication key-chains {
```

```
    key-chain bfd-pim {
      key 1 {
        secret "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm";
        start-time "2009-6-1.09:46:02 -0700";
      }
      key 2 {
        secret "$9$a5jiKW9l.reP38ny.TszF2/9";
        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
50.0.0.2                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
50.0.0.2                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
```

## 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

This example uses the following hardware and software components:

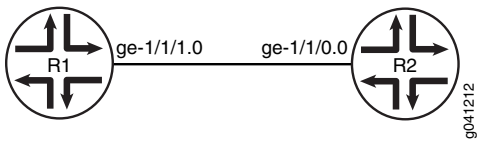
- Two peer routers.
- Junos OS 12.2 or later.

## Overview

In this example. Device R1 and Device R2 are peers. Each router runs PIM, connected over a common medium.

Figure 23 on page 153 shows the topology used in this example.

### Figure 23: 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-1/1/1 unit 0 description toRouter2
set interfaces ge-1/1/1 unit 0 family inet6
set interfaces ge-1/1/1 unit 0 family inet6 address e80::21b:c0ff:fed5:e4dd
set protocols pim interface ge-1/1/1 family inet6 bfd-liveness-detection authentication
  algorithm keyed-sha-1
set protocols pim interface ge-1/1/1 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
  "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm"
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
  "$9$a5jiKW9l.reP38ny.TszF2/9"
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
  "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm"
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
  "$9$a5jiKW9l.reP38ny.TszF2/9"
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-1/1/1 unit 0 description toRouter2
   user@R1# set ge-1/1/1 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-1/1/1.0 family inet6 bfd-liveness-detection
     authentication algorithm keyed-sha-1
   user@R1# set pim interface ge-1/1/1 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
     "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm"
   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 "$9$a5jiKW9l.reP38ny.TszF2/9"
   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-1/1/1 {
  unit 0 {
    description toRouter2;
    family inet6 {
      address e80::21b:c0ff:fed5:e4dd {
      }
    }
  }
}
```

```
user@R1# show protocols
pim {
  interface ge-1/1/1.0 {
    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 inet 6 {
              bfd-liveness-detection {
                authentication {
                  algorithm keyed-sha-1;
                  key-chain bfd-pim;
                }
              }
            }
          }
        }
      }
    }
```

```
user@R1# show security
authentication {
  key-chain bfd-pim {
    key 1 {
      secret "$9$ggaJDmPQ6/tJgF/AtREVsyPsnCtUHm";
      start-time "2012-01-01.09:46:02 -0700";
    }
    key 2 {
      secret "$9$a5jiKW9l.reP38ny.TszF2/9";
      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-1/1/1.0

        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.

**Related Documentation**

- Configuring Basic PIM Settings on page 35
- Example: Configuring BFD for BGP
- Example: Configuring BFD Authentication for BGP

## Example: Configuring Nonstop Active Routing for PIM

- Understanding Nonstop Active Routing for PIM on page 158
- Example: Configuring Nonstop Active Routing with PIM on page 159
- Configuring PIM Sparse Mode Graceful Restart on page 170

### 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**

## 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

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 306 and "Understanding MLD" on page 331.
- For this feature to work with IPv6, the routing device must be running Junos OS Release 10.4 or above.

## Overview

Junos OS supports nonstop active routing in the following PIM scenarios:

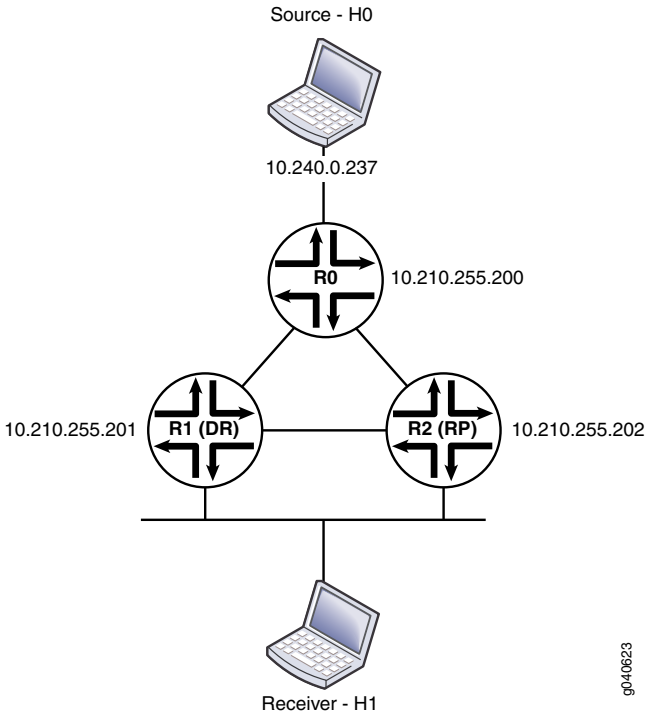
- Dense mode
- Sparse mode
- SSM
- Static RP
- Auto-RP (for IPv4 only)
- Bootstrap router
- Embedded RP on the non-RP router (for IPv6 only)
- BFD support

NOTE: **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 24 on page 161 shows the topology used in this example.

Figure 24: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

**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**

## 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.

**Related Documentation**

- Configuring Basic PIM Settings on page 35

## Configuring 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.

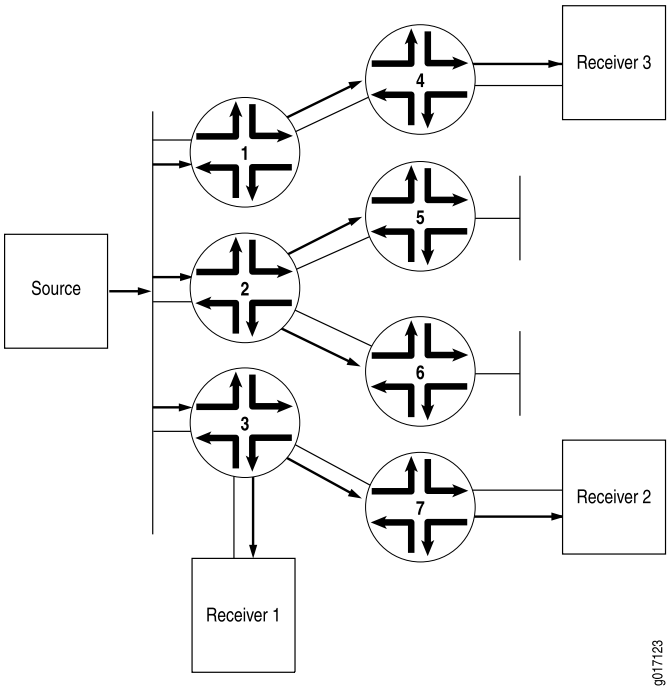
Internet service providers (ISPs) typically appreciate the ability to use any underlying unicast routing protocol with PIM dense mode because they do not need to introduce and manage a separate routing protocol just for RPF checks. While unicast routing

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.

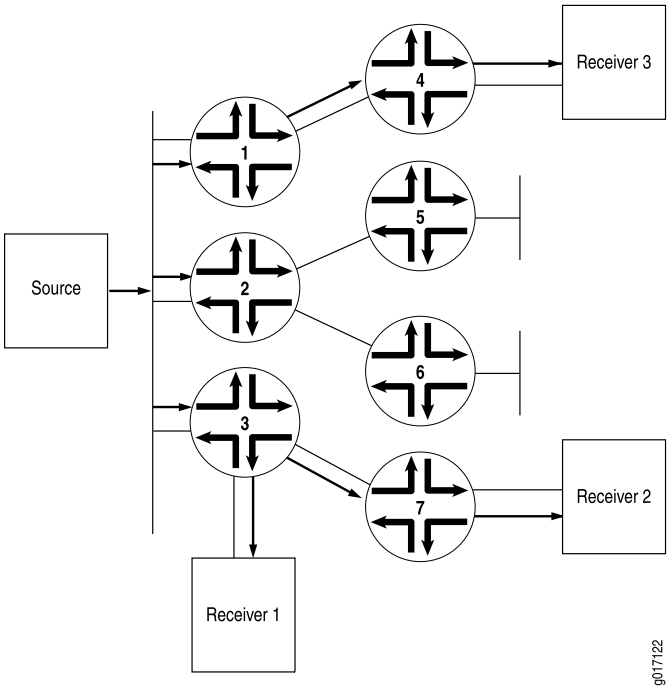
Unlike sparse mode, in which data is forwarded only to routers sending an explicit request, dense mode implements a *flood-and-prune* mechanism, similar to DVMRP. In PIM dense mode, there is no RP. A router receives the multicast data on the interface closest to the source, then forwards the traffic to all other interfaces (see Figure 25 on page 172).

### 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 router has no interested receivers for the data, and the OIL becomes empty, the router sends a prune message upstream to stop delivery of multicast traffic (see Figure 26 on page 173).

Figure 26: 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.

**Related Documentation**

- Configuring PIM Sparse-Dense Mode on page 174
- Configuring Basic PIM Settings on page 35

## 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 51 and "Understanding PIM Dense Mode" on page 171.

### 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 router, 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 router employing sparse-dense mode is a good example of mixing PIM modes on the same network or router 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.

**Related Documentation**

- Configuring PIM Dense Mode on page 171
- Configuring Basic PIM Settings on page 35

## 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.

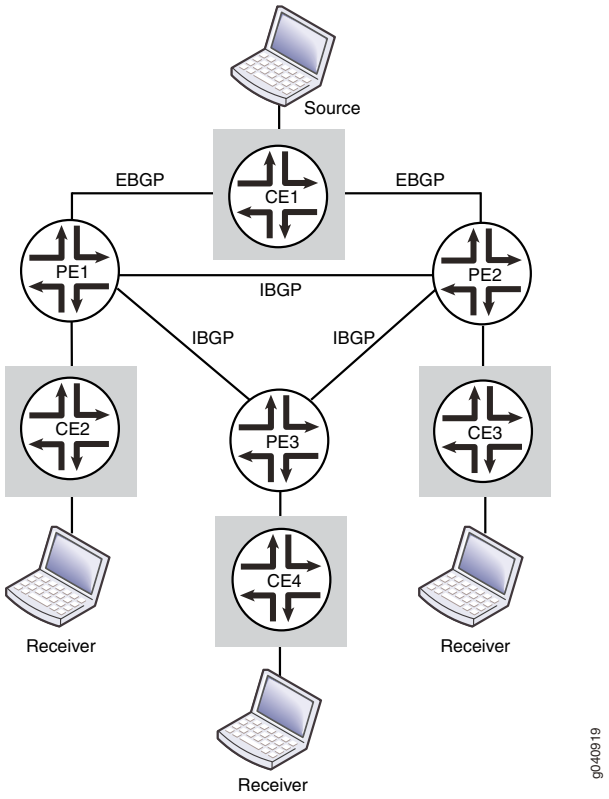
By default, a customer source (C-S) or a customer RP (C-RP) is considered remote if the active **rt_entry** is a secondary route and the primary route is present in a different routing instance. Such determination is being done without taking into consideration the (C-*,G) or (C-S,G) state for which 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 27 on page 177, 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 27: 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 27 on page 177, 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**

- Use Case for PIM Join Load Balancing

## 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 28 on page 183, 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 28 on page 183, 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 225.1.1.1, and Group 2 with group address 225.1.1.2), 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.

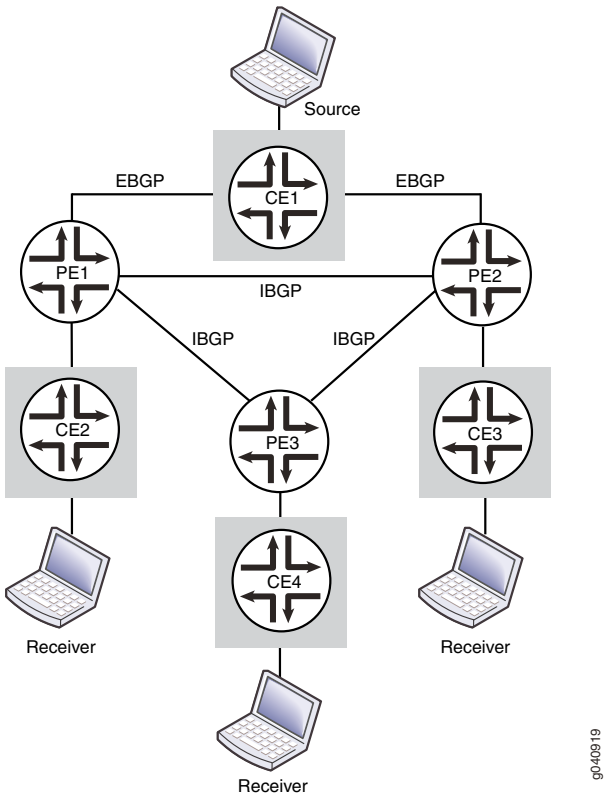
- If PE3 sends Group 1 join messages to PE1 and Group 2 join messages to PE2, there is no change in RPF neighbor. As a result, no join loops are created.
- If PE3 sends Group 1 join messages to PE2 and Group 2 join messages to PE1, there is a change in the RPF neighbor for the different groups resulting 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 28: 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 44.44.44.1
set routing-instances vpn1 protocols bgp group bgp family inet unicast
set routing-instances vpn1 protocols bgp group bgp neighbor 44.44.44.2 peer-as 3
set routing-instances vpn1 protocols bgp group bgp1 type external
set routing-instances vpn1 protocols bgp group bgp1 local-address 11.11.11.1
set routing-instances vpn1 protocols bgp group bgp1 family inet unicast
set routing-instances vpn1 protocols bgp group bgp1 neighbor 11.11.11.2 peer-as 4
set routing-instances vpn1 protocols pim vpn-group-address 224.1.1.1
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 224.1.1.1
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 44.44.44.1
```

```
user@PE1# set bgp group bgp family inet unicast
user@PE1# set bgp group bgp neighbor 44.44.44.2 peer-as 3
user@PE1# set bgp group bgp1 type external
user@PE1# set bgp group bgp1 local-address 11.11.11.1
user@PE1# set bgp group bgp1 family inet unicast
user@PE1# set bgp group bgp1 neighbor 11.11.11.2 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 224.1.1.1
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 44.44.44.1;
                    family inet {
                        unicast;
                    }
                    neighbor 44.44.44.2 {
                        peer-as 3;
                    }
                }
                group bgp1 {
                    type external;
                    local-address 11.11.11.1;
                    family inet {
```

```
            unicast;
          }
          neighbor 11.11.11.2 {
            peer-as 4;
          }
        }
      }
      pim {
        vpn-group-address 224.1.1.1;
        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 186


### 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: 225.1.1.1
     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: 225.1.1.2
     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: 225.1.1.3
    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: 225.1.1.4
    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: 225.1.1.1) and Group 3 (group address: 225.1.1.3) join messages, the PE1 router has selected the EBGP path toward the CE1 router to send the join messages.
- For Group 2 (group address: 225.1.1.2) and Group 4 (group address: 225.1.1.4) 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 175
- Example: Configuring PIM Join Load Balancing On Next-Generation Multicast VPN on page 187

## 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 29 on page 191, 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.

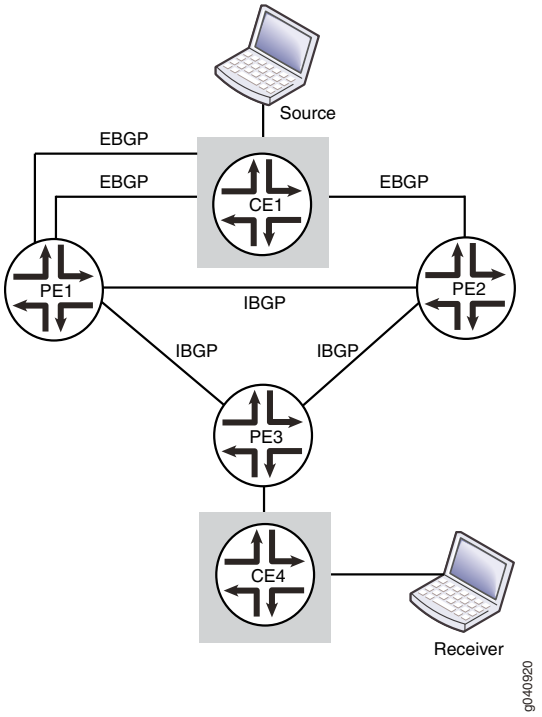
   In Figure 29 on page 191, assuming that the CE4 host is interested in receiving traffic from the Source, and CE4 initiates source join messages for different groups (Group 1 [C-S,C-G1] and Group 2 [C-S,C-G2]), the source join messages arrive on the PE3 router.

   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 29: 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, 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-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 195

### 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> 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
Family : INET

Instance : vpn1
  MVPN Mode : RPT-SPT
  C-mcast IPv4 (S:G)            Ptnl                              St
  0.0.0.0/0:225.1.1.1/32        RSVP-TE P2MP:10.255.10.2, 5834,10.255.10.2
  4.4.4.2/32:225.1.1.1/32       RSVP-TE P2MP:10.255.10.2, 5834,10.255.10.2
  0.0.0.0/0:225.1.1.2/32        RSVP-TE P2MP:10.255.10.14, 47575,10.255.10.14
  4.4.4.2/32:225.1.1.2/32       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):
  - 4.4.4.2/32:225.1.1.1/32 (S,G1) toward the PE1 router (10.255.10.2 is the loopback address of Router PE1).
  - 4.4.4.2/32:225.1.1.2/32 (S,G2) toward the PE2 router (10.255.10.14 is the loopback address of Router PE2).
- For shared join messages (*,G):
  - 0.0.0.0/0:225.1.1.1/32 (*,G1) toward the PE1 router (10.255.10.2 is the loopback address of Router PE1).
  - 0.0.0.0/0:225.1.1.2/32 (*,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 175
- Example: Configuring PIM Join Load Balancing on Draft-Rosen Multicast VPN on page 179

## Example: Configuring PIM Make-Before-Break Join Load Balancing

- Understanding the PIM Automatic Make-Before-Break Join Load-Balancing Feature on page 196
- Example: Configuring PIM Make-Before-Break Join Load Balancing on page 197

### 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.

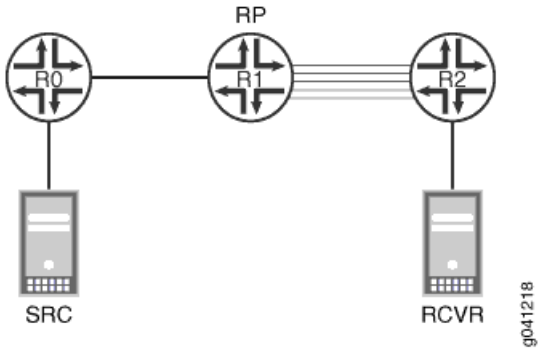
#### *Topology*

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 30 on page 198 shows the topology used in this example.

Figure 30: 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 on page 202
- Verifying PIM on page 203
- Verifying the PIM Automatic MBB Join Load-Balancing Feature on page 205


### *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.

## 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 7 on page 208 describes the different types of PIM system messages that you might see depending on your system log warning and log interval configurations.

**Table 7: 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. |

Table 7: PIM System Log Messages *(continued)*

| System Log Message | Definition |
|---|---|
| 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. |

## 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.

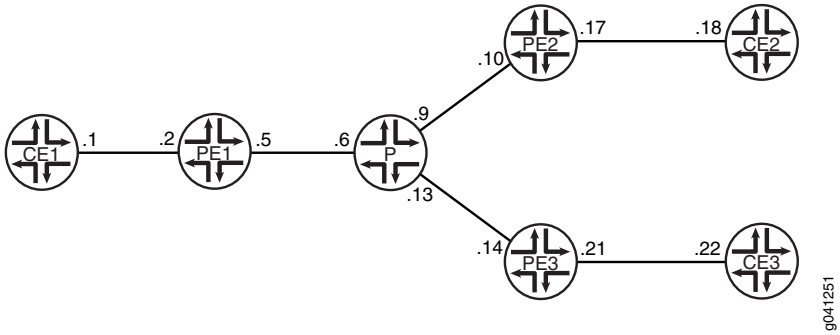
The **group-rp-mapping maximum** statement configures a limit for the number of group-to-RP mappings allowed in the vpn-1 routing instance.

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 31 on page 210 shows the topology used in this example.

Figure 31: PIM State Limits Topology




"CLI Quick Configuration" on page 210 shows the configuration for all of the devices in Figure 31 on page 210. The section "Step-by-Step Procedure" on page 213 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 1.1.1.1/32
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 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.1
```

**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 1.1.1.2/32
set interfaces lo0 unit 102 family inet address 100.1.1.2/32
set protocols mpls interface ge-1/2/1.5
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-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 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 100.1.1.2
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 1.1.1.2
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 1.1.1.3/32
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 1.1.1.3
```

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 1.1.1.4/32
set interfaces lo0 unit 104 family inet address 100.1.1.4/32
set protocols mpls interface ge-1/2/0.10
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-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 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 100.1.1.2
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 1.1.1.4
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 1.1.1.5/32
set interfaces lo0 unit 105 family inet address 100.1.1.5/32
set protocols mpls interface ge-1/2/0.14
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.4
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 100.1.1.2
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 1.1.1.5
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 1.1.1.6/32
set protocols sap listen 224.1.1.1
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 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.6
```

Device CE3

```
set interfaces ge-1/2/0 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 1.1.1.7/32
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 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.7
```

**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 1.1.1.2/32
user@PE1# set lo0 unit 102 family inet address 100.1.1.2/32
```

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 1.1.1.2
user@PE1# set family inet-vpn any
user@PE1# set family inet-mvpn signaling
user@PE1# set neighbor 1.1.1.4
user@PE1# set neighbor 1.1.1.5
```

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 100.1.1.2
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 1.1.1.2
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 1.1.1.2/32;
        }
    }
    unit 102 {
        family inet {
            address 100.1.1.2/32;
        }
    }
}

user@PE1# show protocols
mpls {
```

```
    interface ge-1/2/1.5;
}
bgp {
    group ibgp {
        type internal;
        local-address 1.1.1.2;
        family inet-vpn {
            any;
        }
        family inet-mvpn {
            signaling;
        }
        neighbor 1.1.1.4;
        neighbor 1.1.1.5;
    }
}
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 100.1.1.2;
          }
        }
        interface ge-1/2/0.2 {
          mode sparse;
        }
      }
      mvpn;
    }
  }
```

```
user@PE1# show routing-options
router-id 1.1.1.2;
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.

**Related Documentation**

- Limiting the Number of IGMP Multicast Group Joins on Logical Interfaces on page 323
- Examples: Configuring the Multicast Forwarding Cache on page 286

## PIM Snooping for VPLS



### 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.

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 220 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.

## 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:

- MX Series 3D Universal Edge Routers
- Junos OS Release 12.3 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.

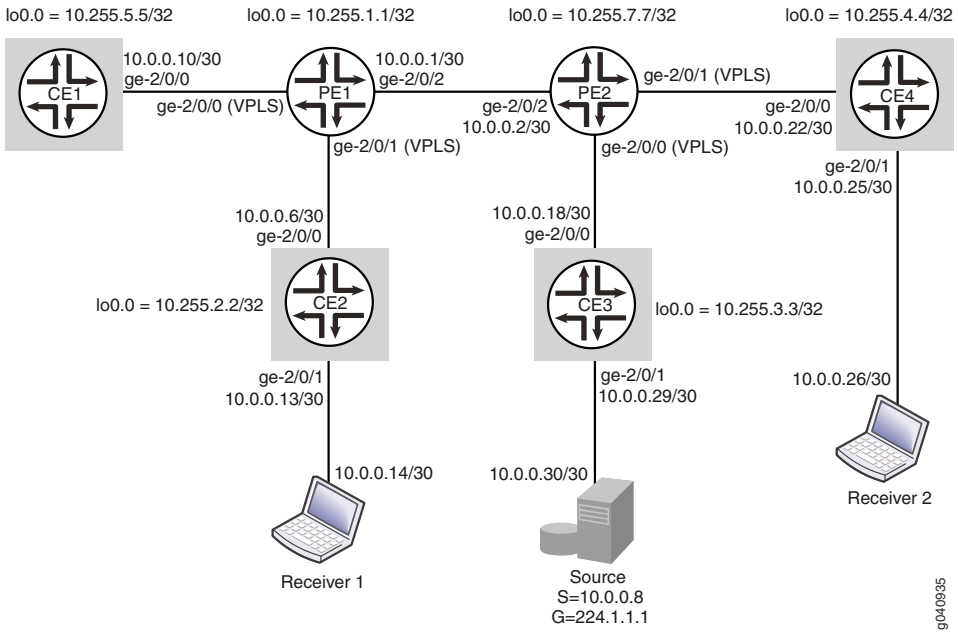
#### *Topology*

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 32 on page 221 shows the topology used in this example.

Figure 32: 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 221.

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 Configuring VPLS Encapsulation on CE-Facing Interfaces 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 227


### *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 224.1.1.1.

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: 224.1.1.1
    Source: *
    Flags: sparse,rptree,wildcard
    Upstream neighbor: 10.0.0.18, Port: ge-2/0/0.0

Group: 224.1.1.1
    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: 224.1.1.1
    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: 224.1.1.1
    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 224.1.1.1 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 224.1.1.1
Nexthop Bulking: OFF

             Family: INET

Group: 224.1.1.1/32
    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: 224.1.1.1/32
    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.

CHAPTER 5

# 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.

## 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.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.

### 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.

## 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 Configuration Guide.

## 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.

## 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 265
- Examples: Configuring the Multicast Forwarding Cache on page 286

# Examples: Configuring Reverse Path Forwarding

- Understanding Multicast Reverse Path Forwarding on page 239
- Multicast RPF Configuration Guidelines on page 241
- Example: Configuring a Dedicated PIM RPF Routing Table on page 241
- Example: Configuring a PIM RPF Routing Table on page 244
- Example: Configuring RPF Policies on page 248
- Example: Configuring PIM RPF Selection on page 250

## 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.

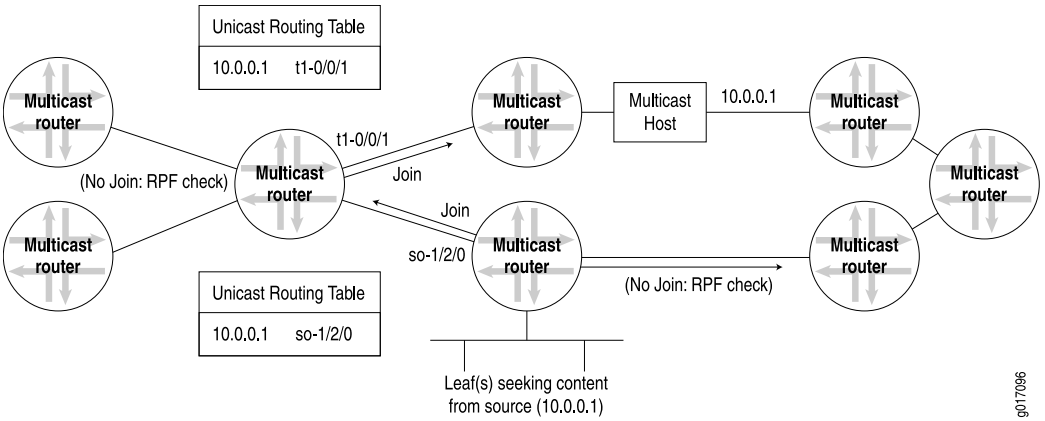
In multicast, the router forwards the packet away from the source to make progress along the distribution tree and prevent routing loops. The router's multicast forwarding state runs more logically by organizing tables based on the reverse path, from the receiver back to the root of the distribution 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 33 on page 239 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 33: 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.

By default, PIM uses **inet.0** as its reverse-path forwarding (RPF) routing table group. PIM uses an RPF routing table group to resolve its RPF neighbor for a particular multicast source address and for the RP address. PIM can optionally use **inet.2** as its RPF routing table group. The **inet.2** routing table is organized more efficiently for RPF checks.

Once configured, the RPF routing table must be applied to a PIM as a routing table group.

## 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 239.

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.

## 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 Junos® OS Network Interfaces.
- Enable PIM. See "PIM Overview" on page 15.

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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set routing-options rib-groups mcast-rpf-rib import-rib inet.2
set protocols pim rib-group mcast-rpf-rib
set rib-group inet if-rib
```

```
set routing-options interface-routes 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 The 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.

## 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 563.
8. Configure IGMP. See "Configuring IGMP" on page 305.
9. Configure the PIM static RP. See "Configuring Static RP" on page 91.
10. Filter PIM register messages from unauthorized groups and sources. See "Example: Rejecting Incoming PIM Register Messages on RP Routers" on page 126 and "Example: Stopping Outgoing PIM Register Messages on a Designated Router" on page 122.

### 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 247
- Verifying the IGMP Version on page 247
- Verifying the PIM Mode and Interface Configuration on page 247
- Verifying the PIM RP Configuration on page 248
- Verifying the RPF Routing Table Configuration on page 248

### *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.

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.

### 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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.

## 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. See the Junos® OS Network Interfaces.

- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- 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 rendezvous 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.

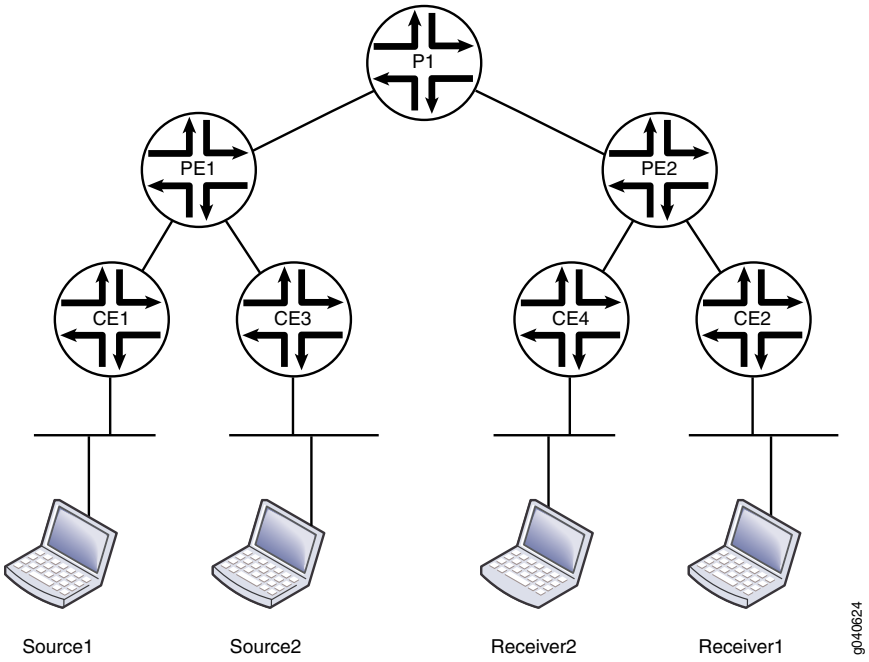
You can use this static RPF selection as a building block for particular applications. For example, an extranet. Suppose you want to split the multicast flows among parallel PIM links or assign one multicast flow to a specific PIM link. With static RPF selection configured, the router sends join and 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 34 on page 252 shows the topology used in this example.

Figure 34: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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

**Related Documentation**

- Example: Configuring Ingress PE Redundancy on page 294

## Example: Configuring Source-Specific Multicast

- Understanding PIM Source-Specific Mode on page 254
- PIM SSM on page 255
- Source-Specific Multicast Groups Overview on page 257
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 258
- Example: Configuring an SSM-Only Domain on page 261
- Example: Configuring PIM SSM on a Network on page 262
- Example: Configuring SSM Mapping on page 263

### Understanding PIM Source-Specific Mode

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.

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. The RP routers must be added and must know all multicast sources, and complicated shared distribution trees must be built to the RPs.

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.

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 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 8 on page 255.

### Table 8: 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.

## PIM SSM

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.

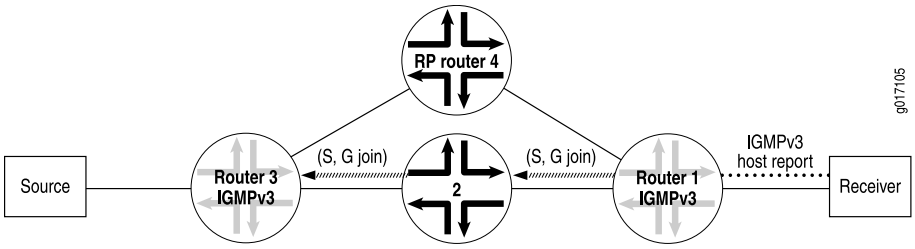
You can also configure the Junos OS to accept any-source multicast (ASM) join messages (*,G) for group addresses that are within the default or configured range of source-specific multicast (SSM) groups. This allows you to support a mix of any-source and source-specific multicast groups simultaneously.

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.

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.

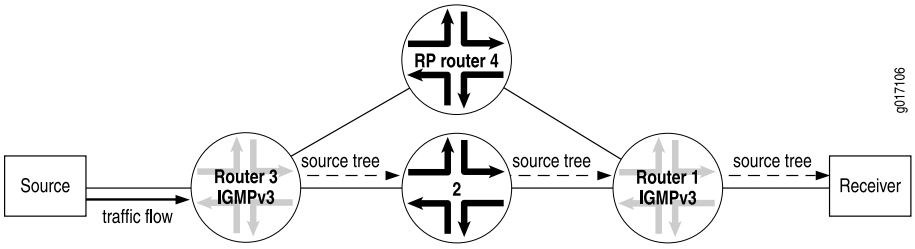
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3), announcing a desire to join group G and source S (see Figure 35 on page 256). The directly connected PIM sparse-mode router, the receiver's DR, sends an (S,G) join message to its RPF neighbor for the source. Notice in Figure 35 on page 256 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

### Figure 35: Receiver Announces Desire to Join Group G and Source S



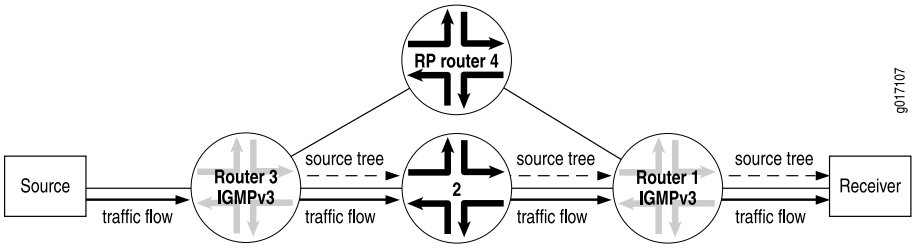

The (S,G) join message initiates the source tree and then builds it out hop by hop until it reaches the source. In Figure 36 on page 256, the source tree is built across the network to Router 3, the last-hop router connected to the source.

### Figure 36: Router 3 (Last-Hop Router) Joins the Source Tree




Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 37 on page 257).

Figure 37: (S,G) State Is Built Between the Source and the Receiver




To configure additional SSM groups, include the **ssm-groups** statement at the **[edit routing-options multicast]** hierarchy level.

## 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

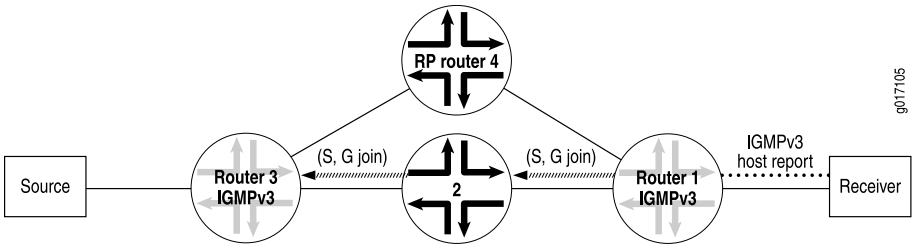
Before you begin, configure the router interfaces. See the Junos® OS Network Interfaces.

### 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.

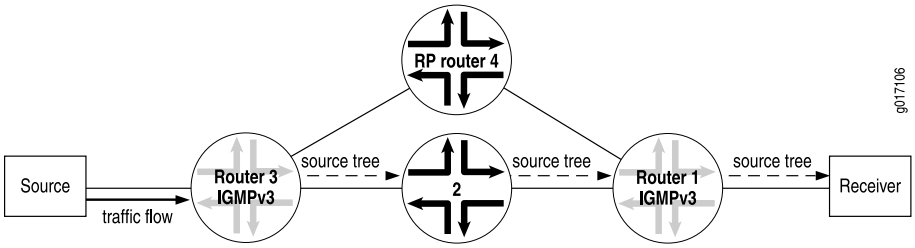
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3 or MLDv2) to join group G and source S (see Figure 38 on page 258). The directly connected PIM sparse-mode router, the receiver's designated router (DR), sends an (S,G) join message to its reverse-path forwarding (RPF) neighbor for the source. Notice in Figure 38 on page 258 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

**Figure 38: Receiver Sends Messages to Join Group G and Source S**



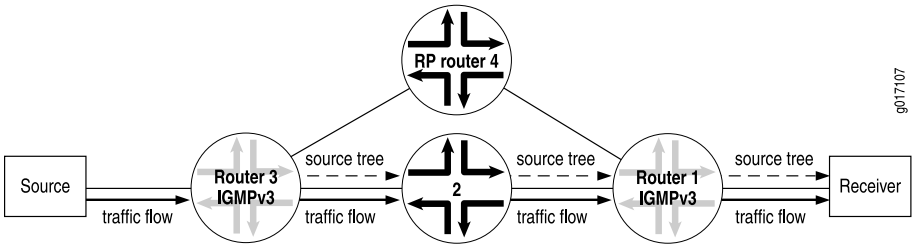

The (S,G) join message initiates the source tree and then builds it out hop by hop until it reaches the source. In Figure 39 on page 259, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 39: Router 3 (Last-Hop Router) Joins the Source Tree




Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 40 on page 259).

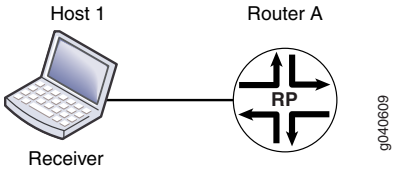
Figure 40: (S,G) State Is Built Between the Source and the Receiver




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 41 on page 259.

Figure 41: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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

## 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**:

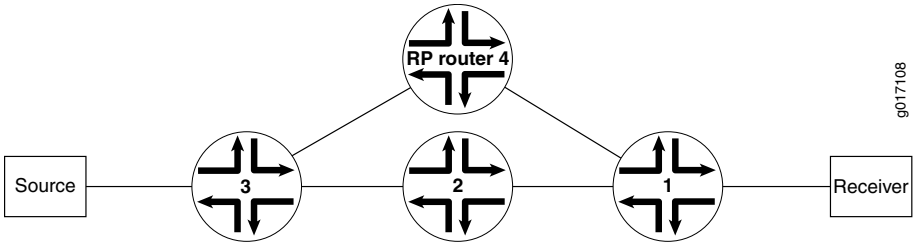
```
[edit]
protocols {
  pim {
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
  igmp {
    interface fe-0/1/2 {
      version 3;
    }
```

```
    }
}
```

## 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 42 on page 262.

### Figure 42: 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       198.58.3.245     213       3      0
   fe-0/0/1.0     Up       198.58.3.241     220       3      0
   fe-0/0/2.0     Up       198.58.3.237     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
```

## 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 on page 35

## Examples: Configuring Bandwidth Management

- Understanding Bandwidth Management for Multicast on page 266
- Bandwidth Management and PIM Graceful Restart on page 266
- Bandwidth Management and Source Redundancy on page 266
- Logical Systems and Bandwidth Oversubscription on page 267
- Example: Defining Interface Bandwidth Maximums on page 268
- Example: Configuring Multicast with Subscriber VLANs on page 270

- Configuring Multicast Routing Over IP Demux Interfaces on page 283
- Classifying Packets by Egress Interface on page 284

## 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 266
- Bandwidth Management and Source Redundancy on page 266
- Logical Systems and Bandwidth Oversubscription on page 267

## 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.

## 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 290.

## 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.

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol. See the Junos OS Routing Protocols Configuration Guide.
- 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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.

## 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 Configuration Guide.
- 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.

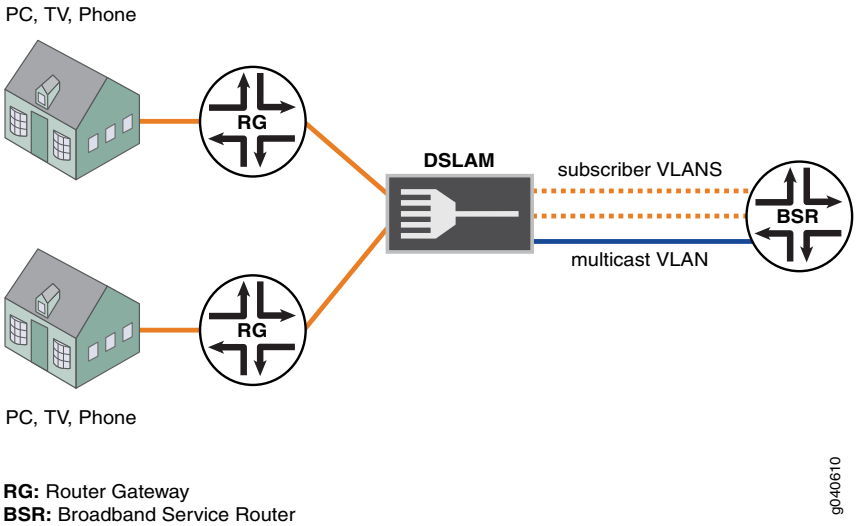
Figure 43 on page 274 shows the scenario.

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 43: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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

## 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. See Configuring Dynamic Subscriber Interfaces Using IP Demux Interfaces in Dynamic Profiles for details.
- 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.**

## 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 will not perform any classification based on the egress interface.

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-class-map  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 Forwarding Classes.

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 231
- Examples: Configuring the Multicast Forwarding Cache on page 286

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol. See the Junos OS Routing Protocols Configuration Guide.
- 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.

  Beginning with Junos OS 12.2, you can optionally configure a warning threshold so the device can log warning messages in the system log when a certain number of entries have been added to the cache. It is helpful to review the system log messages for troubleshooting purposes and to detect if an excessive amount of entries are filling up the cache. When the number of entries exceeds the configured warning threshold, but remains below the suppress threshold, traffic continues to be accepted, and the device logs warning messages in the system log.

The warning threshold is a percentage of the suppress threshold, so you must configure the suppress threshold to configure the warning threshold. The range for the warning threshold is 1 through 100 percent. In this example, you set a warning threshold of 40 percent. With a configured suppress value of 150,000, and a warning threshold of 40 percent, the device logs a warning message in the system log after it receives 60,000 entries.

## 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 forwarding-cache threshold suppress 150000
set routing-options multicast forwarding-cache threshold reuse 70000
set routing-options multicast forwarding-cache threshold log-warning 40
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
   ```

4. (Optional) Configure when warning messages are logged in the system log.

   ```
   [edit routing-options multicast forwarding-cache]
   user@host# set threshold log-warning 40
   ```

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

   ```
   [edit routing-options multicast forwarding-cache]
   user@host# commit
   ```

***Results***

Confirm your configuration by entering the **show routing-options** command.

```
user@host# show routing-options
```

```
multicast {
  forwarding-cache {
    threshold {
      suppress 150000;
      reuse 70000;
      log-warning 40;
    }
    timeout 60;
  }
}
```

## Verification

Confirm that the configuration is working properly.

- Displaying Entries in the IP Multicast Forwarding Table on page 289
- Verifying the IP Multicast Forwarding Cache Configuration on page 289

### *Displaying Entries in the IP Multicast Forwarding Table*

**Purpose** Verify that entries are in the IP multicast forwarding table and that the configured timeout value is displayed.

**Action** From operational mode, enter 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
```

### *Verifying the IP Multicast Forwarding Cache Configuration*

**Purpose** Verify that the suppress threshold, reuse value, and warning threshold have been configured correctly.

**Action** From operation mode, enter the **show multicast forwarding-cache statistics** command.

```
user@host> show multicast forwarding-cache statistics
Instance: master Family: INET
Suppress Threshold                    150000
Reuse Value                            70000
Warning Threshold                         40
Currently Used Entries                    17
```

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol. See the Junos OS Routing Protocols Configuration Guide.
- 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. This example creates a flow map policy called **policyForFlow1**. The policy matches the source address using the **source-address-filter** statement and matches the group address using the **prefix-list-filter** statement.

  NOTE: The addresses must match the configured policy for flow mapping to occur.

- **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 this example, forwarding entries (10.11.11.11, g1) and (10.11.11.12, g1) match the flow map **flowMap1**. In this case, if a particular outbound interface is admitted for entry (10.11.11.11, g1), it is automatically admitted for entry (10.11.11.12, g1), even if there is no longer enough remaining bandwidth available after creating entry (10.11.11.11, g1). The interface is added because only one of the two sources 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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**

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.

   ```
   [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. Apply the flow map policy.

   ```
   [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. 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
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**

**Related Documentation**

- Examples: Configuring Administrative Scoping on page 231
- Examples: Configuring Bandwidth Management on page 265

## Example: Configuring Ingress PE Redundancy



### 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.

### 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. See the Junos® OS Network Interfaces.
- Configure a full mesh of P2P LSPs between the PEs in the backup group. See the Junos OS MPLS Applications Configuration Guide.

## 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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 1.1.1.1/32p2mp-lsp-next-hop p2mp-lsp
   user@host# set route 1.1.1.1/32backup-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

**Related Documentation**

- Examples: Configuring Administrative Scoping on page 231
- Examples: Configuring Bandwidth Management on page 265
- Examples: Configuring the Multicast Forwarding Cache on page 286

## Configuring PIM-to-IGMP and PIM-to-MLD Message Translation

- Understanding PIM-to-IGMP and PIM-to-MLD Message Translation on page 299
- Configuring PIM-to-IGMP Message Translation on page 301
- Configuring PIM-to-MLD Message Translation on page 302

### 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.

## 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
```

## 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
```

**Related Documentation**

- Configuring IGMP on page 305
- Examples: Configuring MLD on page 331

CHAPTER 6

# Internet Group Management Protocol



## Configuring IGMP



### Understanding Group Membership Protocols

There is a big difference between the multicast protocols used between host and router and between the multicast routers themselves. Hosts on a given subnetwork need to inform their router only whether or not they are interested in receiving packets from a certain multicast group. The source host needs to inform its routers 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 routers of their participation in a multicast group. Between adjacent routers, 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 router 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 router 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 router 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 routers:

- IGMPv1—The original protocol defined in RFC 1112. An explicit join message is sent to the router, but a timeout is used to determine when hosts leave a group. This process wastes processing cycles on the router, especially on older or smaller routers.
- IGMPv2—Defined in RFC 2236. Among other features, IGMPv2 adds an explicit leave message to the join message so that routers 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 router 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 router 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 router.

## Understanding IGMP

The Internet Group Management Protocol (IGMP) manages the membership of hosts and routers in multicast groups. IP hosts use IGMP to report their multicast group memberships to any immediately neighboring multicast routers. Multicast routers 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]).

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

For each attached network, a multicast router can be either a querier or a nonquerier. The querier router 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 router can specify particular routers from which it accepts or rejects traffic. With IGMPv3, a multicast router can learn which sources are of interest to neighboring routers.

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 routers, IGMPv3 routers 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.

## 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 563.

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.

## 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.

## 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.

## 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.

## 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.

To filter unwanted IGMP reports:

1. Configure an IGMPv2 policy.

   ```
   [edit policy-statement reject_policy_v2]
   user@host# set from route-filter 224.1.1.1/32 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 224.1.1.1/32 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.

## 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.

## 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.

## 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.

## 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.

## 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.

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

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.

## 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 225.1.1.1.

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 225.1.1.1
   ```

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 225.1.1.1 ;
     }
   }
   ```

3. After you have committed the configuration and the source is sending traffic, use the **show igmp group** command to verify that static group 225.1.1.1 has been created.

   ```
   user@host> show igmp group
   Interface: fe-0/1/2
        Group: 225.1.1.1
            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 225.1.1.1 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 225.1.1.1 {
      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 225.1.1.1, 225.1.1.2, and 225.1.1.3 have been created.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 225.1.1.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 225.1.1.2
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 225.1.1.3
         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 225.1.1.1 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 225.1.1.1 {
        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 225.1.1.1, 225.1.1.3, and 225.1.1.5 have been created.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 225.1.1.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 225.1.1.3
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 225.1.1.5
         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 225.1.1.1 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 225.1.1.1 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 225.1.1.1 {
      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 225.1.1.1 has been created and that source 10.0.0.2 has been accepted.

```
user@host> show igmp group
Interface: fe-0/1/2
     Group: 225.1.1.1
         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 255.1.1.1 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 225.1.1.1 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 225.1.1.1 {
      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 225.1.1.1 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: 225.1.1.1
         Source: 10.0.0.2
         Last reported by: Local
         Timeout: 0 Type: Static
     Group: 225.1.1.1
         Source: 10.0.0.3
```

```
        Last reported by: Local
        Timeout: 0 Type: Static
    Group: 225.1.1.1
        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 225.1.1.1 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 225.1.1.1 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 225.1.1.1 {
         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 225.1.1.1 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: 225.1.1.1
           Source: 10.0.0.2
           Last reported by: Local
           Timeout: 0 Type: Static
       Group: 225.1.1.1
           Source: 10.0.0.4
           Last reported by: Local
           Timeout: 0 Type: Static
       Group: 225.1.1.1
   ```

```
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 225.1.1.1.

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 225.1.1.1 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 225.1.1.1 {
         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 225.1.1.1 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: 225.1.1.1
            Group mode: Exclude
            Source: 10.0.0.2
            Last reported by: Local
            Timeout: 0 Type: Static
   ```

## 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 9 on page 322 describes the recordable IGMP events.

### Table 9: 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
```

## 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.

Beginning with Junos OS 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.

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.

## 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. |

| Flag | Description |
|---|---|
| **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. |
| **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 232.1.1.2
   ```

7. View the trace file.

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

### 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*]**

### 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.

**Related Documentation**

- Examples: Configuring MLD on page 331

## Example: Configuring SSM Maps for Different Groups to Different Sources



### 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.

### 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 you to 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, and then paste the commands into the CLI at the **[edit]** hierarchy level.

```
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 329
- Displaying the PIM Groups on page 329
- Displaying the Entries in the IP Multicast Forwarding Table on page 329

### *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 IGMP logical interface (fe-0/1/0.0), 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 mroute 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 254
- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 506

CHAPTER 7

# Multicast Listener Discovery



## Examples: Configuring MLD



### 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 router maintains a list of host multicast addresses that have listeners for each subnetwork, as well as a timer for each address. However, the router does not need to know the address of each listener—just the address of each host. The router 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.

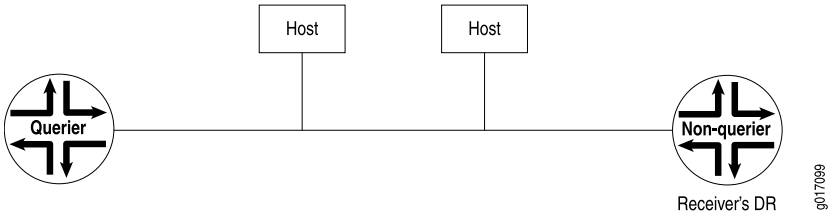
MLD is an integral part of IPv6 and must be enabled on all IPv6 routers and hosts that need to receive IP multicast traffic. The Junos OS supports MLD versions 1 and 2. Version 2 is supported for source-specific multicast (SSM) include and exclude modes.

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 router can be either a querier or a nonquerier. A querier router, usually one per subnet, solicits group membership information by transmitting MLD queries. When a host reports to the querier router that it has interested listeners, the querier router forwards the membership information to the rendezvous point (RP) router 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 router. The RPT is the initial path used by the sender to transmit information to the interested listeners. Nonquerier routers do not transmit MLD queries on a subnet but can do so if the querier router fails.

All MLD-configured routers start as querier routers on each attached subnet (see Figure 44 on page 332). The querier router on the right is the receiver's DR.

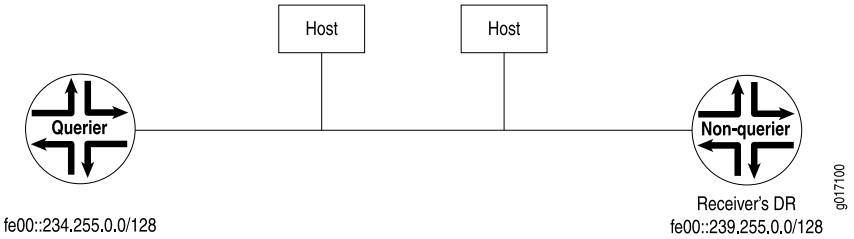
### Figure 44: Routers Start Up on a Subnet




To elect the querier router, the routers exchange query messages containing their IPv6 source addresses. If a router hears a query message whose IPv6 source address is numerically lower than its own selected address, it becomes a nonquerier. In Figure 45 on page 333, the router on the left has a source address numerically lower than the one on the right and therefore becomes the querier router.

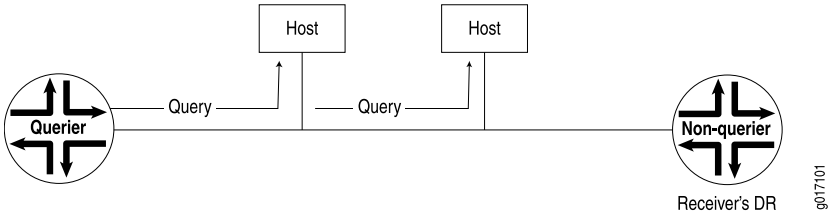
NOTE: In the practical application of MLD, several routers on a subnet are nonqueriers. If the elected querier router fails, query messages are exchanged among the remaining routers. The router with the lowest IPv6 source address becomes the new querier router. The IPv6 Neighbor Discovery Protocol (NDP) 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 45: Querier Router Is Determined



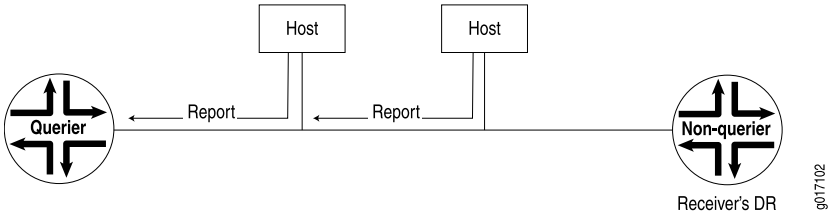

The querier router sends general MLD queries on the **link-scope all-nodes** multicast address FF02::1 at short intervals to all attached subnets to solicit group membership information (see Figure 46 on page 333). Within the query message is the *maximum response delay* value, specifying the maximum allowed delay for the host to respond with a report message.

Figure 46: General Query Message Is Issued



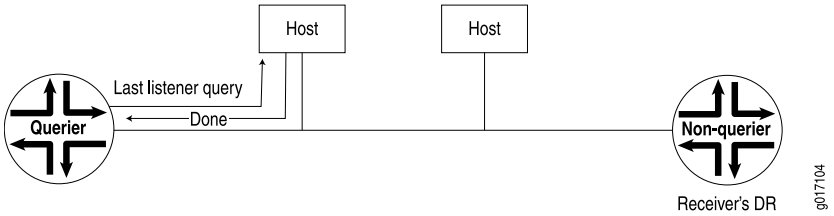

If interested listeners are attached to the host receiving the query, the host sends a report containing the host's IPv6 address to the router (see Figure 47 on page 333). If the reported address is not yet in the router's list of multicast addresses with interested listeners, the address is added to 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 47: Reports Are Received by the Querier Router



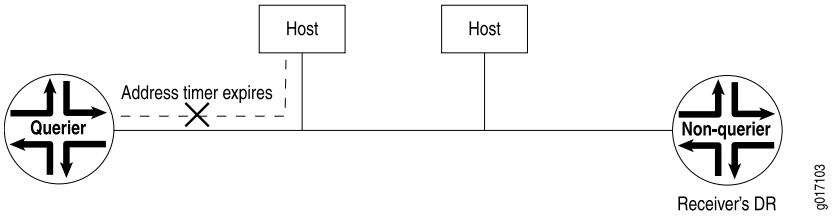

If the host has no interested multicast listeners, it sends a done message to the querier router. On receipt, the querier router issues a multicast address-specific query containing the last **listener query interval** value to the multicast address of the host. If the router does not receive a report 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 48 on page 334).

### Figure 48: Host Has No Interested Receivers and Sends a Done Message to Router




If a done message is not received by the querier router, the querier router continues to send multicast address-specific queries. If the timer set for the address on receipt of the last report expires, the querier router 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 49 on page 334).

### Figure 49: Host Address Timer Expires and Address Is Removed from Multicast Address List




## 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;
    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, 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Enable IPv6 unicast routing. See the Junos OS Routing Protocols Configuration Guide.
- Enable PIM. See "PIM Overview" on page 15.

### 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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.

## 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

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 ff02::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 ff02::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 ff02::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 ff02::1:ff05:1a8d has been created.

   ```
   user@host> show mld group
   Interface: fe-0/1/2
   Group: ff02::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.

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 ff02::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 ff02::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 ff02::1:ff05:1a8d, ff02::1:ff05:1a8e, and ff02::1:ff05:1a8f have been created.

   ```
   user@host> show mld group

   Interface: fe-0/1/2
        Group: ff02::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   Interface: fe-0/1/2
        Group: ff02::1:ff05:1a8e
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   Interface: fe-0/1/2
        Group: ff02::1:ff05:1a8f
            Source: fe80::2e0:81ff:fe05:1a8d
            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 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 ff02::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 ff02::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 ff02::1:ff05:1a8d, ff02::1:ff05:1a8f, and ff02::1:ff05:1a91 have been created.

   ```
   user@host> show mld group

   Interface: fe-0/1/2
        Group: ff02::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   Interface: fe-0/1/2
        Group: ff02::1:ff05:1a8f
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   Interface: fe-0/1/2
        Group: ff02::1:ff05:1a91
            Source: fe80::2e0:81ff:fe05:1a8d
            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 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 ff02::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 ff02::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 ff02::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 ff02::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: ff02::1:ff05:1a8d
            Source: fe80::2e0:81ff:fe05:1a8d
            Last reported by: Local
            Timeout: 0 Type: Static
   ```

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 ff02::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 ff02::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 ff02::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 ff02::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: ff02::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff02::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8e
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff02::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8f
         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 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 ff02::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 ff02::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 ff02::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 ff02::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: ff02::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8d
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff02::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a8f
         Last reported by: Local
         Timeout: 0 Type: Static
Interface: fe-0/1/2
     Group: ff02::1:ff05:1a8d
         Source: fe80::2e0:81ff:fe05:1a91
         Last reported by: Local
         Timeout: 0 Type: Static

Interface: fe-0/1/2
Group: ff02::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a8d
Last reported by: Local
Timeout: 0 Type: Static
Group: ff02::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a8f
Last reported by: Local
Timeout: 0 Type: Static
Group: ff02::1:ff05:1a8d
Group mode: Include
Source: fe80::2e0:81ff:fe05:1a91
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 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 ff02::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 ff02::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 ff02::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 ff02::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: ff02::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.

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Enable IPv6 unicast routing. See the Junos OS Routing Protocols Configuration Guide.
- Enable PIM. See "PIM Overview" on page 15.

### Overview

Table 10 on page 349 describes the recordable MLD join and leave events.

**Table 10: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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
```

## 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 Subscriber Management, Release 13.1.

Beginning with Junos OS 12.2, you can optionally configure a system log warning threshold for MLD muticast 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.

## Tracing MLD 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. |

| Flag | Description |
|---|---|
| **leave** | Trace leave group messages. |
| **mtrace** | Trace mtrace packets. Use the **mtrace** command to troubleshoot the software. |
| **normal** | Trace normal events. |
| **packets** | Trace all MLD packets. |
| **policy** | Trace policy processing. |
| **query** | Trace MLD 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. |
| **timer** | Trace timer processing. |

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

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 MLD trace file.

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

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

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

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

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

5. (Optional) Enable unrestricted file access.

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

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

   ```
   [edit protocols mld traceoptions]
   user@host# set flag all | match fe-1/0/1.0
   ```

7. View the trace file.

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

## 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]**

**Related Documentation**

- Configuring IGMP on page 305

CHAPTER 8

# Internet Group Management Protocol Snooping



## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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.

## 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]**

## 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 Junos® OS Network Interfaces.
- Configure an interior gateway protocol. See the Junos OS Routing Protocols Configuration Guide.
- 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. However, this interval must 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 50 on page 364 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 50: Networks Without IGMP Snooping Configured

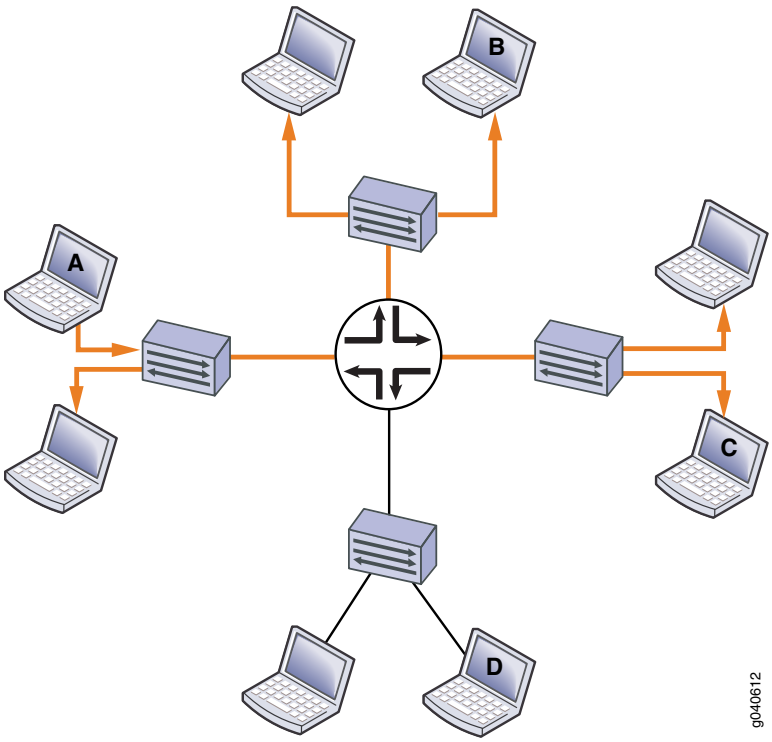
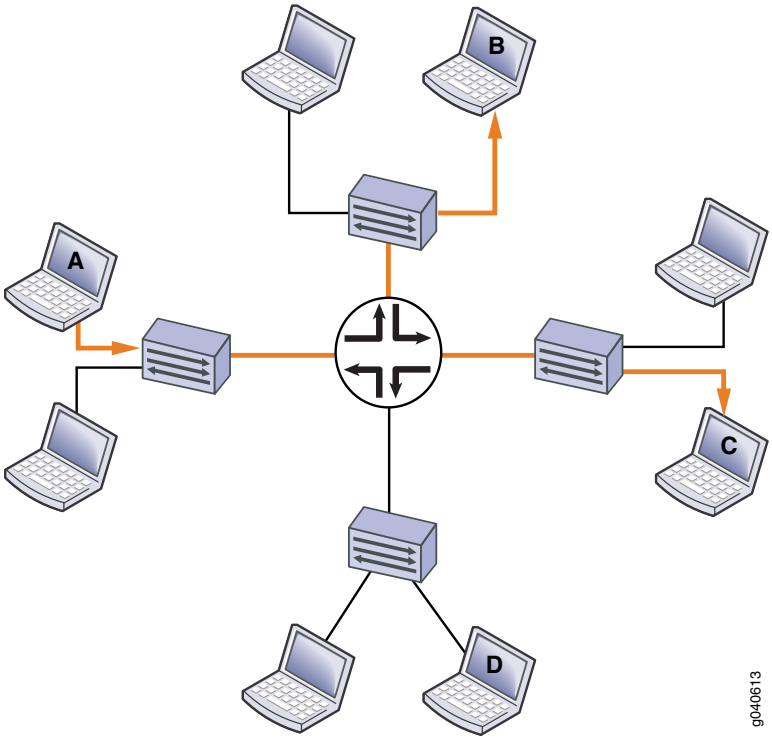



Figure 51 on page 365 shows the same networks with IGMP snooping configured. The Layer 2 devices forward multicast traffic to registered receivers only.

Figure 51: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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

## 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
```

**Related Documentation**

- Understanding Multicast Snooping on page 355

CHAPTER 9

# Multicast Snooping



## Example: Configuring Multicast 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.

## 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.

NOTE: **You can configure multicast snooping to ignore messages about spanning tree topology changes for the virtual-switch routing-instance type only.**

## 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 bridge-domains *bridge-domain-name*]**
- **[edit routing-instances *routing-instance-name*]**
- **[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*]**
- **[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]**

By default, multicast snooping is disabled. You can enable multicast snooping in VPLS or virtual switch instance types in the instance hierarchy or in one or more bridge domains.

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. Multicast snooping options configured at the bridge domain level only apply to that particular bridge domain. The options configured at the bridge domain take precedence over the options configured at the instance level.

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**.

## Example: Configuring Multicast Snooping

This example shows how to configure multicast snooping in a bridge or VPLS routing-instance scenario.



### 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 Junos® OS Network Interfaces.
- Configure an interior gateway protocol. See the Junos OS Routing Protocols Configuration Guide.
- 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.

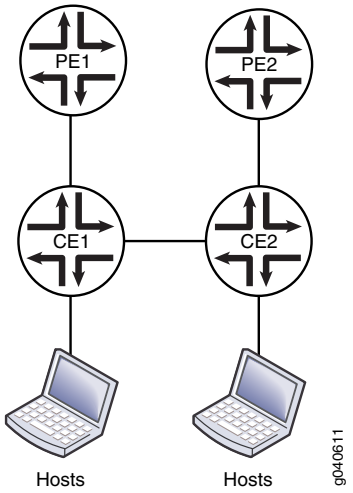
  The PE router interface does not wait to receive membership reports in response to the next general or group-specific query. Instead, the IGMP snooping process sends a general query message toward the CE router. The hosts connected to the CE router reply with reports for all groups they are interested 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 52 on page 376 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 52: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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;
      }
    }
    graceful-restart {
      restart-duration 120;
    }
  }
}
```

```
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 multicast snooping statistics
- show route table

## 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.

## 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 in a bridge domain named bridge1:

1. Enable state replication for MC-LAG interfaces.

   ```
   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

   Instance: bridge-domain bridge1
   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.

## 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 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:

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.

2. Verify your configuration using the **show multicast-snooping-options** operational mode command.

   ```
   [edit]
   user@host> show multicast-snooping-options

   graceful-restart {
       restart-duration 200;
   }
   ```

3. Enter commit from the configuration mode.

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

**Related Documentation**

- Example: Configuring Multicast Snooping on page 373

CHAPTER 10

# Multiprotocol BGP Multicast VPN



## Examples: Configuring Multiprotocol BGP Multicast VPNs



### 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.

## 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.



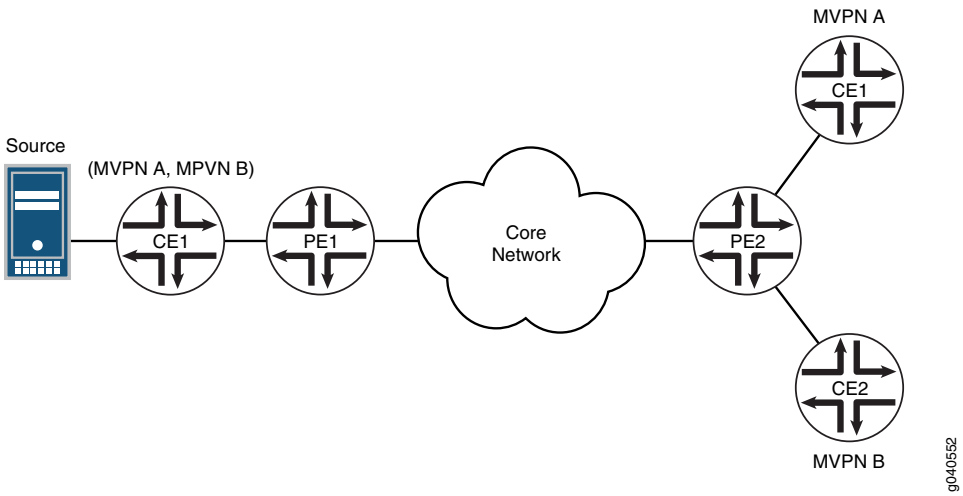
### Requirements

Before you begin:

- Configure the router interfaces. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure a BGP-MVPN control plane. See "MBGP-Based Multicast VPN Trees" on page 26 in the Multicast Protocols Configuration Guide.

- Configure LDP as the signaling protocol on all P2MP provider and provider-edge routers. See LDP Operation in the Junos OS MPLS Applications Configuration Guide.
- 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 Configuration Guide.
- 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 and Junos Services Interfaces Configuration Guide in the in the Junos Services Interfaces Configuration Release 12.3.
- 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 and Junos Services Interfaces Configuration Guide in the Junos Services Interfaces Configuration Release 12.3.
- 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 53 on page 385, the CE1 and CE2 routers belong to different MVPNs. However, they want to receive a multicast stream being sent by S. 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.

### Figure 53: Extranet Configuration of MBGP MVPN with P2MP LDP LSPs as Data Plane



## Overview

This example shows how to configure P2MP LSP LSPs 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.

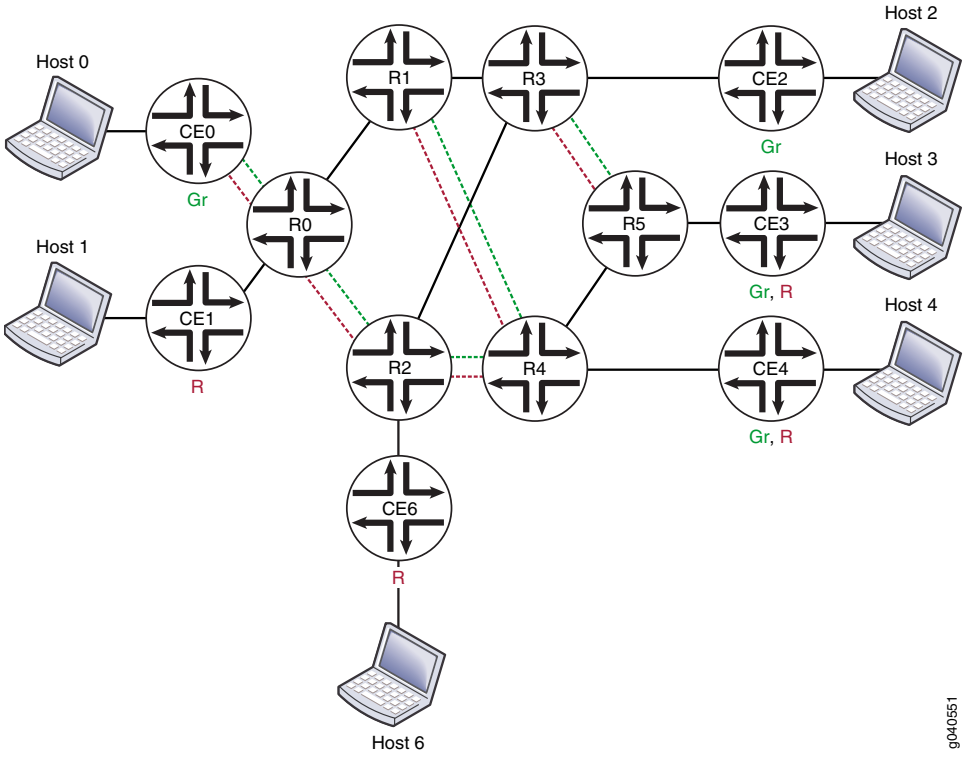
When you configure selective provider tunnels, leaves discover the P2MP LSP root as follows. A PE router with a receiver for a customer multicast stream behind it needs to discover the identity of the PE router (and the provider tunnel information) with the source of the customer multicast stream behind 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 54 on page 387 shows the topology used in this example.

Figure 54: P2MP LDP LSPs as the Data Plane for Intra-AS MBGP MVPNs




In Figure 54 on page 387, 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.

## 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 running Junos OS Release 10.4 or later. 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.

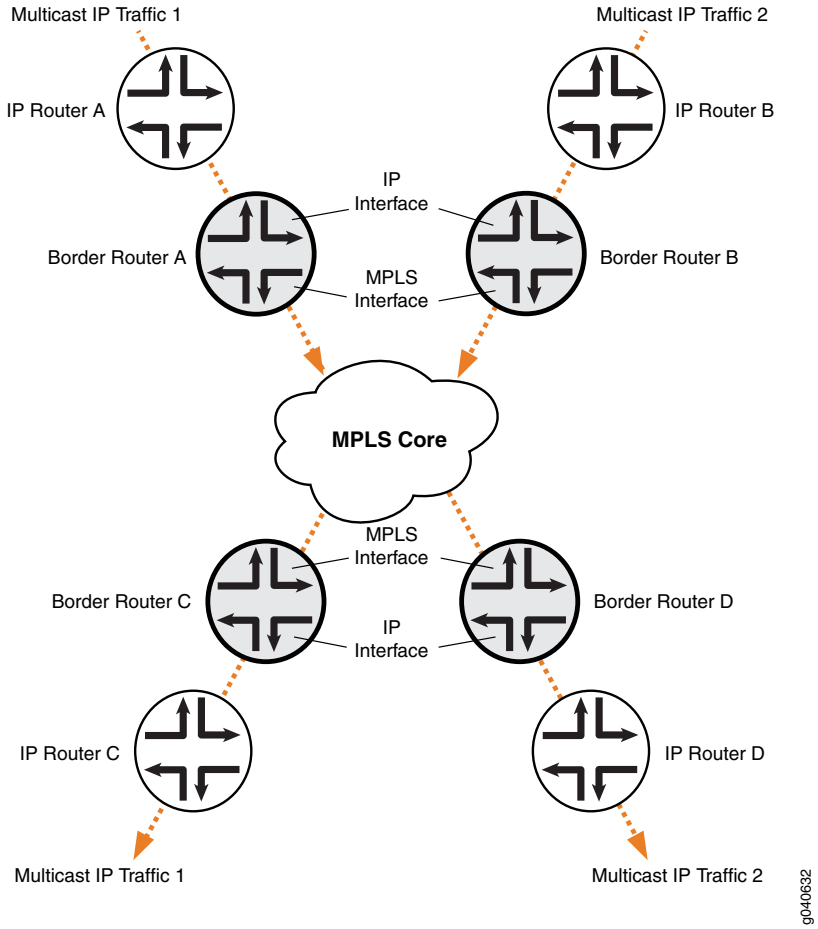
For each mpls-internet-multicast routing instance, the **ingress-replication** statement is required under the **provider-tunnel** statement and also under the **[edit routing-instances routing-instance-name provider-tunnel selective group source]** hierarchy level.

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—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 55 on page 391. 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 55: 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.

```
set routing-instances IM-A instance-type mpls-internet-multicast
set routing-instances IM-A provider-tunnel ingress-replication create-new-ucast-tunnel
set routing-instances IM-A provider-tunnel ingress-replication label-switched-path
  label-switched-path-template default-template
set routing-instances IM-A provider-tunnel selective group group-address source
  source-address ingress-replication label-switched-path
set routing-instances IM-A protocols mvpn
set protocols pim mpls-internet-multicast
```

**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 IP multicast instance **IM-A** 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.

1. Configure the routing instance type for IM-A to be **mpls-internet-multicast**.

```
[edit routing-instances]
user@host# set IM-A instance-type mpls-internet-multicast
```

2. 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]
user@host# set IM-A provider-tunnel ingress-replication create-new-ucast-tunnel
```

NOTE: Alternatively, use the **label-switched-path-template** statement to configure a point-point LSP for the ingress tunnel.

3. Configure the point-to-point LSP to use the default template settings (this is needed only when using RSVP tunnels).

```
[edit routing-instances]
user@host# set IM-A provider-tunnel ingress-replication label-switched-path
label-switched-path-template default-template
```

4. Configure selective ingress replication provider tunnels.

```
[edit routing-instances]
user@host# set IM-A provider-tunnel selective group 232.1.1.1/32 source 192.168.195.145/32
ingress-replication label-switched-path
```

5. Configure the MVPN Protocol in the routing instance.

```
[edit routing-instances]
user@host# set IM-A protocols mvpn
user@host# up
```

6. Add the **mpls-internet-multicast** configuration statement under the **[edit protocols pim]** hierarchy level in the master instance.

```
[edit protocols]
user@host# set pim mpls-internet-multicast
```

7. Commit the configuration.

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

8. Use the **show ingress-replication mvpn** command to check the ingress replication status.

```
[edit]
user@host# run 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
```

9. Use the **show mvpn instance** command to show the ingress replication tunnel type.

```
[edit]
user@host# run show mvpn instance IM-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 : IM-A
   MVPN Mode : SPT-ONLY
   Provider tunnel: S-P-tnl:INGRESS-REPLICATION:MPLS Label 18:10.255.245.6
   Neighbor               S-P-tnl
   10.255.245.2           INGRESS-REPLICATION:MPLS Label 22:10.255.245.2
   10.255.245.7           INGRESS-REPLICATION:MPLS Label 19:10.255.245.7
```

## 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.



### 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 systems 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

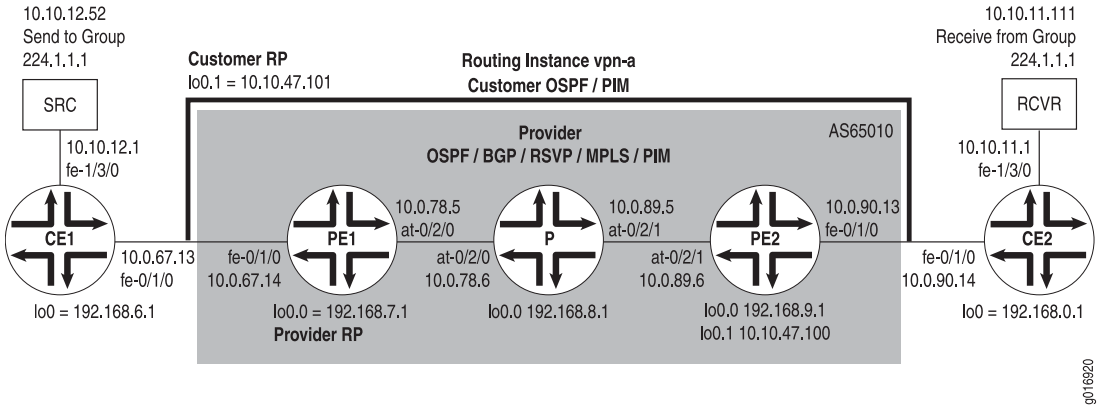
## Overview and Topology

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 56 on page 394.

### Figure 56: 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 56 on page 394, 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 and P routers, enable the provider instance of PIM. Add the core-facing ATM interfaces. On the PE routers, also configure the **lo0.0** interface. Specify the mode as **sparse** and the version as **2**.

```
user@PE1# set protocols pim interface at-0/2/0.0 mode sparse
user@PE1# set protocols pim interface at-0/2/0.0 version 2
user@PE1# set protocols pim interface lo0.0 mode sparse
user@PE1# set protocols pim interface lo0.0 version 2

user@P# set protocols pim interface at-0/2/0.0 mode sparse
user@P# set protocols pim interface at-0/2/0.0 version 2
user@P# set protocols pim interface at-0/2/1.0 mode sparse
user@P# set protocols pim interface at-0/2/1.0 version 2

user@PE2# set protocols pim interface at-0/2/1.0 mode sparse
user@PE2# set protocols pim interface at-0/2/1.0 version 2
user@PE2# set protocols pim interface lo0.0 mode sparse
user@PE2# set protocols pim interface lo0.0 version 2
```

Use the **show pim interfaces** command to verify that PIM sparse-mode is enabled on the core-facing ATM interfaces.

2. On the PE routers, enable the VPN customer instance of 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.

3. On the CE routers, enable the customer instance of 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. The **pim-asm** statement instructs the router to accept the multicast stream from any source.

   ```
   user@PE1# set routing-instances vpn-a provider-tunnel pim-asm group-address
     224.1.1.1
   ```

   Use the **show configuration routing-instance vpn-a** command to verify that the multicast group address is correct on Router PE1.

2. On Router PE2, configure the provider tunnel. Specify the multicast address to be used.

   ```
   user@PE2# set routing-instances vpn-a provider-tunnel pim-asm group-address
     224.1.1.1
   ```

   Use the **show configuration routing-instance vpn-a** command to verify that the multicast group address is correct on Router PE2.

*Configuring the Rendezvous Point*

**Step-by-Step Procedure**

1. Configure Router PE1 to be the rendezvous point for the provider instance of PIM. Specify the **lo0.0** address of Router PE1.

   ```
   user@PE1# set protocols pim rp local address 192.168.7.1
   ```

   Use the **show pim rps** command to verify that to correct local IP address is configured for the provider instance RP.

2. Configure the static rendezvous point on Router P and the PE2 router for the provider instance of PIM. Specify the **lo0.0** address of Router PE1. Specify the version as **2**.

   ```
   user@P# set protocols pim rp static address 192.168.7.1 version 2

   user@PE2# set protocols pim rp static address 192.168.7.1 version 2
   ```

   Use the **show pim rps** command to verify that the correct static IP address is configured for the provider instance RP.

3. Configure Router PE1 to be the rendezvous point for the customer instance of PIM. 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 customer instance RP.

4. On Router PE2, configure the static rendezvous point for the customer instance of PIM. 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 customer instance RP.

5. On the CE routers, configure the static rendezvous point for the customer instance of PIM. 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 customer instance RP.

6. Use the **commit check** command to verify that the configuration can be successfully committed. If the configuration passes the check, commit the configuration.
7. Start the multicast sender device connected to CE1.
8. Start the multicast receiver device connected to CE2.
9. Verify that the receiver is receiving the multicast stream.
10. 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;
    }
  }
  pim {
    rp {
      local {
        address 192.168.7.1;
      }
    }
```

```
      interface at-0/2/0.0 {
        mode sparse;
        version 2;
      }
      interface lo0.0 {
        mode sparse;
        version 2;
      }
    }
  }
  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 {
        pim-asm {
          group-address 224.1.1.1;
        }
      }
      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;
        }
      }
    }
    pim {
      rp {
        static {
          address 192.168.7.1 {
            version 2;
          }
        }
      }
      interface at-0/2/0.0 {
        mode sparse;
        version 2;
      }
      interface at-0/2/1.0 {
        mode sparse;
        version 2;
      }
    }
  }
```

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;
      }
    }
    pim {
      rp {
        static {
          address 192.168.7.1 {
            version 2;
          }
        }
      }
      interface lo0.0 {
        mode sparse;
        version 2;
      }
      interface at-0/2/1.0 {
        mode sparse;
        version 2;
      }
    }
}
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 {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
```

```
    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;
      }
    }
  }
```

## 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.
- Configure the BGP-to-OSPF routing policy. See the Routing Policy Configuration 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.

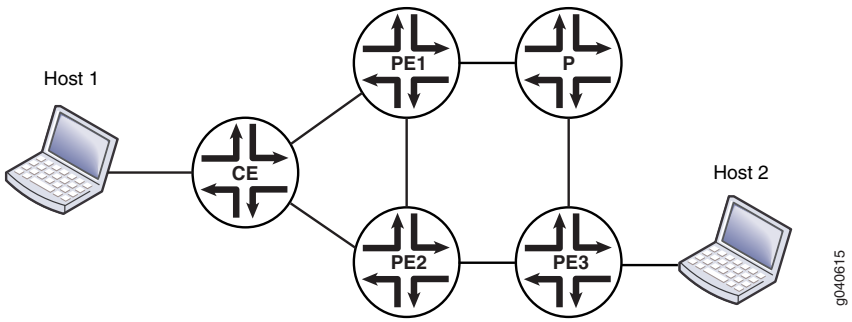
- **provider-tunnel pim-ssm group-address**—Specifies a valid SSM VPN group address. The SSM VPN group address and the source address are advertised by the type-1 autodiscovery route. On receiving an autodiscovery route with the SSM VPN group address and the source address, a PE router sends an (S,G) 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 57 on page 413 shows the topology used in this example.

Figure 57: 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 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 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 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
```

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 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
```

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 {
      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 {
      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**

## 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.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure the point-to-multipoint static LSP. See Configuring Point-to-Multipoint LSPs for an MBGP MVPN.

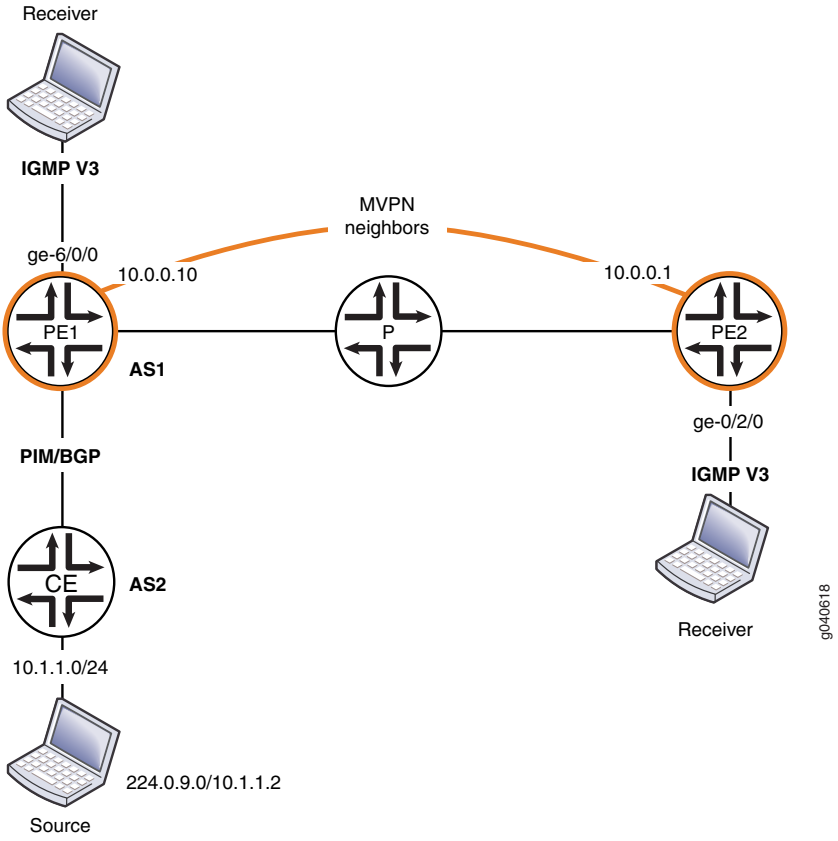
### Overview

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 58 on page 422 shows the topology used in this example.

Figure 58: 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**

## 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.

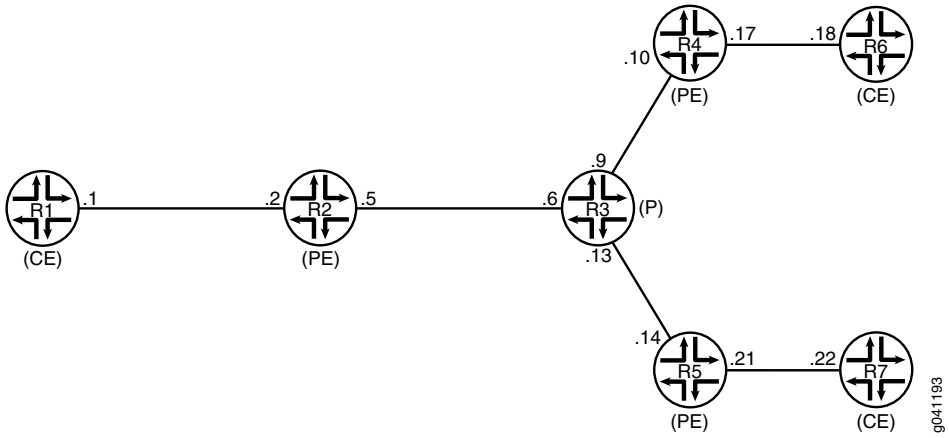
### Overview

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 59 on page 426 shows the topology used in this example.

Figure 59: 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 426 section. The "Configuring Device R4" on page 429 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 1.1.1.1/32
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 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.1
```

**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 1.1.1.2/32
set interfaces lo0 unit 102 family inet address 100.1.1.2/32
set protocols mpls interface ge-1/2/1.5
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-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 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 100.1.1.2
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 1.1.1.2
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 1.1.1.3/32
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 1.1.1.3
```

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 1.1.1.4/32
set interfaces lo0 unit 104 family inet address 100.1.1.4/32
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 1.1.1.4
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 1.1.1.2 import dampPolicy
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 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 100.1.1.2
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 1.1.1.4
set routing-options autonomous-system 1001
```

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 1.1.1.5/32
set interfaces lo0 unit 105 family inet address 100.1.1.5/32
set protocols mpls interface ge-1/2/0.14
set protocols bgp group ibgp type internal
set protocols bgp group ibgp local-address 1.1.1.5
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.4
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 100.1.1.2
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 1.1.1.5
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 1.1.1.6/32
set protocols sap listen 224.1.1.1
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 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.6
```

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 1.1.1.7/32
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 100.1.1.2
set protocols pim interface all
set routing-options router-id 1.1.1.7
```

*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 1.1.1.4/32
user@R4# set lo0 unit 104 family inet address 100.1.1.4/32
```

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 1.1.1.4
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 1.1.1.2 import dampPolicy
user@R4# set neighbor 1.1.1.5
```

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 100.1.1.2
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-instances vpn-A]
user@R4# set routing-options router-id 1.1.1.4
user@R4# set routing-options 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 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 1.1.1.4/32;
    }
  }
  unit 104 {
    family inet {
      address 100.1.1.4/32;
    }
  }
}
```

```
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 protocols
rsvp {
  interface all {
    aggregate;
  }
}
mpls {
  interface all;
  interface ge-1/2/0.10;
}
bgp {
  group ibgp {
    type internal;
    local-address 1.1.1.4;
    family inet-vpn {
      unicast;
      any;
    }
    family inet-mvpn {
      signaling {
        damping;
```

```
        }
      }
      neighbor 1.1.1.2 {
        import dampPolicy;
      }
      neighbor 1.1.1.5;
    }
  }
  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 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 100.1.1.2;
        }
      }
      interface ge-1/2/1.17 {
        mode sparse;
      }
    }
    mvpn;
  }
}
```

```
user@R4# show routing-optons
router-id 1.1.1.4;
```

```
autonomous-system 1001;
```

## Verification

Confirm that the configuration is working properly.


- Verifying That Route Flap Damping Is Disabled on page 434
- Verifying Route Flap Damping on page 434


### *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...
1.1.1.2                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
1.1.1.5                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** When **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 1.1.1.2.

**Related Documentation**

- Example: Configuring MBGP MVPN Extranets on page 442
- Multiprotocol BGP MVPNs Overview on page 24

## Configuring MBGP MVPN Wildcards

- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 435
- Configuring a Selective Provider Tunnel Using Wildcards on page 440
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 440

### 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 435
- Scenarios for Using Wildcard S-PMSI on page 436
- Types of Wildcard S-PMSI on page 437
- Differences Between Wildcard S-PMSI and (S,G) S-PMSI on page 437
- Wildcard (*,*) S-PMSI and PIM Dense Mode on page 438
- Wildcard (*,*) S-PMSI and PIM-BSR on page 438
- Wildcard Source and the 0.0.0.0/0 Source Prefix on page 439

#### About S-PMSI

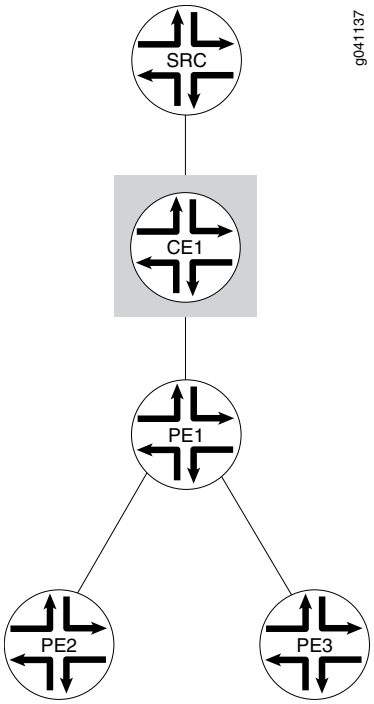
The provider multicast service interface (PMSI) is a BGP tunnel attribute that contains the tunnel ID used by the PE router for transmitting traffic through the core of the provider network. A selective PMSI (S-PMSI) autodiscovery route advertises binding of a given MVPN customer multicast flow to a 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 60 on page 436 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 60: 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 224.1.1.0/24 {
          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, 224.1.1.0/24) is bound to **sptnl3**. A join message for (*, 224.1.1.0/24) is bound to **sptnl2**.

## 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 224.0.0/24 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 (*,224.0.0/24) join messages to a selective provider tunnel.

   ```
   [edit routing-instances vpna provider-tunnel selective group 224.0.0/24
     wildcard-source]
   user@router# set rsvp-te (Routing Instances Provider Tunnel Selective)
     label-switched-path-template provider-tunnel3
   ```

## 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 224.1.1.0/24 {
          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, 224.1.1.1) join message is bound to the sptnl3 S-PMSI autodiscovery route.
- A customer multicast (10.2.1.1, 224.1.1.1) join message is bound to the sptnl2 S-PMSI autodiscovery route.
- A customer multicast (10.1.1.1, 224.2.1.1) 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.

**Related Documentation**

- Example: Configuring MBGP MVPN Extranets on page 442
- Examples: Configuring Multiprotocol BGP Multicast VPNs on page 383
- Multiprotocol BGP MVPNs Overview on page 24

## Example: Configuring MBGP MVPN Extranets



### 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.

## 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.

## 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:

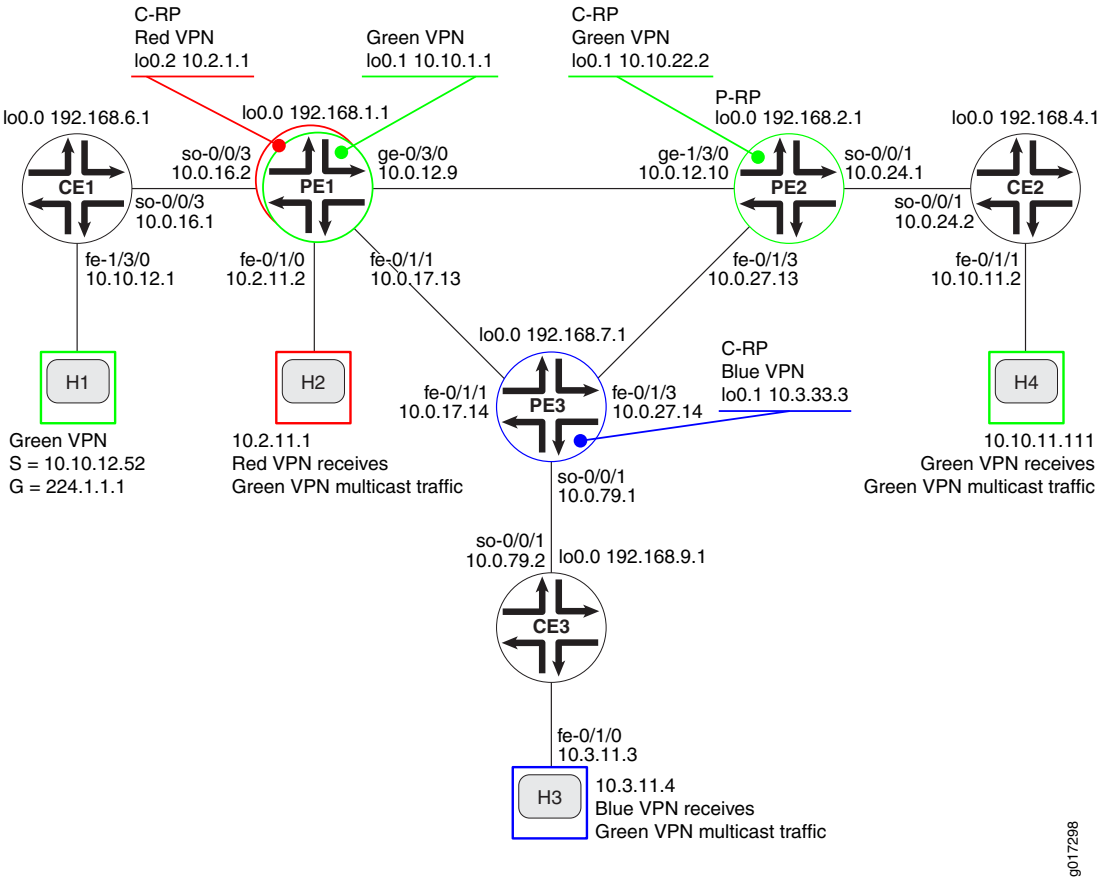
- Junos OS Release 9.5 or later
- Six M Series, T Series, TX Series, or MX Series Juniper routers
- One adaptive services PIC or MultiServices PIC in each of the M Series or T Series routers acting as PE routers
- One host system capable of sending multicast traffic and supporting the Internet Group Management Protocol (IGMP)
- Three host systems capable of receiving multicast traffic and supporting IGMP

### Overview and Topology

In the network topology shown in Figure 61 on page 445:

- 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 61: 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;
      }
    }
  }
```

**Related Documentation**

- Examples: Configuring Multiprotocol BGP Multicast VPNs on page 383
- Multiprotocol BGP MVPNs Overview on page 24

## Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs

- Understanding Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 481
- Example: Configuring Redundant Virtual Tunnel Interfaces in MBGP MVPNs on page 482

### 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.

## 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

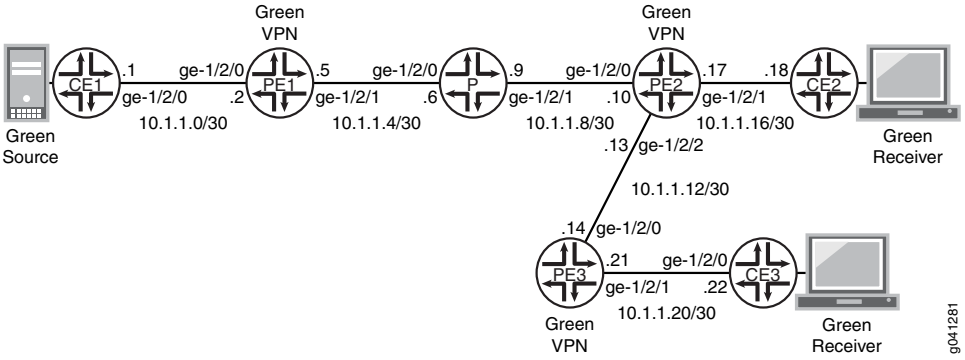
The routing device that has redundant VT interfaces configured must be running Junos OS Release 12.3 or later.

### 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 62 on page 483 shows the topology used in this example.

Figure 62: Multiple VT Interfaces in MBGP MVPN Topology




"CLI Quick Configuration" on page 483 shows the configuration for the customer edge (CE), provider (P), and provider edge (PE) devices in Figure 62 on page 483. The section "Step-by-Step Procedure" on page 486 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 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-1/2/0.0
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-1/2/0 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 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-1/2/0.0
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-1/2/0 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 1.1.1.7/32
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ge-1/2/0.0
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-1/2/0 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 1.1.1.3/32
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 1.1.1.3
```

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 1.1.1.2/32
set interfaces lo0 unit 1 family inet address 100.1.1.2/32
set protocols mpls interface ge-1/2/1.0
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-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 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 100.1.1.2
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 1.1.1.2
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 1.1.1.4/32
set interfaces lo0 unit 1 family inet address 100.1.1.4/32
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 1.1.1.4
set protocols bgp group ibgp family inet-vpn any
set protocols bgp group ibgp family inet-mvpn signaling
set protocols bgp group ibgp neighbor 1.1.1.2
set protocols bgp group ibgp neighbor 1.1.1.5
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 100.1.1.2
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 1.1.1.4
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 1.1.1.5/32
set interfaces lo0 unit 1 family inet address 100.1.1.5/32
set protocols mpls interface ge-1/2/0.0
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-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 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 100.1.1.2
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 1.1.1.5
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 1.1.1.4/32
user@PE2# set lo0 unit 1 family inet address 100.1.1.4/32
```

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 1.1.1.4
```

```
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
```

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 100.1.1.2
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 1.1.1.4
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 1.1.1.4/32;
        }
    }
    unit 1 {
        family inet {
            address 100.1.1.4/32;
        }
    }
}
```

user@PE2# **show protocols**

```
mpls {
    interface ge-1/2/0.0;
    interface ge-1/2/2.0;
}
bgp {
    group ibgp {
        type internal;
        local-address 1.1.1.4;
```

```
    family inet-vpn {
      any;
    }
    family inet-mvpn {
      signaling;
    }
    neighbor 1.1.1.2;
    neighbor 1.1.1.5;
  }
}
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 100.1.1.2;
            }
          }
          interface ge-1/2/1.0 {
            mode sparse;
          }
        }
        mvpn;
      }
    }
```

```
user@PE2# show routing-options
router-id 1.1.1.4;
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**

- Multiprotocol BGP MVPNs Overview on page 24

CHAPTER 11

# Draft-Rosen Multicast VPN



## 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.

## 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.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure the VPN. See the Junos OS VPNs Configuration Guide.
- Configure the VPN import and VPN export policies. See Configuring Policies for the VRF Table on PE Routers in VPNs in the Junos OS VPNs Configuration Guide.
- 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 54 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 503.

  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

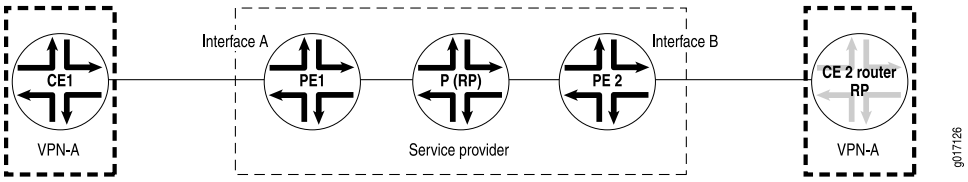
In multicast Layer 3 VPNs, the multicast PE routers must use the primary loopback address (or router ID) for sessions with their internal BGP peers. If the PE routers use a route reflector and the next hop is configured as **self**, Layer 3 multicast over VPN will not work, because PIM cannot transmit 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 63 on page 495 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 63: Multicast Connectivity on the CE Routers



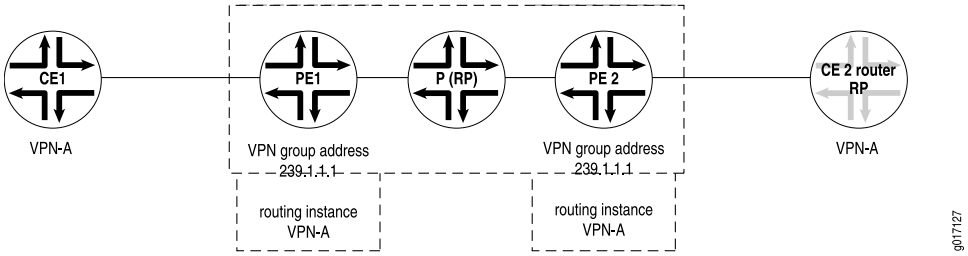

- **protocols pim version 2**—Enables PIM version 2 on the **lo0** interface of all PE routers and CE routers. You can either configure that specific interface or configure all interfaces with the **interface all** statement.
- **group-address**—In a routing instance, configure multicast connectivity 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 64 on page 496, the shared VPN group address configuration is 239.1.1.1.

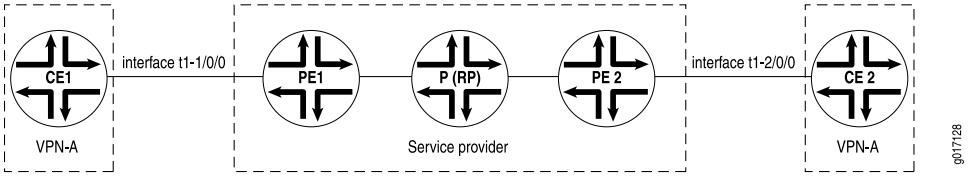
Figure 64: Multicast Connectivity for the VPN




- **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 65 on page 496.

Figure 65: 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 vpn-group-address 239.1.1.1
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-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 vpn-group-address 239.1.1.1
[edit routing-instances VPN-A]
user@host# set protocols provider-tunnel pim-asm group-address 239.1.1.1
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 vpn-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 t1-2/0/0:0.0 mode sparse
```

```
[edit routing-instances VPN-A]
user@host# set pim interface t1-2/0/0:0.0 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
[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 {
        vpn-group-address 239.1.1.1;
        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-1/1/0.49154          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
```

## 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 49,152 through 65,535. 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 494.

   ```
   [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
   ```

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-1/1/0.49152  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-1/1/0.49152  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.

**Related Documentation**

- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 506
- Example: Configuring Draft-Rosen MVPN Interoperability on page 516

## Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs

- Understanding Source-Specific Multicast VPNs on page 507
- Draft-Rosen 7 Multicast VPN Control Plane on page 507
- Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 507

## 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.

## 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 54 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 503.

## 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:

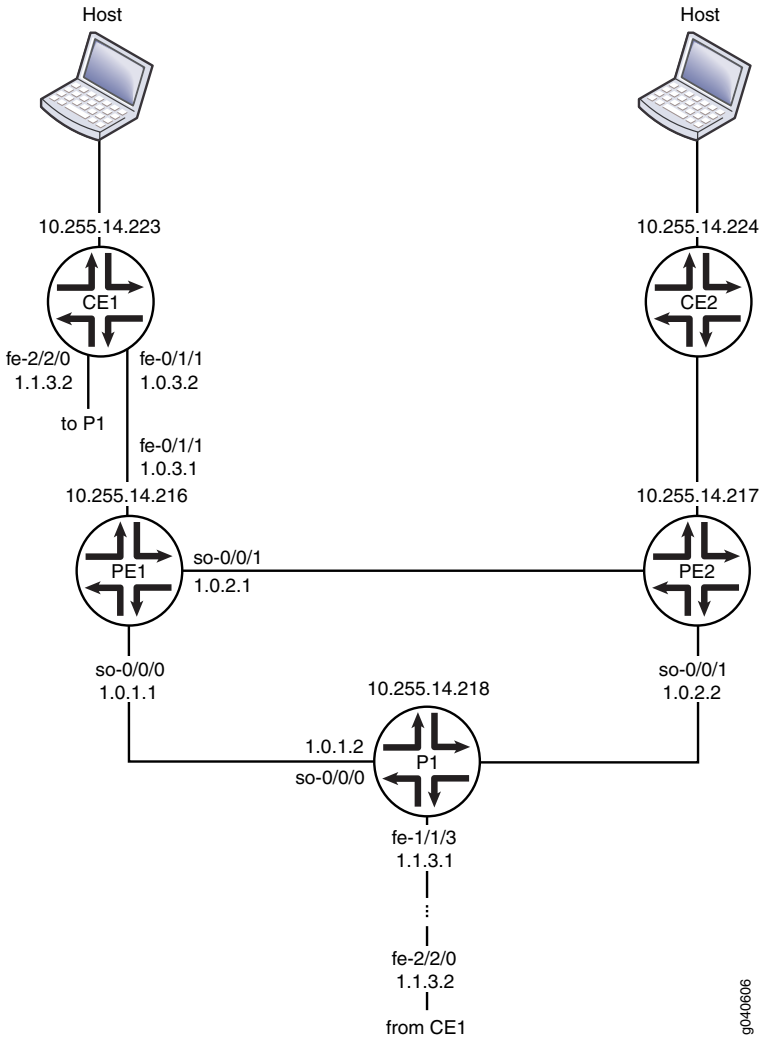
- **protocols bgp group *group-name* family inet-mdt signaling**—Enables MDT-SAFI signaling in BGP.
- **routing-instance *instance-name* protocols mvpn autodiscovery-only intra-as inclusive**—Enables the multicast VPN to use the MDT-SAFI autodiscovery NLRI.
- **routing-instance *instance-name* protocols 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 autodiscovery inet-mdt**—Enables PIM to learn about neighbors from the MDT-SAFI autodiscovery NLRI.

- **routing-instance *instance-name* provider-tunnel 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 66 on page 510 shows the topology for this example.

Figure 66: 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 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 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 pim-ssm group-address 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 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]
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.

**Related Documentation**

- Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs on page 493
- Example: Configuring Draft-Rosen MVPN Interoperability on page 516

# Example: Configuring Draft-Rosen MVPN Interoperability



## Configuring Draft-Rosen Multicast VPNs

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.

There are three functional areas to configure for draft-rosen MVPNs with service provider tunnels operating in SSM mode:

- Provider tunnel
- Autodiscovery mechanism
- Control plane protocol

  In the unicast environment for Layer 3 VPNs, all VPN state information is contained within the PE routers.

  However, with multicast for Layer 3 VPNs, PIM adjacencies are established in one of the following ways:

  - You can set PIM adjacencies between the customer edge (CE) router and the PE router through a VPN routing and forwarding (VRF) instance at the **[edit routing-instances *instance-name* protocols pim]** hierarchy level. You must include the **vpn-group-address** statement at this hierarchy level, specifying a multicast group. The RP listed in the VRF-instance is the VPN customer RP (C-RP).
  - You can also set the master PIM instance and the PE's IGP neighbors by configuring statements at the **[edit protocols pim]** hierarchy level. You must add the multicast group specified in the VRF instance to the master PIM instance. The set of master PIM adjacencies throughout the service provider network makes up the forwarding path that becomes an RP tree rooted at the service provider RP (SP-RP). Therefore, provider (P) routers within the provider core must maintain multicast state information for the VPNs.

For this configuration to work properly, you need two types of RP routers for each VPN:

- A VPN C-RP—An RP router located somewhere within the customer VPN
- An SP-RP—An RP router located within the service provider network

NOTE: A PE router can act as an SP-RP or the VPN C-RP of a VPN. However, when auto-RP and BSR are used, the PE cannot be a C-RP. It can, however, learn another router as C-RP by means of the auto-RP or BSR protocols.

## Understanding MVPN Interoperation with Other Vendors

This section presents notes on interoperating with other vendors in a draft-rosen multicast VPN.

If your Juniper Networks routers must interoperate with other vendors' routers, take one of these approaches:

- Configure the other vendors' routers to interoperate with the Juniper Networks routers.
- Configure the Juniper Networks routers to interoperate with the other vendors' routers.

If you are configuring the other vendors' routers to operate like Juniper Networks routers, verify the following:

- All provider tunnels use the same group address.
- On all routers, the **lo0.*n*** interface IP address in the routing instance matches the IP address on the **lo0.0** interface in the master instance.

By default the Junos OS attaches a route target to multicast distribution tree (MDT) subsequent address family identifier (SAFI) network layer reachability information (NLRI) route advertisements. Some vendors do not support attaching route targets to the MDT-SAFI route advertisements.

For interoperability with these vendors, the Junos OS allows importing of MDT-SAFI route advertisements without a route target being attached. The MDT-SAFI is imported if the MDT default address in the MDT-SAFI prefix matches the MDT default address configured within the routing instance.

## Example: Configuring Draft Rosen Interoperability and a VPN Tunnel Source

This example shows how to change the behavior of draft-rosen in the Junos OS for interoperability with certain other vendors' routing platforms.



### Requirements

Before you begin:

- Configure the router interfaces. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure the VPN. See the Junos OS VPNs Configuration Guide.
- 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 54 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 503.

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

By default, the local loopback address configured in a VPN routing and forwarding (VRF) routing instance is used as the source address when PIM hello messages, join messages, and prune messages are sent over multicast tunnel interfaces.

In the Junos OS default implementation of draft-rosen, **mt** interfaces are created dynamically in each VRF. PIM hello messages are sent over the **mt** interfaces to discover neighbors in the same VPN. The Junos OS requires that the **lo0.mvpn** address be configured in each routing instance. The local **lo0.mvpn** address is used as the source address when building PIM hello, join, and prune messages over the **mt** interface. The **lo0.mvpn** address is independent in routing instances and can be different from the **lo0.main** address in the master instance. All the source and destination address lookups and RPF checks are done in the routing instance.

For compatibility with certain other vendors' routers, the address used in the VRF routing instance for multicast tunnel interfaces must be the same as the primary loopback address configured in the master routing instance. In this example, **mt** interfaces use **lo0.0** as the source address. Each VRF routing instance uses the **lo0.0** address as the source address when sending PIM control packets over the **mt** interface.

Before configuring the loopback address used for PIM control messages to be the primary loopback address configured in the default routing instance, ensure that:

- The loopback address specified is configured in the master routing instance.
- The **inet** address family is enabled on the interface.

This example also configures a specific VPN tunnel source address in one routing instance. That address, instead of the **lo0.main** address, is used as the source address for the **mt** interface in that routing instance. The tunnel source is a static local address that is routable in the master instance. The routing instance uses the VPN tunnel source to form PIM neighbors over the **mt** interface. This enables different routing instances to take different paths even in the provider domain.

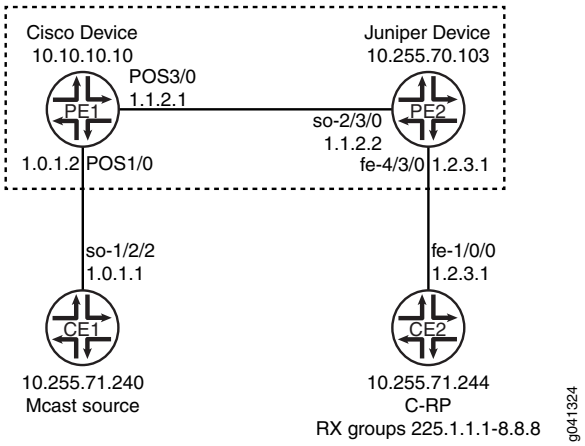
NOTE: For information about upgrading Juniper Networks routers from a software release that does not support the **default-vpn-source** configuration statement to a release that does, see the *Juniper Networks Junos 10.1 Software Release Notes*.

This example includes the following settings:

- **default-vpn-source**—In the master instance, changes the draft-rosen behavior so that the primary loopback address configured in the master routing instance is the multicast tunnel interface address in all VRF routing instances. Because the configuration includes **default-vpn-source**, you do not need to configure loopback addresses in the routing instances.
- **tunnel-source**—In the **vrf-blue** routing instance, overrides the **default-vpn-source** statement. The **vrf-blue** routing instance (because it contains the **unnel-source** statement) overrides the **default-vpn-source** statement and uses the address configured on **lo0.200** as the source address. This address is 192.27.11.136 in this example. However, **vrf-white** does not contain the **tunnel-source** statement. In this case, the **default-vpn-source** statement takes effect, and **vrf-white** uses the address configured on **lo0.0** as the source address. This address is 192.27.0.136 in this example.

Figure 67 on page 519 shows the topology used in this example.

Figure 67: VPN Tunnel Source Topology




This example shows the Junos OS configuration 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 PE2**

```
set interfaces fe-4/3/0 description "TO CE2"
set interfaces fe-4/3/0 unit 0 family inet address 1.2.3.1/24
```

```
set interfaces fe-4/3/0 unit 0 family iso
set interfaces fe-4/3/0 unit 0 family mpls
set interfaces so-2/3/0 description "TO PE1"
set interfaces so-2/3/0 sonet-options fcs 32
set interfaces so-2/3/0 sonet-options no-payload-scrambler
set interfaces so-2/3/0 unit 0 family inet address 1.1.2.2/24
set interfaces so-2/3/0 unit 0 family iso
set interfaces so-2/3/0 unit 0 family mpls
set interfaces lo0 unit 1 family inet address 10.255.70.103/32
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 10.255.70.103
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 10.10.10.10
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 ldp interface fxp0.0 disable
set protocols pim interface lo0.0
set protocols pim interface all
set protocols pim interface fxp0.0 disable
set protocols pim default-vpn-source
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 vpna instance-type vrf
set routing-instances vpna interface fe-4/3/0.0
set routing-instances vpna interface lo0.1
set routing-instances vpna route-distinguisher 1:0
set routing-instances vpna provider-tunnel pim-ssm group-address 232.1.1.1
set routing-instances vpna vrf-target target:1:1
set routing-instances vpna protocols ospf export bgp_ospf
set routing-instances vpna protocols ospf area 0.0.0.0 interface lo0.1 passive
set routing-instances vpna protocols ospf area 0.0.0.0 interface fe-4/3/0.0
set routing-instances vpna protocols pim mvpn autodiscovery inet-mdt
set routing-instances vpna protocols pim rp local address 2.2.2.2
set routing-instances vpna protocols pim interface lo0.1
set routing-instances vpna protocols pim interface fe-4/3/0.0
set routing-instances vpna protocols mvpn autodiscovery-only intra-as inclusive
set routing-options autonomous-system 1
```

**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 default draft-rosen behavior and configure a VPN tunnel source in a routing instance:

1. On the PE routers, configure the interfaces.

```
[edi interfacest]
user@PE2# set fe-4/3/0 description "TO CE2"
```

```
user@PE2# set fe-4/3/0 unit 0 family inet address 1.2.3.1/24
user@PE2# set fe-4/3/0 unit 0 family iso
user@PE2# set fe-4/3/0 unit 0 family mpls

user@PE2# set so-2/3/0 description "TO PE1"
user@PE2# set so-2/3/0 sonet-options fcs 32
user@PE2# set so-2/3/0 sonet-options no-payload-scrambler
user@PE2# set so-2/3/0 unit 0 family inet address 1.1.2.2/24
user@PE2# set so-2/3/0 unit 0 family iso
user@PE2# set so-2/3/0 unit 0 family mpls

user@PE2# set lo0 unit 1 family inet address 10.255.70.103/32
```

2. Configure the MPLS and a signaling protocol.

```
[edit protocols mpls]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable

[edit protocols ldp]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
```

3. Configure the routing protocols in the master instance.

```
[edit protocols bgp group int]
user@PE2# set type internal
user@PE2# set local-address 10.255.70.103
user@PE2# set family inet unicast
user@PE2# set family inet-vpn unicast
user@PE2# set family inet-vpn multicast
user@PE2# set family inet-mdt signaling
user@PE2# set neighbor 10.10.10.10

[edit protocols ospf area 0.0.0.0]
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
```

4. Configure PIM with the default-vpn-source statement.

```
[edit protocols pim]
user@PE2# set interface lo0.0
user@PE2# set interface all
user@PE2# set interface fxp0.0 disable
user@PE2# set default-vpn-source
```

5. Configure the routing policy.

```
[edit policy-options policy-statement bgp_ospf term 1]
user@PE2# set from protocol bgp
user@PE2# set then accept
```

6. Configure the routing instance.

```
[edit routing-instances vpna]
user@PE2# set instance-type vrf
user@PE2# set interface fe-4/3/0.0
```

```
user@PE2# set interface lo0.1
user@PE2# set route-distinguisher 1:0
user@PE2# set provider-tunnel pim-ssm group-address 232.1.1.1
user@PE2# set vrf-target target:1:1
user@PE2# set protocols ospf export bgp_ospf
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 fe-4/3/0.0
user@PE2# set protocols pim mvpn autodiscovery inet-mdt
user@PE2# set protocols pim rp local address 10.255.70.103
user@PE2# set protocols pim interface lo0.1
user@PE2# set protocols pim interface fe-4/3/0.0
user@PE2# set protocols mvpn autodiscovery-only intra-as inclusive
```

7. Configure the routing table options.

```
[edit routing-options]
user@PE2# set autonomous-system 1
```

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

```
user@PE2# commit
```

***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@PE2# show interfaces
fe-0/3/0 {
    description "TO CE2";
    unit 0 {
        family inet {
            address 1.2.3.1/24;
        }
        family iso;
        family mpls;
    }
}
so-2/3/0 {
    description "TO PE1";
    sonet-options {
        fcs 32;
        no-payload-scrambler;
    }
    unit 0 {
        family inet {
            address 1.1.2.2/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 1 {
```

```
      family inet {
        address 10.255.70.103/32;
      }
    }
}
```

```
user@PE2# show protocols
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
bgp {
    group int {
        type internal;
        local-address 10.255.70.103;
        family inet {
            unicast;
        }
        family inet-vpn {
            unicast;
            multicast;
        }
        family inet-mdt {
            signaling;
        }
        neighbor 10.10.10.10;
    }
}
ospf {
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
ldp {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
pim {
    interface lo0.0;
    interface all;
    interface fxp0.0 {
        disable;
    }
    default-vpn-source;
}
```

```
user@PE2# show policy-options
policy-statement bgp_ospf {
    term 1 {
        from protocol bgp;
```

```
      then accept;
    }
}
```

```
user@PE2# show routing-intances
vpna {
    instance-type vrf;
    interface fe-0/3/0.0;
    interface lo0.1;
    route-distinguisher 1:0;
    provider-tunnel {
        pim-ssm {
            group-address 232.1.1.1;
        }
    }
    vrf-target target:1:1;
    protocols {
        ospf {
            export bgp_ospf;
            area 0.0.0.0 {
                interface lo0.1 {
                    passive;
                }
                interface fe-0/3/0.0;
            }
        }
        pim {
            mvpn {
                autodiscovery {
                    inet-mdt;
                }
            }
            rp {
                local {
                    address 2.2.2.2;
                }
            }
            interface lo0.1;
            interface fe-0/3/0.0;
        }
        mvpn {
            autodiscovery-only {
                intra-as {
                    inclusive;
                }
            }
        }
    }
}
```

```
user@PE2# show routing-options
autonomous-system 1;
```

### Verification

To verify the configuration, run the **show pim mdt instance vpna** should show the data MDT and the VPN tunnel source. Other useful commands are **show pim join** (main instance), **show multicast route**, and **show pim neighbors**.

**Related Documentation**

- Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs on page 493
- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 506

## Examples: Configuring Data MDTs



### 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.

## 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 54" in the Multicast Protocols Configuration Guide and ""Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 503" in the Multicast Protocols Configuration 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 11 on page 529. Key configuration statements for the VRF instance to which your PE router is attached are highlighted in Table 12 on page 530. For complete configuration details, see ""Example: Configuring Source-Specific Multicast for Draft-Rosen Multicast VPNs" on page 507" in the Multicast Protocols Configuration 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 11 on page 529 highlights some of these statements[†].

### Table 11: Data MDTS—Key Prerequisites in the Master Instance

| Statement | Description |
|---|---|
| [edit protocols]<br>pim {<br>  interface *interface-name* <*options*>;<br>} | Enables the PIM protocol on PE router interfaces. |
| [edit protocols]<br>bgp {<br>  group *name* {<br>  type internal;<br>  peer-as *autonomous-system*;<br>  neighbor *address*;<br>  family inet-mdt {<br>  signaling;<br>  }<br>  }<br>}<br><br>[edit routing-options]<br>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]<br>multicast {<br>  ssm-groups [ *ip-addresses* ];<br>} | (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.<br><br>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 507" in the Multicast Protocols Configuration 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 12 on page 530 highlights some of these statements‡.

### Table 12: Data MDTs—Key Prerequisites in the VRF Instance

| Statement | Description |
|---|---|
| [edit routing-instances *name*]<br>instance-type vrf;<br>vrf-target *community*; | Creates a VRF table (***instance-name*.mdt.0**) that contains the routes originating from and destined for the Layer 3 VPN.<br><br>Creates a VRF export policy that automatically accepts routes from the ***instance-name*.mdt.0** routing table. ensures proper PE autodiscovery using the **inet-mdt** address family<br><br>You must also configure the **interface** and **route-distinguisher** statements for this type of routing instance. |
| [edit routing-instances *name*]<br>protocols {<br>  mvpn {<br>  autodiscovery-only {<br>  intra-as {<br>  inclusive;<br>  }<br>  }<br>  }<br>} | Enables the MVPN control plane for autodiscovery only, using intra-AS autodiscovery routes over an inclusive provider multicast service interface (PMSI). |
| [edit routing-instances *name*]<br>protocols {<br>  pim {<br>  mvpn {<br>  autodiscovery {<br>  inet-mdt;<br>  }<br>  }<br>  }<br>} | Configures the PE router in a VPN to use an MDT-SAFI NLRI for autodiscovery of other PE routers: |
| [edit routing-instances *name*]<br>provider-tunnel {<br>  pim-ssm {<br>  group-address *address*;<br>  }<br>} | Configures the PIM-SSM provider tunnel default MDT group address.<br><br>NOTE: For this example, it assumed that you previously configured the PIM-SSM provider tunnel default MDT for the VPN instance **ce1** with the group address **239.1.1.1**.<br><br>To verify the configuration of the default MDT tunnel for the VRF instance to which the PE router is attached, use the show pim mvpn operational mode command. |

Table 12: Data MDTs—Key Prerequisites in the VRF Instance *(continued)*

| Statement | Description |
|---|---|
| ‡ 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 507" in the Multicast Protocols Configuration 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 13 on page 531 summarizes the data MDT configuration statements for PIM-SSM provider tunnels.

Table 13: Data MDTs for PIM-SSM Provider Tunnels in a Draft-Rosen MVPN

| Statement | Description |
|---|---|
| [edit routing-instances *name*]<br>provider-tunnel {<br>  mdt {<br>  group-range *multicast-prefix*;<br>  }<br>} | 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.<br><br>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.<br><br>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*]<br>provider-tunnel {<br>  mdt {<br>  tunnel-limit *limit*;<br>  }<br>} | Configures the maximum number of data MDTs that can be created for the VRF instance.<br><br>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.<br><br>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.<br><br>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.<br><br>NOTE: For this example, you limit the number of data MDTs for the VRF instance to 10. |

Table 13: Data MDTs for PIM-SSM Provider Tunnels in a Draft-Rosen MVPN *(continued)*

| Statement | Description |
|---|---|
| [edit routing-instances *name*]<br>provider-tunnel {<br>  mdt {<br>  threshold {<br>  group *group-address* {<br>  source *source-address* {<br>  rate *threshold-rate*;<br>  }<br>  }<br>  }<br>  }<br>}<br>} | 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.<br>• **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.<br>• **source *source-address***—Unicast IP prefix of one or more multicast sources in the specified default MDT group.<br>• **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).<br>The default ***threshold-rate*** is 10 kilobits per second (Kbps).<br>NOTE:<br>For this example, you configure the following data MDT threshold:<br>• Multicast group address or address range to which the threshold limits apply—224.0.9.0/32<br>• Multicast source address or address range to which the threshold limits apply—10.1.1.2/32<br>• Data rate—10 Kbps<br>When the traffic stops or the rate falls below the threshold value, the source PE router switches back to the default MDT. |

### *Topology*

Figure 68 on page 533 shows a default MDT.

Figure 68: Default MDT

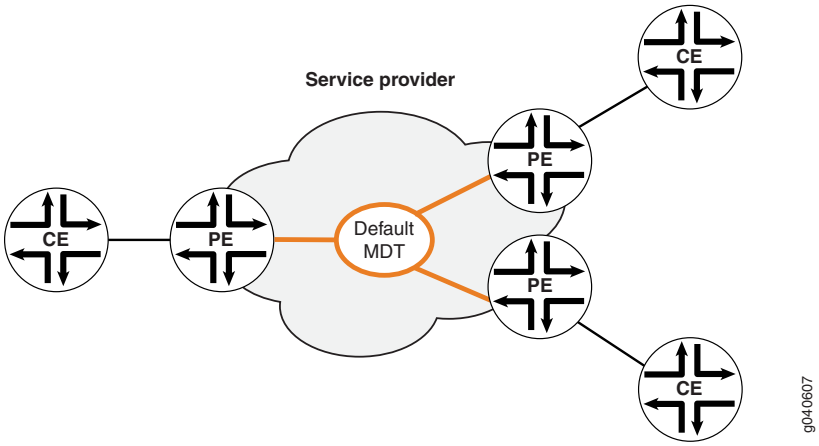
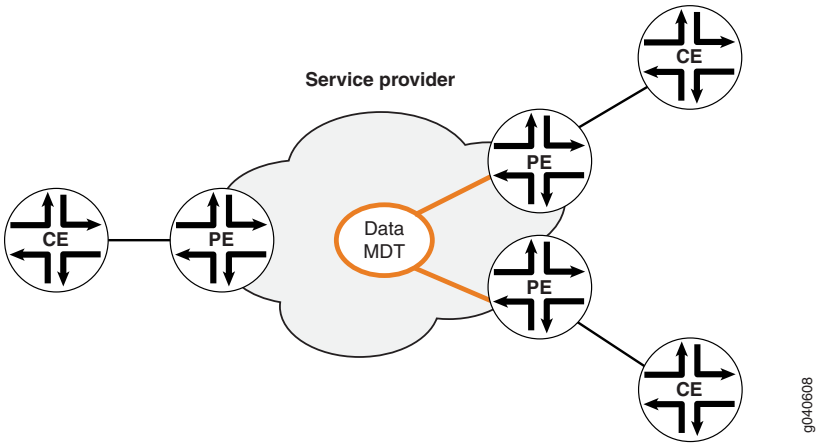



Figure 69 on page 533 shows a data MDT.

Figure 69: 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 534
- (Optional) Enabling Logging of Detailed Trace Information for Multicast Tunnel Interfaces on the Local PE Router on page 535
- Results on page 536

**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 ce1 provider-tunnel mdt group-range 239.10.10.0/24
set routing-instances ce1 provider-tunnel mdt tunnel-limit 10
```

```
set routing-instances ce1 provider-tunnel 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 {
            autodiscovery {
              inet-mdt;
            }
          }
        }
        mvpn {
          autodiscovery-only {
            intra-as {
              inclusive;
            }
          }
        }
      }
    }
```

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 536
- Monitor Data MDT Group Addresses Cached by All PE Routers in the Multicast Group on page 537
- (Optional) View the Trace Log for Multicast Tunnel Interfaces on page 537

***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.

## 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. See Multicast over Layer 3 VPNs.
- 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 54 and "Load Balancing Multicast Tunnel Interfaces Among Available PICs" on page 503.

## 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 70 on page 540 shows a default MDT.

Figure 70: Default MDT

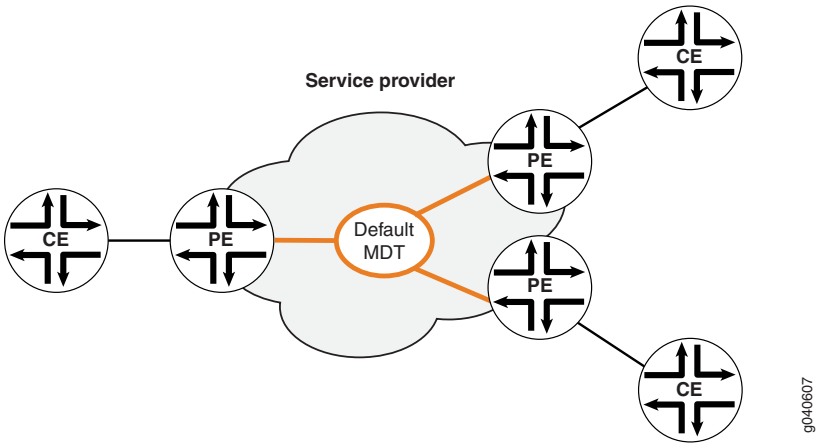
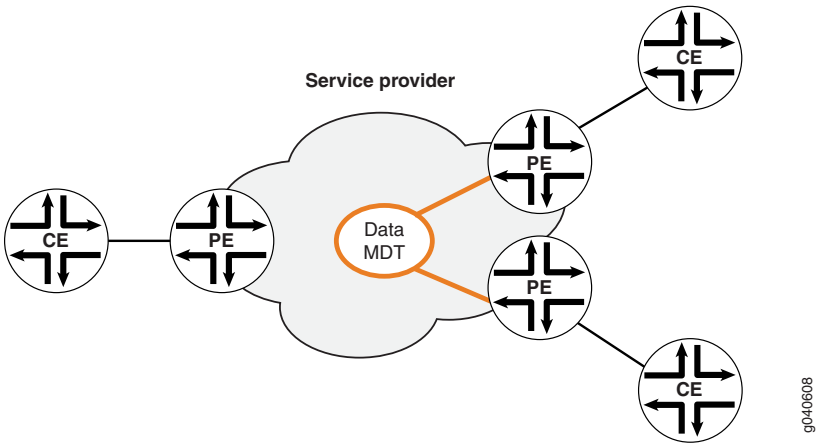



Figure 71 on page 540 shows a data MDT.

Figure 71: 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.

## 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.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure PIM Sparse Mode on the interfaces. See "Enabling PIM Sparse Mode" on page 55.

### Overview

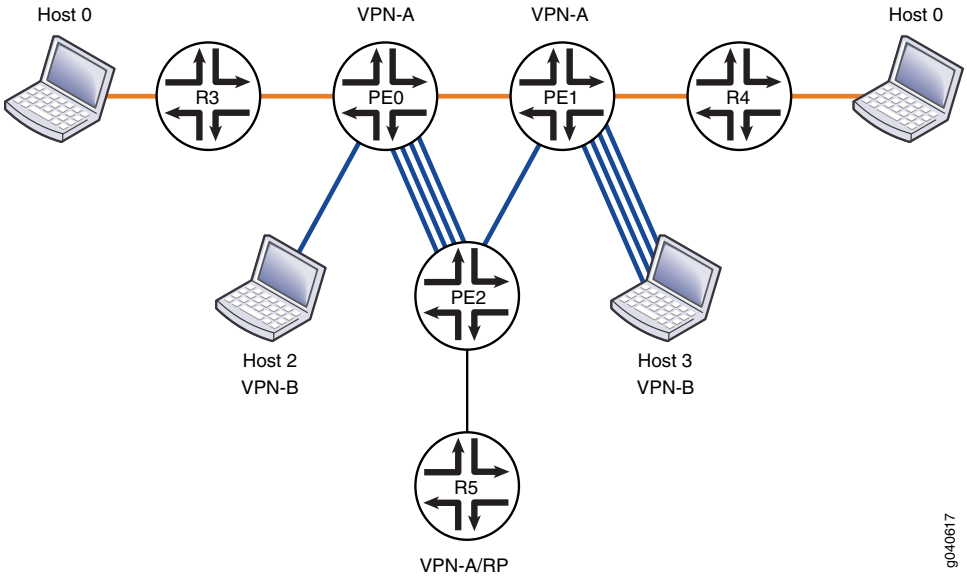
A limited number of multicast group addresses are available for use in data MDT tunnels. By default, when the available multicast group addresses are all used, no new data MDTs can be created.

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 72 on page 543 shows the topology used in this example.

Figure 72: 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

**Related Documentation**

- Example: Configuring Any-Source Draft-Rosen 6 Multicast VPNs on page 493
- Example: Configuring Source-Specific Draft-Rosen 7 Multicast VPNs on page 506
- Example: Configuring Draft-Rosen MVPN Interoperability on page 516

CHAPTER 12

# Automatic Multicast Tunneling



## Example: Configuring Automatic IP Multicast Without Explicit Tunnels



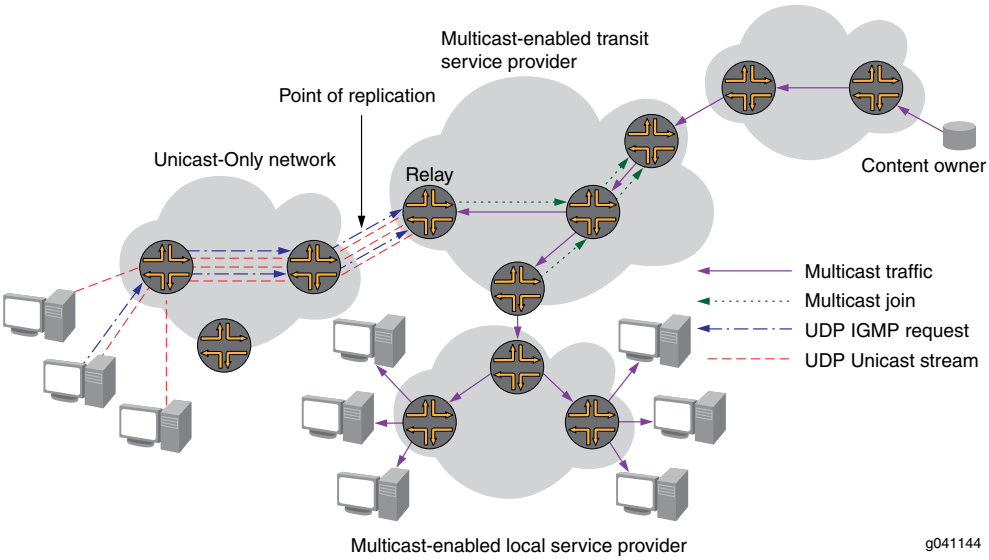
### Understanding AMT

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 except the MX80 router and all Modular Port Concentrators (MPCs) that use the Junos Trio chipset. 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 73: 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 558"Example: Configuring the AMT Protocol" on page 558

AMT supports PIM sparse mode. AMT does not support dense mode operation.

## 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)*.

## 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.

## 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.

## 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
  ```

## 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:

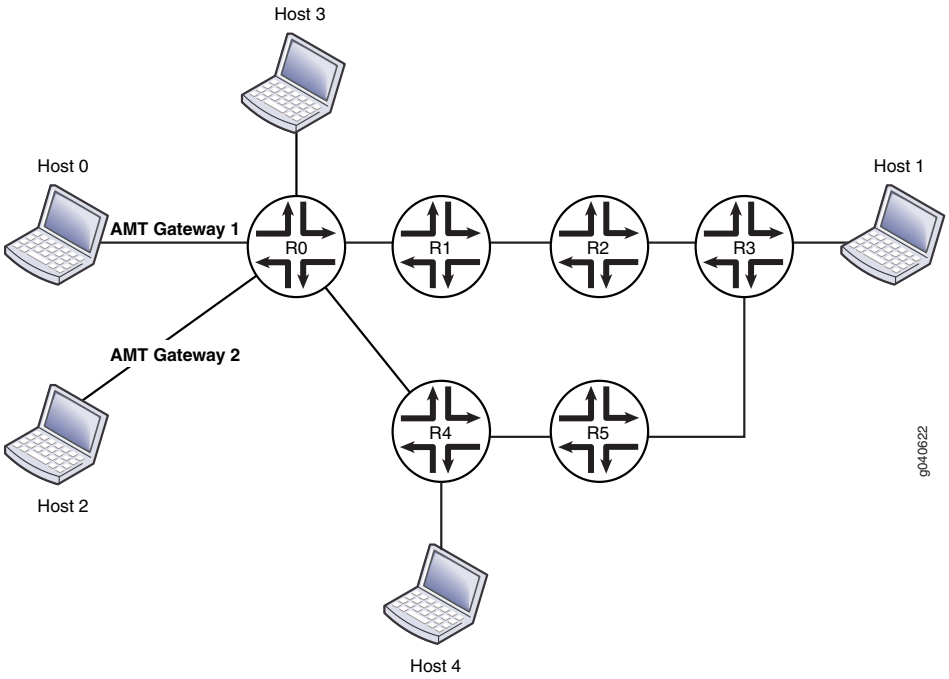
- Configure the router interfaces. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Configure a multicast group membership protocol (IGMP or MLD). See "Understanding IGMP" on page 306 and "Understanding MLD" on page 331.

### 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 74 on page 559 shows the topology used in this example.

Figure 74: 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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

**Related Documentation**

- Understanding AMT on page 549

CHAPTER 13

# Session Announcement Protocol



## 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 31.

### 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*

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.

CHAPTER 14

# Multicast Source Discovery Protocol



## 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. For more information about configuring MSDP mesh groups, see "Example: Configuring MSDP with Active Source Limits and Mesh Groups" on page 575.

Router R 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 X are as follows:

1. If Router X originated the source-active message (Router X is Router S), then Router X is also the peer-RPF neighbor, and its source-active messages are accepted.
2. If Router X is a member of the Router R mesh group, or is the configured peer, then Router X is the peer-RPF neighbor, and its source-active messages are accepted.
3. If Router X is the BGP next hop of the active multicast RPF route toward Router S (Router X installed the route on Router R), then Router X is the peer-RPF neighbor, and its source-active messages are accepted.
4. If Router X 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 X's AS number, then Router X is the peer-RPF neighbor, and its source-active messages are accepted.
5. If Router X uses the same next hop as the next hop to Router S, then Router X is the peer-RPF neighbor, and its source-active messages are accepted.
6. If Router X fits none of these criteria, then Router X 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.

## 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.

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Enable PIM. See "PIM Overview" on page 15.

### 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.

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 an MSDP source-address message if it is the DR on the upstream interface.

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 14 on page 569 for a list of match conditions.

### Table 14: MSDP Source-Active Message Filter Match Conditions

| Match Condition | Matches On |
|---|---|
| **interface** | Router interface or interfaces specified by name or IP address |

Table 14: MSDP Source-Active Message Filter Match Conditions *(continued)*

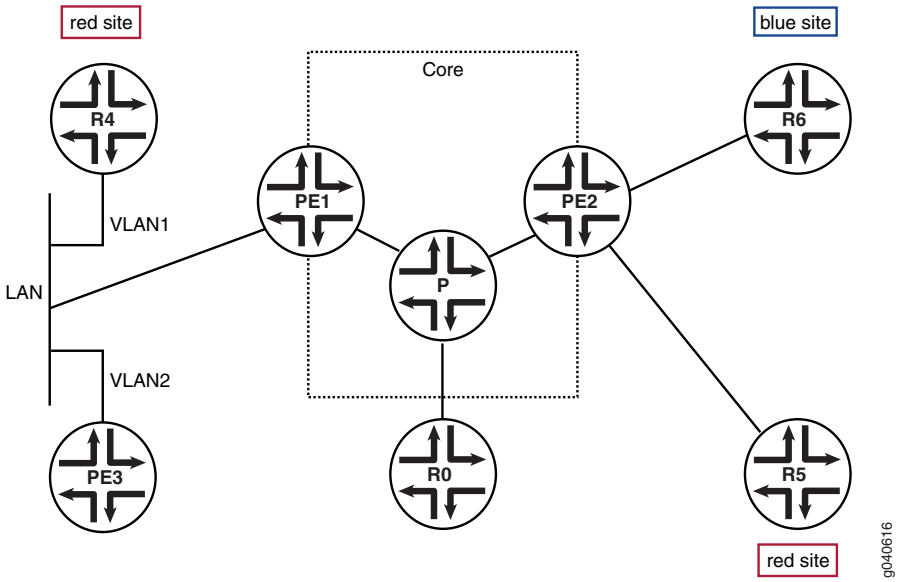
| Match Condition | Matches On |
|---|---|
| **neighbor** | Neighbor address (the source address in the IP header of the source-active message) |
| **route-filter** | Multicast group address embedded in the source-active message |
| **source-address-filter** | Multicast source address embedded in the 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 75 on page 571 shows the topology for this example.

Figure 75: 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, 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 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 "$9$z4l-3Ctp0B1EcF3eMW8-dDjH"; ## 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**

## Configuring the Interface to Accept Traffic from a Remote Source

You can configure an incoming interface to accept traffic from a remote source. A remote source is a source that is not on the same subnet as the incoming interface. This enables the remote source to be learned and advertised by MSDP so that receivers in other MSDP areas can join the source. You do not need to disable RPF checking, but you do need to ensure that the best path to reach the remote source is through the incoming interface.

In this sample configuration, the incoming interface (**ge-1/3/0**) is on a provider edge (PE) router on the receiver side of a multicast VPN.

To accept traffic from a remote source:

1. Edit the incoming interface.

   ```
   [edit protocols pim interface ge-1/3/0.0]
   user@host# set accept-remote-source
   ```

2. If the incoming interface is not the only way to reach the remote source, ensure that the best path to reach the remote source is through the incoming interface. One way to do this is to use AS path prepending 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"
   ```

   Another way to do this might be to configure a static route on the receiver side PE router to the source.

4. After the configuration is committed, use the **show pim statistics** and **show msdp source** commands to verify that the interface is accepting traffic from the remote source.

## 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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.
- Enable PIM sparse mode. See "PIM Overview" on page 15.
- Configure the router as a PIM sparse-mode RP. See "Configuring Local PIM RPs" on page 92.

## 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 discard (RED) to drop some but not all MSDP active source messages. Beginning with Junos OS 12.2, you can optionally configure a warning threshold so the device can log warning messages in the system log when a certain number of source-active messages have been received. It is helpful to review the system log messages for troubleshooting purposes and to detect if an excessive amount of source-active messages have been received. These log messages convey when the configured message limit has been exceeded, when the configured warning threshold has been exceeded, and when the number of messages drop below the configured warning threshold.

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.

The warning threshold is a percentage of maximum number of MSDP source-active messages received, so you must configure the source-active message limit to configure a warning threshold. The range for the warning threshold is 1 through 100 percent. You

can further specify the amount of time (in seconds) between the log messages. The range is 6 through 32,767 seconds.

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.

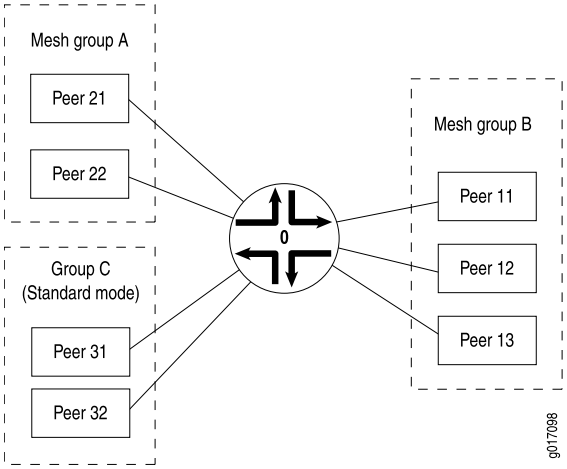
NOTE: An alternative way to bypass the peer-RPF check is to configure a default peer. In networks with only one MSDP peer, especially stub networks, the source-active message always needs to be accepted. An MSDP default peer is an MSDP peer from which all source-active messages are accepted without performing the peer-RPF check. You can establish a default peer at the peer or group level by including the default-peer statement.

Table 15 on page 578 explains how flooding is handled by peers in this example. Figure 76 on page 578 illustrates source-active message flooding between different mesh groups and peers within the same mesh group.

### Table 15: 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 76: 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.
- **active-source-limit log-warning 80**—(Optional) Applies a warning threshold of 80 percent. In this example, the active source maximum is 10,000, so the device will start logging warning messages once it receives 8,000 active source messages.

- **active-source-limit log-interval 20**—(Optional) Applies a 20 second waiting period between system log messages.
- **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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set protocols msdp data-encapsulation disable
```

```
set protocols msdp active-source-limit maximum 10000
set protocols msdp active-source-limit log-warning 80
set protocols msdp active-source-limit log-interval 20
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;
    log-warning 80;
    log-interval 20;
  }
  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

## 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. |

| Flag | Description |
|---|---|
| **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
```

## 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.

## 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 565

## Configuring Multiple Instances of MSDP

MSDP instances are supported only for VRF 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 568
- Junos OS MPLS Applications Configuration Guide
- Junos OS VPNs Configuration Guide

CHAPTER 15

# Pragmatic General Multicast



## Configuring PGM



### Understanding Pragmatic General Multicast

Multicast applications often require real-time operation. These applications cannot take advantage of Transmission Control Protocol (TCP) reliability features such as sequencing, retransmission, and flow control through windowing between sender and receiver. The User Datagram Protocol (UDP), the major transport layer alternative to TCP, is not as reliable as it needs to be for multicast traffic. Pragmatic General Multicast (PGM) is a 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 to request replacement information if the receiver application requires it. PGM is Internet Protocol number 113.

Although PGM mainly deals with the operation of multicast source and receiver, PGM-enabled routers (called PGM network elements) play a *router assistance* role in the initial delivery and potential replacement of multicast traffic. PGM routers are not mandatory in PGM, but they can provide the following benefits when placed anywhere between the source and receivers:

- Reduce the load on the multicast source by aggregating duplicate messages to the source. PGM routers are required to perform this function.
- Limit the flooding of repair data (replacement information) to only those downstream receivers that requested the repair data. PGM routers are required to perform this function.

- Act as designated local repairers (DLRs) by caching the repair data and resending it to receivers that request it later. DLR functions are a PGM option, and PGM routers are not required to perform this role.

PGM adds reliability to multicast traffic streams. It is not a complete multicast protocol like the Distance Vector Routing Multicast Protocol (DVMRP) or Protocol Independent Multicast (PIM). Adding PGM to a router does not enable the router to perform multicast functions. Instead, a PGM router with multicast capabilities and a preconfigured multicast protocol such as PIM can offer more reliable multicast services to PGM sources and receivers. PGM is not an alternative to multicast routing protocols, but an enhancement of the multicast capabilities already present and configured on the router.

## PGM Architecture and PGM Routers

PGM is defined in RFC 3208 and forms a reliable transport layer for multicast applications. Almost any multicast application can use PGM. Applications most suitable for PGM include stock market ticker update information, news reports, weather warnings, and other information that must reach multiple listeners in its entirety and in a timely fashion.

The basic PGM architecture consists of a multicast content source, one or more receivers, and zero or more routers between the source and receivers. All end devices must be PGM-enabled, although there can be non-PGM routers between the source and receiver. If all routers are non-PGM routers, then no routers are capable of the PGM router assistance function, and all PGM functions take place directly between the source and receiver.

PGM sources send sequenced content in sessions to receivers, using multicast protocols. Other, non-PGM protocols allow receivers to learn about a particular source, its sessions, and its location. PGM receivers listen to multicast original data (ODATA), detect missing content through the sequence numbers, and send unicast negative acknowledgments (NAKs) back to the source. NAKs are answered by multicast NAK confirmations (NCFs), which suppress any NAKs from receivers on the same subnet that have not yet sent a NAK upstream. The source sends multicast repair data (RDATA) to receivers containing the missing content. PGM routers assist in this process by making sure that the negative acknowledgments follow the same path as the outbound content upstream to the source, and by suppressing duplicate negative acknowledgments and repair information.

PGM sources must maintain a “sliding window” of retransmittable information. There is no concept of group membership in PGM, so receivers never need to communicate with the source unless they request repair data with a negative acknowledgment. However, this means that the PGM source determines the window size for each receiver, in contrast to almost all other protocols, and requires a certain processing power in each receiver. The absence of positive receiver-to-source acknowledgments also means that PGM scales well and cuts down on control message traffic that can easily overwhelm a multicast network.

PGM receivers can start receiving a PGM session from a PGM source at any time and request any missing previous information that the receiving application needs. If the session is not long enough or if the transmit window is too small so that the source does not maintain a long session history, the receiver cannot get all required information.

## PGM-Enabled Source

A PGM-enabled source of multicast content generates sequenced packets of ODATA that are multicast to receivers. Interleaved with the content packets are source path messages (SPMs), which tell PGM routers and receivers about their upstream next-hop PGM device—either another PGM router or the PGM source.

ODATA packets and SPMs are multicast from the source. A PGM router always appends its own IP address to the SPM before it is multicast on the downstream interfaces. The SPMs are sent by the source and upstream PGM routers with the router alert option set in the IP headers so that PGM routers do not have to examine every packet in the session for SPM packets.

The PGM source acknowledges a received NAK by multicasting NAK confirmations (NCFs) downstream to the next PGM device on the path to the receiver. NCFs make sure that PGM routers and receivers do not bombard sources with NAKs. Downstream PGM routers suppress all subsequent NAKs that indicate the same missing information once one NCF is received from the upstream device.

The PGM source also responds to NAKs by multicasting RDATA packets with the same sequence number as the one indicated by the NAK. RDATA packets have the router alert option set in the IP header so that PGM routers can distinguish them from ODATA packets.

PGM sources organize their packets in sessions. PGM sources are not required to retain copies of information older than the current session, although they might. Long sessions are not necessarily kept on the source in their entirety.

PGM sources identify themselves through a global source ID (GSID). This globally unique source identifier is formed from the low-order 48 bits of the Message Digest 5 (MD5) signature of the Domain Name System (DNS) name of the source.

## PGM-Enabled Receivers

The PGM architecture requires one or more PGM-enabled receivers of the multicast content generated by a PGM source. PGM receivers accept all types of downstream PGM messages: ODATA, SPMs, NCFs, and RDATA.

Receivers process the ODATA packets as they arrive from the source, constantly checking the 32-bit sequence number in the ODATA PGM header for gaps in the sequence. If the receiver detects missing information, it generates a NAK for that sequence number. The NAK is unicast upstream to the PGM next hop, which is a router or the source, as determined by the last address in the received SPM.

A receiver detects that its NAK was received by the PGM next hop when it receives an NCF in response to its NAK. If several receivers on a subnet are missing the same ODATA packet, receivers getting an NCF for the packet before sending a NAK suppress the NAK. If a receiver does not get an NCF in response to a NAK, the receiving application can send a NAK again or continue, with the certainty that information is missing.

After the NCF, PGM receivers are sent an RDATA packet with the same sequence number indicated in the NAK and a copy of the missing ODATA. NCFs and RDATA can originate

from the source or a router acting as a designated local repairer (DLR) for a subnet. The receiver now has complete information about what is missing.

PGM receivers can request almost anything from the PGM source. However, because the source determines the window size, there is no guarantee that older information is available.

## PGM-Enabled Routers

Multicast-capable routers can implement the PGM router assistance functions, although not all multicast routers must be PGM-enabled routers. Mandatory PGM router assistance functions include aggregating duplicate NAKs sent to the source to reduce the load on the multicast source, generating NCFs in response to NAKs, and flooding RDATA packets to only those downstream receivers that requested it with a NAK. Optionally, a PGM router can be a DLR, caching PGM information and cutting down on network traffic by resending RDATA packets locally.

There can be zero or more PGM-enabled network elements (routers) between the source and receiver. If there are no PGM routers between the source and receiver, then all PGM messages flow directly between the source and receiver, and no router assistance functions are possible. Both PGM and non-PGM routers can be freely mixed on a network because PGM is a transport layer protocol and is not involved with router multicast functions.

PGM routers also receive SPMs from the source or an upstream PGM router and forward them downstream, inserting the router's own downstream IP interface address into the SPM so that receivers always know their upstream PGM next hop.

When a PGM router receives unicast NAKs from a downstream PGM router or receiver, the router unicasts one NAK for each missing sequence number to the next-hop PGM device upstream toward the source. The address of the PGM next-hop device is determined by received SPMs.

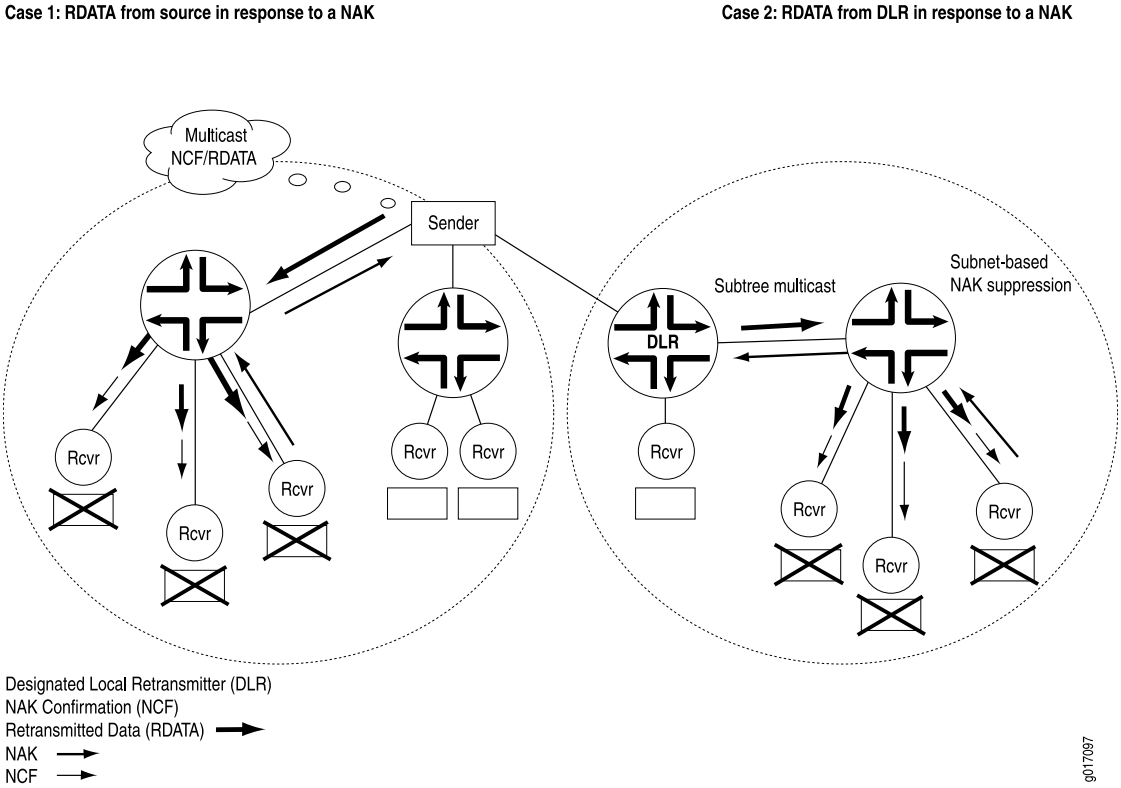
The PGM router multicasts NCFs in response to received NAKs on the downstream interfaces that received the NAKs. NCFs are not multicast on interfaces that have not received NAKs.

PGM routers must multicast all ODATA and RDATA packets that they receive from upstream PGM devices. Normal multicast protocols are used to determine downstream interfaces.

If the PGM router is a DLR, it responds to received NAKs with an NCF and with its own RDATA packet. NAKs are not forwarded upstream from a DLR.

Figure 77 on page 591 shows the overall PGM architecture and the role of PGM-enabled routers.

Figure 77: PGM Architecture and General Operation




The figure shows only NAKs, NCFs, and RDATA flows. RDATA can come from either the source (left) or a DLR router (right). In both cases, unicast NAKs from a receiver are forwarded upstream by the routers, and multicast NCFs are generated downstream. Subnet NAK suppression is shown, as well as RDATA from the source or DLR sent only to the portions of the network requesting it.

## PGM Configuration Guidelines

Pragmatic General Multicast (PGM) allows the router to participate in defined PGM router assistance functions between PGM-enabled sources and receivers. Although PGM is a transport layer protocol and does not do IP packet routing, PGM must be explicitly configured on the router.

To enable PGM globally on the router, include the **pgm** statement:

```
pgm;
```

You can include this statement at the following hierarchy levels:

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

To trace the operation of PGM, include the **traceoptions** statement:

```
traceoptions {
    flag flag <flag-modifier>;
```

}

You can include this statement at the following hierarchy levels:

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

You can specify the following PGM-specific options in the **flag** statement:

- **all**—Trace all PGM packets.
- **init**—Trace all PGM initialization events.
- **packets**—Trace all PGM packet processing.
- **parser**—Trace all PGM parser processing.
- **route-socket**—Trace all PGM route-socket events.
- **show**—Trace all PGM **show** command servicing.
- **state**—Trace all PGM state transitions.

By default, PGM is enabled on every interface of the router, but global, explicit configuration is required. No options are available for PGM operation.

**Related Documentation**

-

CHAPTER 16

# Distance Vector Multicast Routing Protocol



## Examples: Configuring DVMRP



### Understanding DVMRP

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.

## Configuring DVMRP

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.

## Example: Configuring DVMRP

This example shows how to use DVMRP to announce routes used for multicast routing as well as multicast data forwarding.



### Requirements

Before you begin:

- Configure the router interfaces. See the Junos® OS Network Interfaces.

- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.

## 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

## Example: Configuring DVMRP to Announce Unicast Routes

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. See the Junos® OS Network Interfaces.
- Configure an interior gateway protocol or static routing. See the Junos OS Routing Protocols Configuration Guide.

### 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, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
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

## Tracing DVMRP 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 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. |

| Flag | Description |
|---|---|
| **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
   ```

**Related Documentation**

- Understanding DVMRP on page 593

CHAPTER 17

# PIM 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.

**Description** Accept traffic from a remote source. A remote source is a source that is not on the same subnet as the incoming interface. This statement enables the remote source to be learned and advertised by MSDP so that receivers in other MSDP areas can join the source. You do not need to disable RPF checking, but you do need to ensure that the best path to reach the remote source is through the incoming 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 575
- Example: Allowing MBGP MVPN Remote Sources on page 421

## 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.

**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 72
- Example: Configuring Bidirectional PIM on page 78

## 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.

**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 92

## 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.

**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.

**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 96

## 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.

**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 146
- Configuring BFD Authentication for PIM on page 150
- authentication on page 611

## 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.

**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 101

## 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 140

## authentication (Protocols PIM)

**Syntax**

```
authentication {
  algorithm algorithm-name;
  key-chain key-chain-name;
  loose-check;
}
```

**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 9.6.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

**Description**

Configure the algorithm, security keychain, and level of authentication for BFD sessions running on PIM interfaces.

The remaining statements are explained separately.

**Options**

The 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**

- Configuring BFD Authentication for PIM on page 150
- Configuring BFD for PIM on page 148
- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 146
- bfd-liveness-detection on page 614
- key-chain (Protocols PIM) on page 646
- loose-check on page 650

## 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.

**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 111

## 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 72
- Example: Configuring Bidirectional PIM on page 78

## 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*],
[edit protocols pim interface *interface-name* family (inet | inet6)],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[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.

**Description**

Configure bidirectional forwarding detection (BFD) timers and authentication for PIM.

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**

- Configuring BFD for PIM on page 148
- Configuring BFD Authentication for PIM on page 150

## 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.

**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 72
- Example: Configuring Bidirectional PIM on page 78

## 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.

**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 72
- Example: Configuring Bidirectional PIM on page 78

## 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.

**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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108

## 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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108
- bootstrap-import on page 619

## 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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108
- **bootstrap-export on page 618**

## 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 106

## 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 175

## 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.

**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.

**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 148
- bfd-liveness-detection on page 614
- threshold on page 690

## 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.

**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 72
- Example: Configuring Bidirectional PIM on page 78

## disable (PIM Graceful Restart)

**Syntax**

```
disable;
```

**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**

Explicitly disable PIM sparse mode graceful restart.

**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 170

## 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 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 42
- disable (PIM Graceful Restart) on page 623

## 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 50

## 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 129
- rp-register-policy on page 682

## 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.

**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 117

## export (Protocols PIM 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.

**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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108
- **import (Protocols PIM Bootstrap) on page 639**

## 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.

**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 121

## 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.

**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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108

## family (Protocols PIM)

**Syntax**

```
family (inet | inet6) {
  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],
[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 11.3 for the QFX Series.

**Description**

Enable the PIM protocol for the specified family.

**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 statement is explained separately.

**Related Documentation**

- Disabling PIM on page 42
- disable (PIM Graceful Restart) on page 623
- disable (PIM) on page 624

## 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.

**Related Documentation**

- Configuring PIM and the Bidirectional Forwarding Detection (BFD) Protocol on page 146
- Disabling PIM on page 42

## 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.

**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 92

## 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.

**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 170

## 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.

**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 250

## 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.
Support for bidirectional RP addresses introduced in Junos OS Release 12.1.

**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 92 in the Multicast Protocols Configuration Guide
- Configuring PIM Embedded RP for IPv6 on page 117 in the Multicast Protocols Configuration Guide
- Example: Configuring Bidirectional PIM on page 78

## 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 206

## hello-interval (Protocols PIM)

**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 on page 637
- Modifying the PIM Hello Interval on page 39

## 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:** 0 through 255

**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 92 in the Multicast Protocols Configuration Guide
- Example: Configuring Bidirectional PIM on page 78

## 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 197
- Configuring PIM Join Load Balancing on page 56
- clear pim join-distribution on page 935
- join-load-balance on page 644
- standby-path-creation-delay on page 688

## 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.

**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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108
- export (Protocols PIM Bootstrap) on page 627

## 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.

**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 125

## 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 142

## 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) {
    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;
  }
  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.

**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 40

## 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.

**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 197
- Configuring PIM Join Load Balancing on page 56
- **clear pim join-distribution on page 935** in the Junos OS Operational Mode Commands

## 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.

**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 59

## 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.

**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 150
- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 146
- authentication on page 611

## 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.
The remaining statements are explained separately.

**Description**

Configure the routing device's RP properties.

**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 92

## 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 101

## 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 934

## 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.

**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 150
- Understanding Bidirectional Forwarding Detection Authentication for PIM on page 146
- **authentication on page 611**

## 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 111

## 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 934

## 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 117

## 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.

**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 148

## 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.

**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 148
- bfd-liveness-detection on page 614
- minimum-interval on page 654
- threshold on page 691

## 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.

**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 148

## 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 173 in the Multicast Protocols Configuration Guide
- Configuring PIM Sparse-Dense Mode Properties on page 175 in the Multicast Protocols Configuration Guide
- Example: Configuring Bidirectional PIM on page 78

## 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.

**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 148

## 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 120

## 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.

**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 250

## 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.

**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 148
- bfd-liveness-detection on page 614

## 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 170 in the Multicast Protocols Configuration Guide
- Understanding Bidirectional PIM on page 72
- Example: Configuring Bidirectional PIM on page 78

## 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 devices.

**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.

## 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.

**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 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** *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 72
- Example: Configuring Bidirectional PIM on page 78
- robustness-count on page 679

## 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 91
- Configuring PIM Auto-RP on page 111

## 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.

**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 59
- propagation-delay on page 674
- reset-tracking-bit on page 676

## 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.

**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 537
- Configuring PIM Dense Mode Properties on page 173
- Configuring PIM Sparse-Dense Mode Properties on page 175

## Configuring Virtual Tributary Mapping

You can configure virtual tributary mapping standard as either International Telephony Union standard (itu-t) or KLM standard (klm). Here, the KLM standard is set by default.

To configure virtual tributary mapping on Channelized STM1 IQ and IQE PICs:

1. In configuration mode, go to the **[edit interfaces *interface-name* sonet-options]** hierarchy level, where the interface name is **cau4-*fpc/pic/port***.

   ```
   [edit]
   user@host# edit interfaces cau4-fpc/pic/port sonet-options
   ```

2. Configure virtual tributary mapping option in KLM standard. By default, virtual tributary mapping uses KLM standard.

   ```
   [edit interfaces cau4-fpc/pic/port sonet-options]
   user@host# set vtmapping klm
   ```

3. Configure virtual tributary mapping option in ITU-T standard alternatively.

   ```
   [edit interfaces cau4-fpc/pic/port sonet-options]
   user@host# set vtmapping itu-t
   ```

To configure virtual tributary mapping on STM1 PIC:

1. In configuration mode, go to the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level.

   ```
   [edit]
   user@host# edit chassis fpc slot-number pic pic-number
   ```

2. Configure virtual tributary mapping option in KLM mode. By default, virtual tributary mapping uses KLM mode.

   ```
   [edit chassis fpc slot-number pic pic-number]
   user@host# set vtmapping klm
   ```

3. Configure virtual tributary mapping option in ITU-T mode alternatively.

   ```
   [edit chassis fpc slot-number pic pic-number]
   ```

user@host# set vtmapping itu-t

Configuring Channelized STM1 Interfaces lists the KLM mappings used by the Channelized STM1-to-E1 PIC interfaces.

**Related Documentation**

- Configuring SONET/SDH Physical Interface Properties
- SONET/SDH Physical Interfaces Configuration Hierarchy
- SONET/SDH Physical Interface Properties Overview

## 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.

**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.

**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 250

## 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.

**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 106
- Configuring PIM Bootstrap Properties for IPv4 or IPv6 on page 108
- bootstrap-priority on page 620

## 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.

**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 a 32-bit number

**Default:** 0 (The routing device has the least likelihood of becoming the designated router.)

**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 49

## 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.
Support for bidirectional RP addresses introduced in Junos OS Release 12.1.

**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 92 in the Multicast Protocols Configuration Guide
- Example: Configuring Bidirectional PIM on page 78

## 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.

**Description** Set a delay for implementing a PIM prune message on the upstream router on a multicast network for which join suppression has been enabled. The router waits for the prune pending period to detect whether a join message is currently being suppressed by another router.

**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 59
- override-interval on page 664
- reset-tracking-bit on page 676

## 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 206
- clear pim join on page 934
- clear pim register on page 937

## 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.

**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 59
- override-interval on page 664
- propagation-delay on page 674

## restart-duration (Protocols PIM)

**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 170

## 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.

**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 241

## 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.

**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 72
- Example: Configuring Bidirectional PIM on page 78

## rp

**Syntax**

```
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-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.

**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.

**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 51

## 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.

**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 129
- dr-register-policy on page 625

## 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.

**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.

**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 101

## 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]

**Release Information** Statement introduced in JUNOS Release 10.4.
Statement introduced in Junos OS Release 11.3 for the QFX Series.

**Description** Configure the PIM RPF next-hop neighbor for a specific group and source for a VRF routing instance.

The remaining statements are explained separately.

**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 250

## 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 206
- clear pim join on page 934

## 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.

**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.

**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 250

## 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.

**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.

**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 142

## 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 197
- Configuring PIM Join Load Balancing on page 56
- clear pim join-distribution on page 935
- join-load-balance on page 644
- idle-standby-path-switchover-delay on page 638

## 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.

**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.

**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 96

## 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.

**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 148
- bfd-liveness-detection on page 614
- detection-time on page 622
- minimum-interval on page 654
- minimum-receive-interval on page 656

## 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.

**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 148
- bfd-liveness-detection on page 614

## 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 934

## 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.

**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 40
- Tracing DVMRP Protocol Traffic on page 601
- Tracing MSDP Protocol Traffic on page 581
- Configuring PIM Trace Options on page 40

## traceoptions (PIM Snooping)

**Syntax**

```
traceoptions {
  file;
  flag [all | general | hello | join | normal | packets | policy | prune | route | state | task | timer
    ]
  [detail | disable | receive | send];
}
```

**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.

**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 219

## 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.

**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.

**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 148
- bfd-liveness-detection on page 614
- threshold on page 691
- minimum-interval on page 655
- minimum-receive-interval on page 656

## tunnel-devices

**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 503

## version (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.

**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 148

## version (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.

**Description** Specify the version of PIM.

**Options** *version*—PIM version number.

**Range:** 1 or 2

**Default:** PIMv1 for rendezvous point (RP) mode (at the **[edit protocols pim rp static address *address*]** hierarchy level). PIMv2 for 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 55
- Configuring PIM Dense Mode Properties on page 173
- Configuring PIM Sparse-Dense Mode Properties on page 175

## 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]

**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**
- vlan

## 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.

**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 Configuration Guide

## 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.

**Description**

Use a wildcard for the multicast source instead of (or in addition to) a specific multicast source.

The remaining statements are explained separately.

**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 250

CHAPTER 18

# IGMP Configuration Statements

## 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 322

## 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 322

## disable (Protocols IGMP)

**Syntax** disable;

**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.

**Description** Disable IGMP 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 IGMP on page 326

## 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 315

## 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.

**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 315

## 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.

**Description** Specify the number of static groups to be created.

**Options** *number*—Number of static groups.

**Default:**

**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 315

## 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.

**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 315

## group-limit

**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.

**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 323
- group-threshold on page 711
- log-interval on page 716

## 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.

**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 312

## 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 323
- group-limit on page 709
- log-interval on page 716

## igmp

**Syntax**

```
igmp {
  accounting;
  interface interface-name {
    disable;
    (accounting | no-accounting);
    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.

**Description** Enable IGMP on the router. IGMP must be enabled for the router to receive multicast packets.

The remaining statements are explained separately.

**Default** IGMP is disabled on the router. 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 309

## 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.

Starting in Junos OS Release 9.3, both IGMP version 2 and IGMP version 3 do host tracking when the **immediate-leave** statement is configured. This means that the multicast group leaves only when the last host leaves. The routing 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 311

## interface (Protocols IGMP)

**Syntax**

```
interface interface-name {
  disable;
  (accounting | no-accounting);
  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.

**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 309

## log-interval (Protocols IGMP Interface)

**Syntax** log-interval *seconds*;

**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 minimum time interval (in seconds) between sending consecutive log messages to the system log for multicast groups. 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 323
- group-limit on page 709
- group-threshold on page 711

## 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.

**Description** Limit the transmission rate of IGMP packets

**Options** **packets-per-second**—Maximum number of IGMP packets transmitted in one second by the router.

**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 315

## oif-map

**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.

**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 270

## 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.

**Description** Specify that IGMP run on the interface and either not send and receive control traffic or selectively send and receive control traffic such as IGMP 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 would be equivalent to not using the passive statement.

**Options**

**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 270
- Enabling IGMP on page 309

## promiscuous-mode (Protocols IGMP)

**Syntax** promiscuous-mode;

**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 8.3.
Statement introduced in Junos OS Release 9.0 for EX Series switches.
Statement introduced in Junos OS Release 9.2 for dynamic profiles.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

**Description** Specify that the interface accepts IGMP reports from hosts on any subnetwork. Note that 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.

**Required Privilege Level** routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

**Related Documentation**
- Configuring a Dynamic Profile for Client Access
- Accepting IGMP Messages from Remote Subnetworks on page 312

## 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.

**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 IGMP Host-Query Message Interval on page 309
- query-last-member-interval (Protocols IGMP) on page 720
- query-response-interval (Protocols IGMP) on page 721

## 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.

**Description** Specify how often the querier router 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 313
- query-interval (Protocols IGMP) on page 719
- query-response-interval (Protocols IGMP) on page 721

## 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.

**Description** Specify how long the querier router 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 310
- query-interval (Protocols IGMP) on page 719
- query-last-member-interval (Protocols IGMP) on page 720

## 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.

**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 314

## 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.

**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.

**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 315

## 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.

**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 315

## 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.

**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 315

## 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.

**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 263

## 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.

**Description** Apply an SSM map policy to an IGMP interface.

**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 326

## 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.

**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.

**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 315

## 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.

**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 324

## 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*]

**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 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 315

CHAPTER 19

# MLD Configuration Statements

## 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 349

## 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 349

## 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 354

## 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 342

## 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.

**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 342

## 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 342

## group-increment (Protocols MLD)

**Syntax** group-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 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 342

## group-limit

**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.

To confirm the configured group limit on the interface, use the show mld interface command.

**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**

- Configuring the Number of MLD Multicast Group Joins on Logical Interfaces on page 351

## 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 router running MLD version 1 or version 2 (MLDv1 or MLDv2), 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).

**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 339

## 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 351
- group-limit on page 735
- log-interval on page 739

## 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 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 MLD on page 338

## interface (Protocols MLD)

**Syntax**

```
interface interface-name {
  disable;
  (accounting | no-accounting);
  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.

**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 335

## log-interval (Protocols MLD Interface)

**Syntax** log-interval *seconds*;

**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 minimum time interval (in seconds) between sending consecutive log messages to the system log for multicast groups. 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 351
- group-limit on page 735
- group-threshold on page 736

## 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 router.

**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 341

## mld

**Syntax**

```
mld {
  accounting;
  interface interface-name {
    (accounting | no-accounting);
    disable;
    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;
  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.

**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 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 MLD on page 335

## oif-map

**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 270

## 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 270

## 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 336
- query-last-member-interval (Protocols MLD) on page 744
- query-response-interval (Protocols MLD) on page 745

## 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]

**Release Information** Statement introduced before Junos OS Release 7.4.

**Description** Specify how often the querier router 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**
- Modifying the MLD Last-Member Query Interval on page 338
- query-interval (Protocols MLD) on page 744
- query-response-interval (Protocols MLD) on page 745

## query-response-interval (Protocols MLD)

**Syntax** query-response-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.
Statement introduced in Junos OS Release 9.0 for EX Series switches.

**Description** Specify how long the querier router waits to receive a response to a host-query message from a host.

**Options** ***seconds***—Time interval. This interval must be less than the interval between general host-query messages.

**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 337
- query-interval (Protocols MLD) on page 744
- query-last-member-interval (Protocols MLD) on page 744

## 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 340

## 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 342

## 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 342

## 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 342

## 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 263

## 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 an MLD interface.

**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 326

## 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.

**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 342

## traceoptions (Protocols MLD)

**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 mld],
[edit protocols mld]

**Release Information** Statement introduced before Junos OS Release 7.4.

**Description** Configure MLD 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 MLD trace options are those inherited from the **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 **mld-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.

MLD Tracing Flags

- **leave**—Leave group messages.
- **mtrace**—Mtrace packets. Use the **mtrace** command to troubleshoot the software.
- **packets**—All MLD packets.
- **query**—MLD 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**—Traces errors and significant events during normal packet processing

**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 MLD Protocol Traffic on page 352

## 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 336

CHAPTER 20

# IGMP Snooping Configuration Statements

## 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.

**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 361

## group-limit

**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 361

## 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 361
- multicast-router-interface on page 760

## igmp-snooping

**Syntax**

```
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;
  }
}
```

**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.

**Description**

Enable IGMP snooping on the router.

**Default** IGMP snooping is disabled on the router.

**Options** The 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**

- Understanding IGMP Snooping on page 356
- IGMP Snooping in MC-LAG Active-Active on MX Series Routers Overview

## 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 361

## interface (Bridge Domains)

**Syntax**

```
interface interface-name {
  group-limit limit;
  host-only-interface;
  multicast-router-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.

**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 361

## 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*]

**Release Information** Statement introduced in Junos OS Release 8.5.
Statement introduced in Junos OS Release 9.1 for EX Series switches.

**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-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 IGMP snooping.

Configure an interface as a bridge interface toward other multicast routers.

**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 EX Series Switches
- Example: Configuring IGMP Snooping on page 361
- Configuring IGMP Snooping (CLI Procedure)
- IGMP Snooping in MC-LAG Active-Active on MX Series Routers Overview
- host-only-interface on page 755
- show igmp-snooping membership

## proxy (Bridge Domains)

**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.

**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 361

## query-interval (Bridge Domains)

**Syntax** query-interval *seconds*;

**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 before Junos OS Release 8.5.

**Description** Configure the interval for host-query message timeouts.

**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 page 361
- query-last-member-interval (Bridge Domains) on page 763
- query-response-interval (Bridge Domains) on page 764

## query-last-member-interval (Bridge Domains)

**Syntax** query-last-member-interval *seconds*;

**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** 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 361
- query-interval on page 762
- query-response-interval on page 764

## 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-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** Specify how long to wait to receive a response to a specific query message from a host.

**Options** ***seconds***—Time interval. This interval must 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 page 361
- query-interval (Bridge Domains) on page 762
- query-last-member-interval (Bridge Domains) on page 763

## 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 361

## 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 361

## 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.

**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.

**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 361

## 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.

**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 361

## 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]

**Release Information**

Statement introduced in Junos OS Release 8.5.

**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 367
- Configuring IGMP Snooping on page 359

## vlan (Bridge Domains)

**Syntax**

```
vlan vlan-id {
  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.

**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.

**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 360
- igmp-snooping on page 756

CHAPTER 21

# Multicast Snooping Configuration Statements

## 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 373

## 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 373

## 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 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 Multicast Snooping on page 373

## 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 373
- query-response-interval (Bridge Domains) on page 764

## 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 373

## multicast-snooping-options

**Syntax**

```
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>;
  }
}
```

**Hierarchy Level**

[edit bridge-domains *bridge-domain-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name*],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*],
[edit routing-instances *routing-instance-name*],
[edit routing-instances *routing-instance-name* bridge-domains *bridge-domain-name*]

**Release Information**

Statement introduced in Junos OS Release 8.5.

**Description**

Establish multicast snooping option values.

**Options**

The 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**

- Configuring Multicast Snooping on page 372
- Enabling Bulk Updates for Multicast Snooping on page 378
- Example: Configuring Multicast Snooping on page 373

## 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 372
- multicast-snooping-options on page 774

## 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 378

## 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 373

## 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 373

## 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 372
- Example: Configuring Multicast Snooping on page 373
- Enabling Bulk Updates for Multicast Snooping on page 378
- Example: Configuring Multicast Snooping on page 373

CHAPTER 22

# Multicast Routing Options Configuration Statements

## 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 12.3 for ACX Series routers.

**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 258

## 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.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**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**

**backups *addresses***—Specify the address of backup PE routers for ingress PE redundancy when point-to-multipoint LSPs are used for multicast distribution.

**local-address *address***—Specify the address of the local PE router for ingress PE redundancy when point-to-multipoint LSPs are used for multicast distribution.

***pe-group-name***—Specify the name for the group of PE routers that provide ingress PE router redundancy for point-to-multipoint 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 PE Redundancy on page 294
- Configuring Ingress PE Router Redundancy for Point-to-Multipoint LSPs

## 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 294

## bandwidth (Multicast Flow Map)

**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.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**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 290

## 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.

**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 290

## 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.

## forwarding-cache (Multicast)

**Syntax**

```
forwarding-cache {
  family (inet | inet6) {
    threshold {
      log-warning value;
      suppress value <reuse value>;
    }
  }
  threshold {
    log-warning value;
    suppress value <reuse value>;
  }
  timeout minutes;
}
```

**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.

**Description**

Configure multicast forwarding cache properties. These properties include threshold suppression and reuse limits, the threshold at which a warning message is logged, and timeout values.

Specify a value for the threshold at which to suppress new multicast forwarding cache entries and an optional reuse value for the threshold at which the router begins to create new multicast forwarding cache entries. The range for both is from 1 through 200,000. If configured, the reuse value should be less than the suppression threshold 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.

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.

**Default**

By default, there are no limits on the number of multicast forwarding cache entries.

**Options**

**family (inet | inet6)**—(Optional) Apply the configured thresholds to either IPv4 or IPv6 multicast forwarding cache entries.

**Default:** By default, the configured thresholds are applied to both IPv4 and IPv6 multicast forwarding cache entries.

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 the Multicast Forwarding Cache on page 286

## 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.

**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 268
- Example: Configuring Multicast with Subscriber VLANs on page 270

## 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: Creating a Named Scope for Multicast Scoping on page 233

## 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.

**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 294

## maximum-bandwidth (Routing Options)

**Syntax** maximum-bandwidth *bps*;

**Hierarchy Level**
[edit dynamic-profiles *profile-name* routing-instances *instance-name* routing-options multicast interface *interface-name*],
[edit dynamic-profiles *profile-name* routing-options multicast interface *interface-name*]
[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.
**dynamic-profiles** hierarchy level added in Junos OS Release 11.2.

**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 268

## 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 {
        family (inet | inet6) {
            threshold {
                log-warning value;
                suppress value <reuse value>;
            }
        threshold {
            log-warning value;
            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.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**Description**

Configure multicast routing options properties.

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 the Multicast Forwarding Cache on page 286
- Example: Configuring a Multicast Flow Map on page 290
- Example: Configuring Source-Specific Multicast Groups with Any-Source Override on page 258

## 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 286

## 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.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**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 270

## pim-to-igmp-proxy

**Syntax**

```
pim-to-igmp-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 11.3 for the QFX Series.
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 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 remaining statement is 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-to-IGMP Message Translation on page 301

## 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.

**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 302

## 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.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**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 (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 263

## 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.

**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 231
- Example: Creating a Named Scope for Multicast Scoping on page 233
- multicast on page 792

## 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 290

## 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.

**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 270

## 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.

**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 248

## 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.

**Description**

Configure multicast scoping.

**Options**

***scope-name***—Name of the multicast scope.

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: Creating a Named Scope for Multicast Scoping on page 233

## 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.
Statement introduced in Junos OS Release 12.3 for ACX Series routers.

**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 803
- Example: Using a Scope Policy for Multicast Scoping on page 235

## 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 263

## 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.

**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 258

## 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.

**Description**

Configure SSM mapping.

**Options**

***ssm-map-name***—Name of the SSM map.

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 SSM Mapping on page 263

## 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.

**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.

**Related Documentation**

- Example: Configuring Multicast with Subscriber VLANs on page 270

## threshold (Multicast Forwarding Cache)

**Syntax**

```
threshold {
  log-warning value;
  suppress value <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.

**Description**

Configure the global 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.

To confirm the configured threshold values, use the show multicast forwarding-cache statistics command.

**Options**

**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 *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

The remaining statement is explained separately.

**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 286

## 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.

**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.

**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 286

## 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.

**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 301
- Configuring PIM-to-MLD Message Translation on page 302

CHAPTER 23

# MBGP MVPN Configuration Statements

## 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).

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

## 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 389
- Configuring Routing Instances for an MBGP MVPN
- mpls-internet-multicast on page 826
- ingress-replication on page 822

## 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.

**Related Documentation**

- Configuring VRF Route Targets for Routing Instances for an MBGP MVPN

## 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 820
- inet6-mvpn on page 821

## group (Routing Instances)

**Syntax**

```
group address {
  source source-address {
    pim-ssm {
      group-range multicast-prefix;
    }
    ldp-p2mp;
    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 8.5.

**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.

**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-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.

**Related Documentation**
- Configuring PIM-SSM GRE Selective Provider Tunnels

## 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.

**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

## 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.

**Related Documentation**

- Configuring NLRI Parameters for an MBGP MVPN

## 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**
- Configuring NLRI Parameters for an MBGP MVPN
- BGP Configuration Guidelines chapter in the *Routing Protocols Configuration Guide*

## 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.

**Related Documentation**

- Limiting Routes to Be Advertised by an MVPN VRF Instance

## 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 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**

**create-new-ucast-tunnel**—A new unicast tunnel to the destination that 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. A template must be specified when and only when **create-new-ucast-tunnel** is included in the configuration..

***template-name***—Name of the point-to-point LSP used for the new unicast tunnel.

**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 389
- Configuring Routing Instances for an MBGP MVPN
- create-new-ucast-tunnel on page 814
- mpls-internet-multicast on page 826

## 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.

**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 482
- Example: Configuring MBGP MVPN Extranets on page 442

## label-switched-path-template

**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 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*]

**Release Information**

Statement introduced in Junos OS Release 8.5.

**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 389
- Configuring Point-to-Multipoint LSPs for an MBGP MVPN
- Flooding Unknown Traffic Using Point-to-Multipoint 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 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*]

**Release Information** Statement introduced in Junos OS Release 11.2.

**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 384

## 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 389
- ingress-replication on page 822

## 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 482
- Example: Configuring MBGP MVPN Extranets on page 442

## mvpn

**Syntax**

```
mvpn {
  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* routing-instances *routing-instance-name* protocols],
[edit routing-instances *routing-instance-name* protocols]

**Release Information** Statement introduced in Junos OS Release 8.4.

**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.

**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-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.

The remaining statements are explained separately.

**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

## p2mp (Protocols LDP)

**Syntax** p2mp;

**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 384
- Point-to-Multipoint LSPs Overview

## pim-asm

**Syntax**

```
pim-asm {
    group-address 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.

**Required Privilege Level**

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

**Related Documentation**

- Configuring PIM Provider Tunnels for an MBGP MVPN

## 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.

**Related Documentation**

- Configuring PIM-SSM GRE Selective Provider Tunnels

## 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 482
- Example: Configuring MBGP MVPN Extranets on page 442

## provider-tunnel

**Syntax**

```
provider-tunnel {
  ingress-replication {
    create-new-ucast-tunnel;
      label-switched-path-template {
        (default-template | lsp-template-name);
      }
    }
  ldp-p2mp;
  pim-asm {
    group-address address;
  }
  mdt {
    data-mdt-reuse;
    group-range multicast-prefix;
    threshold {
      group group-address {
        source source-address {
          rate threshold-rate;
        }
      }
      tunnel-limit limit;
    }
  }
  pim-ssm {
    group-address 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.

**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
- Configuring Point-to-Multipoint LSPs for an MBGP MVPN
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 527

## 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.

**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

## 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.

**Related Documentation**
- Configuring Shared-Tree Data Distribution Across Provider Cores for Providers of MBGP MVPNs

## 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.

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

## 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.

**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 (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.

**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

## 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

## 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 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]

**Release Information** Statement introduced in Junos OS Release 8.5.

**Description** Specify the name of the static point-to-multipoint LSP used for an MBGP MVPN. Use this statement to specify the static LSP for both inclusive and selective point-to-multipoint LSPs.

**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

## 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-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
- Configuring Intra-AS Selective Provider Tunnels

## 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* 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**

Trace traffic flowing through an 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
- **error**—Error conditions
- **general**—General events
- **nlri**—Multicast VPN advertisements received or sent by means of the BGP
- **normal**—Normal events
- **policy**—Policy processing
- **route**—Routing information
- **state**—State transitions
- **task**—Routing protocol task processing
- **timer**—Routing protocol timer processing

- **topology**—Multicast VPN topology changes caused by reconfiguration or advertisements received from other provider edge (PE) routers using BGP

***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

## 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 837
- wildcard-source on page 850

## 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 482
- Example: Configuring MBGP MVPN Extranets on page 442

## 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 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

## wildcard-group-inet

**Syntax**

```
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;
  }
}
```

**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.

**Description**

Configure a wildcard group matching any group IPv4 address.

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**

- wildcard-group-inet6 on page 849
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 440
- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 435
- Configuring a Selective Provider Tunnel Using Wildcards on page 440

## wildcard-group-inet6

**Syntax**

```
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 selective],
[edit routing-instances *routing-instance-name* provider-tunnel selective]

**Release Information**

Statement introduced in Junos OS Release 10.0.

**Description**

Configure a wildcard group matching any group IPv6 address.

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**

- wildcard-group-inet on page 848
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 440
- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 435
- Configuring a Selective Provider Tunnel Using Wildcards on page 440

## wildcard-source

**Syntax**

```
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;
  }
```

**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.

**Description**

Configure a selective provider tunnel for a shared tree using a wildcard source.

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**

- wildcard-group-inet on page 848
- wildcard-group-inet6 on page 849
- Example: Configuring Selective Provider Tunnels Using Wildcards on page 440
- Understanding Wildcards to Configure Selective Point-to-Multipoint LSPs for an MBGP MVPN on page 435
- Configuring a Selective Provider Tunnel Using Wildcards on page 440

CHAPTER 24

# Draft Rosen MVPN Configuration Statements

## autodiscovery

**Syntax**

```
autodiscovery {
  inet-mdt;
}
```

**Hierarchy Level**

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mvpn],
[edit routing-instances *routing-instance-name* protocols pim mvpn],

**Release Information**

Statement introduced in Junos OS Release 9.4.

**Description**

For draft-rosen 7, enable the PE routers in the VPN to discover one another automatically.

**Options**

The 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 507

## autodiscovery-only

**Syntax**

```
autodiscovery-only {
  intra-as {
    inclusive;
  }
}
```

**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 9.4.

**Description** Enable the draft-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 507

## data-mdt-reuse

**Syntax**

```
data-mdt-reuse;
```

**Hierarchy Level** [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mdt],
[edit routing-instances *routing-instance-name* protocols pim mdt]

**Release Information** Statement introduced in Junos OS Release 10.0.

**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 542

## 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.

**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**

- Example: Configuring Draft Rosen Interoperability and a VPN Tunnel Source on page 517
- interface-name on page 857
- Understanding MVPN Interoperation with Other Vendors on page 517

## 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 mdt threshold],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold],
[edit routing-instances *routing-instance-name* provider-tunnel mdt threshold]

**Release Information**

Statement introduced before Junos OS Release 7.4.

**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.

**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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## 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 pim-asm],
[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel pim-asm family (inet | 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 | inet6],
[edit routing-instances *routing-instance-name* provider-tunnel pim-asm],
[edit routing-instances *routing-instance-name* provider-tunnel pim-asm family (inet | inet6],
[edit routing-instances *routing-instance-name* provider-tunnel pim-ssm],
[edit routing-instances *routing-instance-name* provider-tunnel pim-ssm family (inet | inet6]

**Release Information** Statement introduced in Junos OS Release 9.4.

**Description** Configure the PIM-SM or PIM-SSM provider tunnel group address.

**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 507

## group-range (Data MDTs)

**Syntax** group-range *multicast-prefix*;

**Hierarchy Level** [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* provider-tunnel mdt],
[edit routing-instances *routing-instance-name* protocols pim mdt],
[edit routing-instances *routing-instance-name* provider-tunnel mdt]

**Release Information** Statement introduced before Junos OS Release 7.4.

**Description** Establish the group range to use for data MDTs created in this VRF instance. 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.

**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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## inclusive

**Syntax** inclusive;

**Hierarchy Level** [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn autodiscovery-onlyintra-as],
[edit routing-instances *routing-instance-name* protocols mvpn autodiscovery-only intra-as],

**Release Information** Statement introduced in Junos OS Release 9.4.

**Description** For draft-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 507

## inet-mdt (Autodiscovery)

**Syntax** inet-mdt;

**Hierarchy Level** [edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols pim mvpn autodiscovery],
[edit routing-instances *routing-instance-name* protocols pim mvpn autodiscovery],

**Release Information** Statement introduced in Junos OS Release 9.4.

**Description** For draft-rosen 7, configure the PE router in a VPN to use an SSM multicast distribution tree (MDT) subsequent address family identifier (SAFI) NLRI for autodiscovery of other PE routers.

**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 507

## 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**
- Understanding MVPN Interoperation with Other Vendors on page 517
- Example: Configuring Draft Rosen Interoperability and a VPN Tunnel Source on page 517

## intra-as

**Syntax**

```
intra-as {
    inclusive;
}
```

**Hierarchy Level**

[edit logical-systems *logical-system-name* routing-instances *routing-instance-name* protocols mvpn autodiscovery-only],
[edit routing-instances *routing-instance-name* protocols mvpn autodiscovery-only,]

**Release Information**

Statement introduced in Junos OS Release 9.4.

**Description**

For draft-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 507

## 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],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* provider-tunnel]

**Release Information**

Statement introduced before Junos OS Release 7.4.

**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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## 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 507

## mvpn (Draft-Rosen MVPN)

**Syntax**

```
mvpn {
  autodiscovery {
    inet-mdt;
  }
}
```

**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.

**Description**

Configure the control plane to be used for PE routers in the VPN to discover one another automatically.

**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 507

## pim-ssm (Provider Tunnel)

**Syntax**

```
pim-ssm {
  group-address 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 9.4.

**Description**

Configure the PIM-SSM provider tunnel.

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 Source-Specific Multicast for Draft-Rosen Multicast VPNs on page 507

## 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 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 mdt threshold group *group-address* source *source-address*]

**Release Information** Statement introduced before Junos OS Release 7.4.

**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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## 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 507
- Configuring an MPLS-Based Layer 2 VPN (CLI Procedure)

## 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 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 mdt threshold group *group-address*]

**Release Information**

Statement introduced before Junos OS Release 7.4.

**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.

**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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## 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 mdt],
[edit routing-instances *routing-instance-name* protocols pim mdt],
[edit routing-instances *routing-instance-name* provider-tunnel mdt]

**Release Information**

Statement introduced before Junos OS Release 7.4.

**Description**

Establish a threshold to trigger the automatic creation of a data MDT.

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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## 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 mdt],
[edit routing-instances *routing-instance-name* protocols pim mdt],
[edit routing-instances *routing-instance-name* provider-tunnel mdt]

**Release Information** Statement introduced before Junos OS Release 7.4.

**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 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

## 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**

- mvpn (NG-MVPN) on page 860
- Example: Configuring a PIM-SSM Provider Tunnel for an MBGP MVPN on page 412

CHAPTER 25

# AMT Configuration Statements

## 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 553

## 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 555

## 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.

**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 555

## 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.

**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 553

## 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 553

## 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.

**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 553

## 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.

**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 553

## 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 555

## 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.

**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 553

## 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 553

## 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 555

## 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. The query response interval must be less than the query interval.

**Options** ***seconds***—Time to wait to receive a response to a host query message.

**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 555

## 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.

**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 555

## relay (AMT Protocol)

**Syntax**

```
relay {
  accounting;
  family {
    inet {
      anycast-prefix ip-prefix/<prefix-length>;
      local-address ip-address;
    }
  }
  secret-key-timeout minutes;
  tunnel-limit number;
}
```

**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.

**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 553

## 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 555

## 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 553

## 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 555

## 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 553

## 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 553

## 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 555

CHAPTER 26

# Session Announcement Protocol Configuration Statements

## 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 563

## 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 563

## 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 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**

- Configuring the Session Announcement Protocol on page 563
- listen on page 888

CHAPTER 27

# MSDP Configuration Statements

## 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.

**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 575

## 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.

**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 router 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 568

## 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.

**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 575

## 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.

**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 575

## 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.

**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 583

## 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.

**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 568
- import on page 900

## group

**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.

**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.

**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 568

## 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 565
- keep-alive (Protocols MSDP) on page 901
- sa-hold-time (Protocols MSDP) on page 911

## 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.

**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 568
- export on page 897

## 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 565
- hold-time (Protocols MSDP) on page 899
- sa-hold-time (Protocols MSDP) on page 911

## local-address

**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.

**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 568

## 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 575
- log-warning
- maximum on page 905

## 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 575
- log-interval
- maximum on page 905

## maximum

**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.

**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 575
- threshold on page 913

## 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.

**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 575

## 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 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 in a Routing Instance on page 568

## 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.

**Description**

Define an MSDP peering relationship. 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. 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.

**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 568

## 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.

**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 568

## 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 565
- hold-time (Protocols MSDP) on page 899
- keep-alive (Protocols MSDP) on page 901

## source

**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.

**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 575

## threshold

**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.

**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 575
- maximum on page 905

## 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.

**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 581

CHAPTER 28

# PGM Configuration Statements

## pgm

**Syntax**

```
pgm {
  traceoptions {
    flag flag <flag-modifier>;
  }
}
```

**Hierarchy Level**

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

**Release Information**

Statement introduced before Junos OS Release 7.4.

**Description**

Configure PGM globally and set tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

The remaining statement is explained separately.

**Default**

The default PGM trace options are inherited from the routing protocol **traceoptions** statement included 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**

- PGM Configuration Guidelines on page 591

## traceoptions (Protocols PGM)

**Syntax**

```
traceoptions {
  flag flag <flag-modifier>;
}
```

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols pgm],
[edit protocols pgm]

**Release Information** Statement introduced before Junos OS Release 7.4.

**Description** Configure PGM tracing options.

To specify more than one tracing operation, include multiple **flag** statements.

**Default** The default PGM trace options are those inherited from the routing protocol **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**.

**flag *flag***—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

**PGM Tracing Flags**

- **all**—Trace all PGM packets.
- **init**—Trace all PGM initialization events.
- **packets**—Trace all PGM packet processing.
- **parser**—Trace all PGM parser processing.
- **route-socket**—Trace all PGM route-socket events.
- **show**—Trace all PGM **show** command servicing.
- **state**—Trace all PGM state transitions.

**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 the following modifiers:

- **detail**—Detailed trace information
- **receive**—Packets being received
- **send**—Packets being transmitted

**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**

- PGM Configuration Guidelines on page 591

CHAPTER 29

# DVMRP Configuration Statements

## disable (Protocols DVMRP)x

**Syntax** disable;

**Hierarchy Level** [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*]

**Release Information** Statement introduced before Junos OS Release 7.4.

**Description** Explicitly disable DVMRP on the system or 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**
- Example: Configuring DVMRP to Announce Unicast Routes on page 598

## 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** Statement introduced before Junos OS Release 7.4.

**Description** Enable DVMRP on the router.

**Default** DVMRP is disabled on the router.

**Options** The 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 DVMRP on page 594

## export (Protocols DVMRP)

**Syntax** export [ *policy-names* ];

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

**Release Information** 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 924
- Example: Configuring DVMRP to Announce Unicast Routes on page 598

## 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** 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 594

## import (Protocols DVMRP)

**Syntax** import [ *policy-names* ];

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

**Release Information** 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 923
- Example: Configuring DVMRP to Announce Unicast Routes on page 598

## 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** 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.

**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 594

## 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** 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 594

## 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** 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 598

## rib-group (Protocols DVMRP)

**Syntax** rib-group *group-name*;

**Hierarchy Level** [edit logical-systems *logical-system-name* protocols dvmrp],
[edit protocols dvmrp]

**Release Information** 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 594

## 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** 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 601

PART 3

# Administration

CHAPTER 30

# PIM Operational Commands

## clear pim join

**Syntax**

clear pim join
<*group-address*>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

**Syntax (EX Series Switch and the QFX Series)**

clear pim join
<*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.

**Description**

Clear the Protocol Independent Multicast (PIM) join and prune states.

**Options**

**none**—Clear the PIM join and prune states for all groups, family addresses, and instances.

***group-address***—(Optional) Clear the PIM join and prune states for a group address.

**inet | inet6**—(Optional) Clear the PIM join and prune states for IPv4 or IPv6 family addresses, respectively.

**instance *instance-name***—(Optional) Clear the join and prune 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** 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 948

**List of Sample Output**

clear pim join on page 934

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear pim join

```
user@host> clear pim join
```

## clear pim join-distribution

**Syntax**

clear pim join-distribution
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>

**Release Information**

Command introduced in Junos OS Release 10.0.

**Description**

Redistribute the Protocol Independent Multicast (PIM) join states.

You can find out if there are multiple paths available for a source (for example, an RP) with the output of the **show pim source** command.

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**

**none**—Redistribute the PIM join states for the default master instance.

**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 957
- show pim join on page 948
- join-load-balance on page 644 in the *Multicast Protocols Configuration Guide*

**List of Sample Output**

clear pim join-distribution on page 936

**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

```
user@host> clear pim join-distribution
```

## clear pim register

**Syntax**

clear pim register
<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.

**Description**

Clear Protocol Independent Multicast (PIM) register message counters.

**Options**

none—Clear PIM register message counters for all family addresses, instances, and interfaces.

**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 971

**List of Sample Output**

clear pim register on page 938

**Output Fields**

When you enter this command, you are provided feedback on the status of your request.

## Sample Output

clear pim register

```
user@host> clear pim register
```

## clear pim statistics

**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.

**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 971

**List of Sample Output**

clear pim statistics on page 940

**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
...
```

## request pim multicast-tunnel rebalance

**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 945
- Load Balancing Multicast Tunnel Interfaces Among Available PICs on page 503 in the *Junos Multicast Protocols Configuration Guide*

**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 pim bootstrap

**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.

**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 944
show pim bootstrap instance on page 944

**Output Fields**

Table 16 on page 943 describes the output fields for the **show pim bootstrap** command. Output fields are listed in the approximate order in which they appear.

### Table 16: 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. |
| **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**. |

Table 16: show pim bootstrap Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **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
feco:1:1:1:1:0:aff:785c 34 feco:1:1:1:1:0: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

**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.

**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 947

**Output Fields**

Table 17 on page 945 describes the output fields for the **show pim interfaces** command. Output fields are listed in the approximate order in which they appear.

### Table 17: show pim interfaces Output Fields

| Field Name | Field Description |
|---|---|
| **Instance** | Name of the routing instance. |
| **Name** | Interface name. |
| **State** | State of the interface. The state also is displayed in the **show interfaces** command. |

Table 17: show pim interfaces Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **Mode** | PIM mode running on the interface:<br>• **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.<br>• **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.<br>• **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.)<br>• **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. (Not supported on QFX Series.)<br>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:<br>• **Active**—Bidirectional mode is enabled on the interface and on all PIM neighbors.<br>• **DR**—Designated router.<br>• **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.<br>• **NotDR**—Not the designated router.<br>• **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

**Syntax**

show pim join
<brief | detail | extensive | summary>
<inet | inet6>
<instance *instance-name*>
<logical-system (all | *logical-system-name*)>
<*range*>

**Syntax (EX Series Switch and the QFX Series)**

show pim join
<brief | detail | extensive | summary>
<inet | inet6>
<instance *instance-name*>
<*range*>

**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.

**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.

**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***.

**Required Privilege Level**

view

**Related Documentation**

- clear pim join on page 934

**List of Sample Output**

show pim join summary on page 952
show pim join (PIM Sparse Mode) on page 952

**Output Fields** Table 18 on page 949 describes the output fields for the **show pim join** command. Output fields are listed in the approximate order in which they appear.

Table 18: 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:<br>• ***** (wildcard value)<br>• ***ipv4-address***<br>• ***ipv6-address*** | **brief detail extensive** none |
| **RP** | Rendezvous point for the PIM group. | **brief detail extensive** none |

Table 18: show pim join Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Flags** | PIM flags:<br>• **bidirectional**—Bidirectional mode entry.<br>• **dense**—Dense mode entry.<br>• **rptree**—Entry is on the rendezvous point tree.<br>• **sparse**—Sparse mode entry.<br>• **spt**—Entry is on the shortest-path tree for the source.<br>• **wildcard**—Entry is on the shared tree. | **brief detail extensive** none |
| **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).<br>For bidirectional PIM, **RP Link** means that the interface is directly connected to a subnet that contains a phantom RP address. | **brief detail extensive** none |
| **Upstream neighbor** | Information about the upstream neighbor: **Direct**, **Local**, **Unknown**, or a specific IP address.<br>For bidirectional PIM, **Direct** means that the interface is directly connected to a subnet that contains a phantom RP address. | **extensive** |
| **Upstream state** | Information about the upstream interface:<br>• **Join to RP**—Sending a join to the rendezvous point.<br>• **Join to Source**—Sending a join to the source.<br>• **Local RP**—Sending neither join messages nor prune messages toward the RP, because this router is the rendezvous point.<br>• **Local Source**—Sending neither join messages nor prune messages toward the source, because the source is locally attached to this routing device.<br>• **Prune to RP**—Sending a prune to the rendezvous point.<br>• **Prune to Source**—Sending a prune to the source.<br>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. | **extensive** |

Table 18: show pim join Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Downstream neighbors** | Information about downstream interfaces:<br>• **Interface**—Interface name for the downstream neighbor.<br>NOTE: A pseudo PIM-SM interface appears for all IGMP-only interfaces.<br>• **Interface address**—Address of the downstream neighbor.<br>• **State**—Information about the downstream neighbor: **join** or **prune**.<br>• **Flags**—PIM join flags: **R (RPtree)**, **S (Sparse)**, **W (Wildcard)**, or **zero**.<br>• **Uptime**—Time since the downstream interface joined the group.<br>• **Time since last Join**—Time since the last join message was received from the downstream interface.<br>• **Time since last Prune**—Time since the last prune message was received from the downstream interface. | **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** |
| **Bidirectional accepting interfaces** | Interfaces on the router that forward bidirectional PIM traffic.<br>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: 239.1.1.1
    Source: *
    RP: 10.255.14.144
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 239.1.1.1
    Source: 10.255.14.144
    Flags: sparse,spt
    Upstream interface: Local

Group: 239.1.1.1
    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: 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

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)

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
```

```
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)

Instance: PIM.master Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

### 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: 235.1.1.2
    Source: *
    RP: 10.10.47.100
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 235.1.1.2
    Source: 192.168.195.74
    Flags: sparse,spt
    Upstream interface: at-0/3/1.0

Group: 235.1.1.2
    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 detail

```
user@host> show pim join detail
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    Source: *
    RP: 10.255.14.144
    Flags: sparse,rptree,wildcard
    Upstream interface: Local

Group: 239.1.1.1
    Source: 10.255.14.144
    Flags: sparse,spt
    Upstream interface: Local

Group: 239.1.1.1
    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

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
```

### extensive (PIM Sparse Mode)

```
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 239.1.1.1
    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: 239.1.1.1
    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: 239.1.1.1
    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

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

(Bidirectional PIM)

```
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
    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: 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
    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: 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
    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

```
user@host> show pim join extensive
Instance: PIM.master Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

a Directly Connected Phantom RP)

```
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
    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: 235.1.1.2
    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: 235.1.1.2
    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: 235.1.1.2
    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 neighbors

**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.

**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 960
show pim neighbors brief on page 960
show pim neighbors instance on page 960
show pim neighbors detail on page 960
show pim neighbors detail (With BFD) on page 961

**Output Fields**

Table 19 on page 958 describes the output fields for the **show pim neighbors** command. Output fields are listed in the approximate order in which they appear.

### Table 19: 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:<br>• **B**—Bidirectional Capable.<br>• **H**—Hello Option Holdtime.<br>• **G**—Generation Identifier.<br>• **P**—Hello Option DR Priority.<br>• **L**—Hello Option LAN Prune Delay. | **brief** none |
| **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 |
| **Address** | Address of the neighboring PIM router. | **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 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** |

Table 19: show pim neighbors Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Join Suppression supported** | Neighbor is capable of join suppression. | **detail** |
| **Rx Join** | Information about joins received from the neighbor.<br>• **Group**—Group addresses in the join message.<br>• **Source**—Address of the source in the join message.<br>• **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

Interface           IP V Mode        Option      Uptime Neighbor addr
so-1/0/0.0           4 2             HPLG      00:07:10 10.111.10.2
```

### show pim neighbors brief

The output for the **show pim neighbors brief** command is identical to that for the **show pim neighbors** command. For sample output, see show pim neighbors on page 960.

### 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

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: ge-0/0/1.0

    Address: 10.10.1.1, IPv4, PIM v2, Mode: SparseDense, sg Join Count: 0, tsg
Join Count: 2
        Hello Option Holdtime: 65535 seconds
        Hello Option DR Priority: 1
        Hello Option Generation ID: 2053759302
        Hello Option Bi-Directional PIM supported
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                      Join Suppression supported

    Address: 10.10.1.2, IPv4, PIM v2, sg Join Count: 0, tsg Join Count: 2
        BFD: Disabled
        Hello Option Holdtime: 105 seconds 93 remaining
        Hello Option DR Priority: 1
        Hello Option Generation ID: 1734018161
        Hello Option Bi-Directional PIM supported
        Hello Option LAN Prune Delay: delay 500 ms override 2000 ms
                                      Join Suppression 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 rps

**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.

**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 72

**List of Sample Output**

show pim rps on page 965
show pim rps brief on page 965
show pim rps <group-address> (Bidirectional PIM) on page 965
show pim rps <group-address> (PIM Dense Mode) on page 965

show pim rps <group-address> (SSM Range Without asm-override-ssm Configured) on page 965
show pim rps <group-address> (SSM Range With asm-override-ssm Configured and a Sparse-Mode RP) on page 966
show pim rps <group-address> (SSM Range With asm-override-ssm Configured and a Bidirectional RP) on page 966
show pim rps instance on page 967
show pim rps extensive (PIM Sparse Mode) on page 967
show pim rps extensive (Bidirectional PIM) on page 967
show pim rps extensive (PIM Anycast RP in Use) on page 967

**Output Fields** Table 20 on page 963 describes the output fields for the **show pim rps** command. Output fields are listed in the approximate order in which they appear.

### Table 20: 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:<br>• **auto-rp**—Address of the RP known through the Auto-RP protocol.<br>• **bootstrap**—Address of the RP known through the bootstrap router protocol (BSR).<br>• **embedded**—Address of the RP known through an embedded RP (IPv6).<br>• **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** |
| **Mode** | The PIM mode of the RP: bidirectional or sparse.<br>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** |

Table 20: show pim rps Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Device Index** | Index value of the order in which Junos OS finds and initializes the interface.<br><br>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.<br><br>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.<br><br>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**<br><br>*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** |
| **Register State for RP** | Current register state for each group:<br>• **Group**—Multicast group address.<br>• **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:<br>• **First Hop**—PIM-designated routing device that sent the Register message (the source address in the IP header).<br>• **RP Address**—RP to which the Register message was sent (the destination address in the IP header).<br>• **State**:<br>On the designated router:<br>◦ **Send**—Sending Register messages.<br>◦ **Probe**—Sent a null register. If a Register-Stop message does not arrive in 5 seconds, the designated router resumes sending Register messages.<br>◦ **Suppress**—Received a Register-Stop message. The designated router is waiting for the timer to resume before changing to **Probe** state.<br>• On the RP:<br>◦ **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** |

Table 20: show pim rps Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Anycast-PIM Register State** | If anycast RP is configured, the current register state for each group:<br>• **Group**—Multicast group address.<br>• **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.<br>• **Origin**—How the information was obtained:<br>  • **DIRECT**—From a local attachment<br>  • **MSDP**—From the Multicast Source Discovery Protocol (MSDP)<br>  • **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
10.10.1.3       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
```

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 965.

show pim rps <group-address> (Bidirectional PIM)

```
user@host> show pim rps 224.1.1.1
Instance: PIM.master

224.1.0.0/16
        11.4.12.75 (Bidirectional)

RP selected: 11.4.12.75
```

show pim rps <group-address> (PIM Dense Mode)

```
user@host> show pim rps 224.1.1.1
Instance: PIM.master

Dense Mode active for group 224.1.1.1
```

show pim rps <group-address> (SSM Range Without

```
user@host> show pim rps 224.1.1.1
Instance: PIM.master

Source-specific Mode (SSM) active for group 224.1.1.1
```

asm-override-ssm Configured)

show pim rps <group-address> (SSM Range With asm-override-ssm Configured and a Sparse-Mode RP)

```
user@host> show pim rps 224.1.1.1
Instance: PIM.master

Source-specific Mode (SSM) active with Sparse Mode ASM override for group 224.1.1.1

224.1.0.0/16
        11.4.12.75

RP selected: 11.4.12.75
```

show pim rps <group-address> (SSM Range With asm-override-ssm

```
user@host> show pim rps 224.1.1.1
Instance: PIM.master

Source-specific Mode (SSM) active with Sparse Mode ASM override for group 224.1.1.1
```

Configured and a Bidirectional RP)

```
224.1.0.0/16
        11.4.12.75 (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      1 224.0.0.0/4

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:
    224.0.0.0/4, 36s remaining
Active groups using RP:
    225.1.1.1

    total 1 groups active

Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
225.1.1.1       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:
        224.1.3.0/24
        225.1.3.0/24

RP: 10.10.13.2
Learned via: static configuration
Mode: Bidirectional
Time Active: 01:58:07
Holdtime: 150
Group Ranges:
        224.1.1.0/24
        225.1.1.0/24

user@host> show pim rps extensive
```

### show pim rps extensive (PIM Anycast RP in Use)

```
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:
    224.0.0.0/4
Active groups using RP:
    224.10.10.10

    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
224.1.1.1                 10.10.95.2                            DIRECT
224.1.1.2                 10.10.95.2                            DIRECT
224.10.10.10              10.10.70.1                            MSDP
224.10.10.11              10.10.70.1                            MSDP
224.20.20.1               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

**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.

**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 970
show pim source brief on page 970
show pim source detail on page 970

**Output Fields**

Table 21 on page 970 describes the output fields for the **show pim source** command. Output fields are listed in the approximate order in which they appear.

Table 21: 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 interface** | RPF interface toward the source address. |
| **Upstream Neighbor** | Address of the RPF neighbor used to reach the source address. |

## 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 970.

### 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:228.0.0.0
        239.1.1.1
        239.1.1.1

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:239.1.1.1

Instance: PIM.master Family: INET6
```

## show pim statistics

**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.

**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 939

**List of Sample Output**

show pim statistics on page 979
show pim statistics inet interface <interface-name> on page 980
show pim statistics inet6 interface <interface-name> on page 980
show pim statistics instance <instance-name> on page 981
show pim statistics interface <interface-name> on page 983

**Output Fields**

Table 22 on page 972 describes the output fields for the **show pim statistics** command. Output fields are listed in the approximate order in which they appear.

### Table 22: show pim statistics Output Fields

| Field Name | Field Description |
|---|---|
| **Instance** | Name of the routing instance.<br><br>This field only appears if you specify an interface, for example:<br><br>• **inet interface *interface-name***<br>• **inet6 interface *interface-name***<br>• **interface *interface-name*** |
| **Family** | Output is for IPv4 or IPv6 PIM statistics. **INET** indicates IPv4 statistics, and **INET6** indicates IPv6 statistics.<br><br>This field only appears if you specify an interface, for example:<br><br>• **inet interface *interface-name***<br>• **inet6 interface *interface-name***<br>• **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 22: 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.<br><br>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 22: 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 router 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 router 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 router has no state. |
| **Rx Join/Prune on upstream if** | Number of join and prune messages received on the interface used to reach the upstream router, toward the RP. |
| **Rx Join/Prune for invalid group** | Number of join or prune messages received for invalid multicast group addresses. |

Table 22: 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 router, 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 router has no state for the data type. |
| **Rx RP no state** | Number of PIM control packets received for which the router 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 router does not have an encapsulation interface. |
| **No route upstream** | Number of PIM control packets received when the router does not have a unicast route to the the interface used to reach the upstream router, 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 router has an RP mismatch. |

Table 22: 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 router 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.<br><br>The following receive events trigger extraction of an IPv6 embedded RP address on the router:<br><br>• Multicast Listener Discovery (MLD) report for an embedded RP multicast group address<br>• PIM join message with an embedded RP multicast group address<br>• Static embedded RP multicast group address associated with an interface<br>• Packets sent to an embedded RP multicast group address received on the DR<br><br>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 22: 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 22: 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.<br><br>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.<br><br>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. |

## 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
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
```

## 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

```
user@host> show pim statistics inet6 interface ge-0/3/0.0
Instance: PIM.master Family: INET6
```

<interface-name>

```
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
```

## 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
```

CHAPTER 31

# Multicast Routing Options Operational Commands

## clear multicast forwarding-cache

**Syntax**
clear multicast forwarding-cache
<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.

**Options**
**none**—(Same as **logical-system all**) Clear multicast forwarding cache entries.

**inet**—(Optional) Clear multicast forwarding cache entries for IPv4 family addresses.

**inet6**—(Optional) Clear multicast forwarding cache entries for IPv6 family addresses.

**interface *interface-name***—(Optional) Clear multicast forwarding cache entries 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 forwarding-cache statistics on page 989

**List of Sample Output**
clear multicast forwarding-cache on page 986

**Output Fields**
When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear multicast forwarding-cache

```
user@host> clear multicast forwarding-cache
```

## 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 988

**Output Fields**
Table 23 on page 987 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 23: 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 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.

**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 986

**List of Sample Output**
show multicast forwarding-cache statistics on page 990
show multicast forwarding-cache statistics instance on page 990

**Output Fields**
Table 24 on page 989 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 24: 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**. |
| **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. |
| **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. |

Table 24: show multicast forwarding-cache statistics Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **Warning Threshold** | Threshold at which a warning message is logged (percentage of the suppress threshold). |
| **Currently Used Entries** | Number of currently used multicast forwarding cache entries. |

## Sample Output

### show multicast forwarding-cache statistics

```
user@host> show multicast forwarding-cache statistics
Instance: master Family: INET
Suppress Threshold                    100
Reuse Value                            80
Warning Threshold                      90
Currently Used Entries                101

Instance: master Family: INET6
Suppress Threshold                     50
Reuse Value                            50
Warning Threshold                      80
Currently Used Entries                  3
```

### show multicast forwarding-cache statistics instance

```
user@host> show multicast forwarding-cache statistics instance VPN-A
Instance: VPN-A Family: ALL
Suppress Threshold                     20
Reuse Value                            16
Warning Threshold                      50
Currently Used Entries                 17
```

## show multicast flow-map

**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.

**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 992
show multicast flow-map detail on page 992

**Output Fields**
Table 25 on page 991 describes the output fields for the **show multicast flow-map** command. Output fields are listed in the approximate order in which they appear.

### Table 25: 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 |
| **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:  11.11.11.11
    Redundant Sources:  11.11.11.12
    Redundant Sources:  11.11.11.13
```

## show multicast interface

**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.

**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 994

**Output Fields**
Table 26 on page 993 describes the output fields for the **show multicast interface** command. Output fields are listed in the approximate order in which they appear.

### Table 26: 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. |
| **Mapped bandwidth deduction (bps)** | Amount of bandwidth, in bits per second, used by any flows that are mapped to the interface.<br><br>NOTE: Adding the mapped bandwidth deduction value to the local bandwidth deduction value results in the total deduction value for the interface.<br><br>This field does not appear in the output when the no QoS adjustment feature is disabled. |

Table 26: show multicast interface Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **Local bandwidth deduction (bps)** | Amount of bandwidth, in bits per second, used by any mapped flows that are traversing the interface.<br><br>NOTE: Adding the mapped bandwidth deduction value to the local bandwidth deduction value results in the total deduction value for the interface.<br><br>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**).<br><br>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.<br><br>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.<br><br>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.<br><br>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 route

**Syntax**

show multicast route
<brief | detail | extensive | summary>
<active | all | inactive>
<group *group*>
<inet | inet6>
<instance *instance name*>
<logical-system (all | *logical-system-name*)>
<*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.

**Description**

Display the entries in the IP multicast forwarding table. You can display similar information with the **show route table inet.1** command.

**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.

***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

**List of Sample Output**

show multicast route on page 998
show multicast route (Bidirectional PIM) on page 998
show multicast route brief on page 998
show multicast route detail on page 999
show multicast route extensive (Bidirectional PIM) on page 999
show multicast route instance <instance-name> on page 1000
show multicast route summary on page 1000

**Output Fields** Table 27 on page 996 describes the output fields for the **show multicast route** command. Output fields are listed in the approximate order in which they appear.

### Table 27: 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.<br><br>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 |
| **Downstream interface list** | List of interface names to which the packet with this source prefix is forwarded. | 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**.<br><br>NOTE: On QFX 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** |

Table 27: show multicast route Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Incoming interface list ID** | For bidirectional PIM, incoming interface list identifier.<br><br>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** |
| **Upstream protocol** | Protocol running on the interface on which the packet with this source prefix is expected to arrive. | **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

### show multicast route

```
user@host> show multicast route
Family: INET

Group: 228.0.0.0
    Source: 10.255.14.144/32
    Upstream interface: local
    Downstream interface list:
        so-1/0/0.0

Group: 239.1.1.1
    Source: 10.255.14.144/32
    Upstream interface: local
    Downstream interface list:
        so-1/0/0.0

Group: 239.1.1.1
    Source: 10.255.70.15/32
    Upstream interface: so-1/0/0.0
    Downstream interface list:
        mt-1/1/0.49152

Family: INET6
```

### show multicast route (Bidirectional PIM)

```
user@host> show multicast route
Family: INET

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

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

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

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
Family: INET6
```

The output for the **show multicast route brief** command is identical to that for the **show**

show multicast route brief

**multicast route** command. For sample output, see show multicast route on page 998 or show multicast route (Bidirectional PIM) on page 998.

show multicast route detail

```
user@host> show multicast route detail
Family: INET

Group: 228.0.0.0
    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: 239.1.1.1
    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: 239.1.1.1
    Source: 10.255.70.15/32
    Upstream interface: so-1/0/0.0
    Downstream interface list:
        mt-1/1/0.49152
    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: 224.1.1.0/24
    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: 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
    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 instance <instance-name>

```
user@host> show multicast route instance v1 extensive
Instance: v1 Family: INET

Group: 224.1.1.1
    Source: (null)/0
    Upstream interface: fe-1/3/0.111
    Downstream interface list:
        lt-0/3/0.42 lt-0/3/0.46 lt-0/3/0.43
    Number of outgoing interfaces: 3

Group: 224.1.1.2
    Source: (null)/0
    Upstream interface: fe-1/3/0.111
    Downstream interface list:
        lt-0/3/0.42 lt-0/3/0.46 lt-0/3/0.43
    Number of outgoing interfaces: 3

Group: 224.1.1.3
    Source: (null)/0
    Upstream interface: fe-1/3/0.111
    Downstream interface list:
        lt-0/3/0.42 lt-0/3/0.46 lt-0/3/0.43
    Number of outgoing interfaces: 3

Instance: v1 Family: INET6
```

### 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 rpf

**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.

**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 1003
show multicast rpf inet6 on page 1003
show multicast rpf prefix on page 1004
show multicast rpf summary on page 1004

**Output Fields** Table 28 on page 1002 describes the output fields for the **show multicast rpf** command. Output fields are listed in the approximate order in which they appear.

### Table 28: 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.<br><br>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.<br><br>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

127.0.0.1/32
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: fe80::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: fe80::2a0:a5ff:fe28:2e8c

::192.168.195.76/126
Protocol: Direct
Interface: fe-2/2/0.0

::192.168.195.77/128
Protocol: Local

fe80::/64
Protocol: Direct
Interface: so-1/1/1.0

fe80::290:69ff:fe0c:993a/128
Protocol: Local

fe80::2a0:a5ff:fe12:84f/128
Protocol: Direct
Interface: lo0.0

ff02::2/128
Protocol: PIM

ff02::d/128
Protocol: PIM
```

### show multicast rpf prefix

```
user@host> show multicast rpf ff02::/16

Multicast RPF table: inet6.0, 13 entries

ff02::2/128
    Protocol: PIM

ff02::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

**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.

**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 1006
show multicast scope inet on page 1006
show multicast scope inet6 on page 1006

**Output Fields**

Table 29 on page 1005 describes the output fields for the **show multicast scope** command. Output fields are listed in the approximate order in which they appear.

### Table 29: show multicast scope Output Fields

| Field Name | Field Description |
|---|---|
| **Scope name** | Name of the multicast scope. |
| **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
232-net              232.232.0.0/16     fe-0/0/0.1                    0
local                239.255.0.0/16     fe-0/0/0.1                    0
local                ff05::/16          fe-0/0/0.1                    0
larry                ff05::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
232-net              232.232.0.0/16     fe-0/0/0.1                    0
local                239.255.0.0/16     fe-0/0/0.1                    0
```

### show multicast scope inet6

```
user@host> show multicast scope inet6
                                                                Resolve
Scope name           Group Prefix       Interface               Rejects
local                ff05::/16          fe-0/0/0.1                    0
larry                ff05::1234/128     fe-0/0/0.1                    0
```

## show multicast sessions

**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.

**Description**

Display information about announced IP multicast sessions.

**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 1008
show multicast sessions regular-expression detail on page 1008

**Output Fields**

Table 30 on page 1007 describes the output fields for the **show multicast sessions** command. Output fields are listed in the approximate order in which they appear.

### Table 30: 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**

```
user@host> show multicast sessions "NASA TV" detail
SDP Version: 0  Originated by: -@128.223.83.33
 Session: NASA TV (MPEG-1)
```

regular-expression detail

```
 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: 224.2.231.45 ttl 127
Attribute: quality:8
Attribute: framerate:30
Attribute: rtpmap:96 WBIH/90000
Attribute: rtpmap:97 MP4V-ES/90000
Attribute: x-iptv-svr:video 128.223.91.191 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 policy

**Syntax**

show policy
<logical-system (all | *logical-system-name*)>
<*policy-name*>

**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.

**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.

**Required Privilege Level**

view

**Related Documentation**

- show policy damping

**List of Sample Output**

show policy on page 1011
show policy policy-name on page 1011
show policy (Multicast Scoping) on page 1011

**Output Fields**

Table 31 on page 1010 lists the output fields for the **show policy** command. Output fields are listed in the approximate order in which they appear.

### Table 31: show policy Output Fields

| Field Name | Field Description |
|---|---|
| ***policy-name*** | Name of the policy listed. |
| ***term*** | Policy term listed. |
| **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
all_routes
```

**show policy policy-name**

```
user@host> show policy test-statics
Policy test-statics:
    from
         3.0.0.0/8  accept
         3.1.0.0/16  accept
    then reject
```

**show policy (Multicast Scoping)**

```
user@host> show policy test-statics
Policy test-statics:
    from
         multicast-scoping == 8
```

CHAPTER 32

# IGMP Operational Commands

## clear igmp statistics

**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.

**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 1015

**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
```

## show igmp group

**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.

**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

**List of Sample Output**
show igmp group (Include Mode) on page 1018
show igmp group (Exclude Mode) on page 1018
show igmp group brief on page 1018
show igmp group detail on page 1018

**Output Fields**
Table 32 on page 1016 describes the output fields for the **show igmp group** command. Output fields are listed in the approximate order in which they appear.

### Table 32: 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 |

Table 32: show igmp 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** |
| **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:<br>• **Dynamic**—Host reported the membership.<br>• **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: 232.1.1.1
        Group mode: Include
        Source: 10.0.0.2
        Last reported by: 10.9.5.2
        Timeout:      24 Type: Dynamic
    Group: 232.1.1.1
        Group mode: Include
        Source: 10.0.0.3
        Last reported by: 10.9.5.2
        Timeout:      24 Type: Dynamic
    Group: 232.1.1.1
        Group mode: Include
        Source: 10.0.0.4
        Last reported by: 10.9.5.2
        Timeout:      24 Type: Dynamic
    Group: 232.1.1.2
        Group mode: Include
        Source: 10.0.0.4
        Last reported by: 10.9.5.2
        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: 224.0.0.2
        Source: 0.0.0.0
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: 224.0.0.22
        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: 224.0.0.2
        Source: 0.0.0.0
        Last reported by: Local
        Timeout:       0 Type: Dynamic
    Group: 224.0.0.22
        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: 232.1.1.1
        Group mode: Include
        Source: 10.0.0.2
        Source timeout: 12
        Last reported by: 10.9.5.2
        Group timeout:       0 Type: Dynamic
    Group: 232.1.1.1
        Group mode: Include
        Source: 10.0.0.3
        Source timeout: 12
        Last reported by: 10.9.5.2
        Group timeout:       0 Type: Dynamic
    Group: 232.1.1.1
        Group mode: Include
        Source: 10.0.0.4
        Source timeout: 12
        Last reported by: 10.9.5.2
        Group timeout:       0 Type: Dynamic
    Group: 232.1.1.2
        Group mode: Include
        Source: 10.0.0.4
        Source timeout: 12
        Last reported by: 10.9.5.2
        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: 224.0.0.2
        Group mode: Exclude
        Source: 0.0.0.0
        Source timeout: 0
        Last reported by: Local
        Group timeout:       0 Type: Dynamic
    Group: 224.0.0.22
        Group mode: Exclude
        Source: 0.0.0.0
        Source timeout: 0
        Last reported by: Local
        Group timeout:       0 Type: Dynamic
```

## show igmp interface

**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.

**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

**List of Sample Output**

show igmp interface on page 1023
show igmp interface brief on page 1023
show igmp interface detail on page 1023
show igmp interface <interface-name> on page 1023

**Output Fields**

Table 33 on page 1020 describes the output fields for the **show igmp interface** command. Output fields are listed in the approximate order in which they appear.

### Table 33: 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 |

Table 33: show igmp interface Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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.<br><br>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:<br><br>• **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.<br>• **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:<br><br>• **On**—Indicates that the router can accept IGMP reports from subnetworks that are not associated with its interfaces.<br>• **Off**—Indicates that the router can accept IGMP reports only from subnetworks that are associated with its interfaces. | All levels |
| **Passive** | State of the passive mode option:<br><br>• **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.<br>• **Off**—Indicates that the router can run IGMP on the interface and send or receive control traffic such as IGMP reports, queries, and leaves.<br><br>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:<br><br>• **send-general-query**—The interface sends general queries.<br>• **send-group-query**—The interface sends group-specific and group-source-specific queries.<br>• **allow-receive**—The interface receives control traffic. | All levels |
| **OIF map** | Name of the OIF map (if configured) associated with the interface. | All levels |

Table 33: show igmp interface Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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:<br>• **IGMP Query Interval**—Interval (in seconds) at which this router sends membership queries when it is the querier.<br>• **IGMP Query Response Interval**—Time (in seconds) that the router waits for a report in response to a general query.<br>• **IGMP Last Member Query Interval**—Time (in seconds) that the router waits for a report in response to a group-specific query.<br>• **IGMP Robustness Count**—Number of times the router retries a query. | All levels |
| **Derived Parameters** | Derived information:<br>• **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.<br>• **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: 10.111.30.1
    State:         Up Timeout:    None Version:  2 Groups:      4
    SSM Map Policy: ssm-policy-A
Interface: so-1/0/0.0
    Querier: 10.111.10.1
    State:         Up Timeout:    None Version:  2 Groups:      2
    SSM Map Policy: ssm-policy-B
Interface: so-1/0/1.0
    Querier: 10.111.20.1
    State:         Up Timeout:    None Version:  2 Groups:      4
    SSM Map Policy: ssm-policy-C
Immediate Leave: On
Promiscuous Mode: Off

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 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 1023.

### 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 1023.

### show igmp interface <interface-name>

```
user@host# show igmp interface ge-3/2/0.0
Interface: ge-3/2/0.0
  Querier: 20.1.1.1
  State: Up Timeout:    None Version:  3 Groups:      1
  Group limit: 8
  Group threshold: 60
  Group log-interval: 10
  Immediate leave:  Off
  Promiscuous mode:  Off
```

## show multicast pim-to-igmp-proxy

**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.

**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 299

**List of Sample Output**

show multicast pim-to-igmp-proxy on page 1025
show multicast pim-to-igmp-proxy instance on page 1025

**Output Fields**

Table 34 on page 1024 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 34: 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
```

CHAPTER 33

# MLD Operational Commands

## clear mld membership

**Syntax** clear mld membership
<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** **none**—Clear all MLD memberships.

**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 1030

**List of Sample Output** clear mld membership on page 1028

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear mld membership

```
user@host> clear mld membership
```

## 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 1039

**List of Sample Output**
clear mld statistics on page 1029

**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
```

## 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 1028

**List of Sample Output**

show mld group (Include Mode) on page 1032
show mld group (Exclude Mode) on page 1032
show mld group brief on page 1032
show mld group detail (Include Mode) on page 1033
show mld group detail (Exclude Mode) on page 1033

**Output Fields**

Table 35 on page 1030 describes the output fields for the **show mld group** command. Output fields are listed in the approximate order in which they appear.

### Table 35: 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 35: 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:<br>• **Dynamic**—Host reported the membership.<br>• **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 1032

show mld group (Exclude Mode) on page 1032.

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 Multicast 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 1028

**List of Sample Output**
show mld interface on page 1038
show mld interface brief on page 1038
show mld interface detail on page 1038
show mld interface <interface-name> on page 1038

**Output Fields**
Table 36 on page 1035 describes the output fields for the **show mld interface** command. Output fields are listed in the approximate order in which they appear.

### Table 36: 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 |
| **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 |

Table 36: show mld interface Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Groups** | Number of groups on the interface. | All levels |
| **Passive** | State of the passive mode option:<br>• **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.<br>• **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.<br>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:<br>• **send-general-query**—The interface sends general queries.<br>• **send-group-query**—The interface sends group-specific and group-source-specific queries.<br>• **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.<br>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:<br>• **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.<br>• **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 36: show mld interface Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Configured Parameters** | Information configured by the user.<br>• **MLD Query Interval (.1 secs)**—Interval at which this router sends membership queries when it is the querier.<br>• **MLD Query Response Interval (.1 secs)**—Time that the router waits for a report in response to a general query.<br>• **MLD Last Member Query Interval (.1 secs)**—Time that the router waits for a report in response to a group-specific query.<br>• **MLD Robustness Count**—Number of times the router retries a query. | All levels |
| **Derived Parameters** | Derived information.<br>• **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.<br>• **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

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 1038.

### 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 1038.

### show mld interface <interface-name>

```
user@host# show mld interface ge-3/2/0.0
Interface: ge-3/2/0.0
  Querier: 20.1.1.1
  State: Up Timeout:    None Version:  3 Groups:      1
  Group limit: 8
  Group threshold: 60
  Group log-interval: 10
  Immediate leave:  Off
  Promiscuous mode:  Off
```

## 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 1029

**List of Sample Output**

show mld statistics on page 1041
show mld statistics interface on page 1041

**Output Fields**

Table 37 on page 1039 describes the output fields for the **show mld statistics** command. Output fields are listed in the approximate order in which they appear.

### Table 37: 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 37: show mld statistics Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **MLD Message type** | Summary of MLD statistics.<br>• **Listener Query (v1/v2)**—Number of membership queries sent and received.<br>• **Listener Report (v1)**—Number of version 1 membership reports sent and received.<br>• **Listener Done (v1/v2)**—Number of Listener Done messages sent and received.<br>• **Listener Report (v2)**—Number of version 2 membership reports sent and received.<br>• **Other Unknown types**—Number of unknown message types received.<br>• **MLD v2 source required for SSM**—Number of MLD version 2 messages received that contained no source.<br>• **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.<br>• **Bad Length**—Number of messages received with length errors so severe that further classification could not occur.<br>• **Bad Checksum**—Number of messages received with an invalid IP checksum. No further classification was performed.<br>• **Bad Receive If**—Number of messages received on an interface not enabled for MLD.<br>• **Rx non-local**—Number of messages received from nonlocal senders.<br>• **Timed out**—Number of groups that timed out as a result of not receiving an explicit leave message.<br>• **Rejected Report**—Number of reports dropped because of the MLD group policy.<br>• **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 multicast pim-to-mld-proxy

**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 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 1043
show multicast pim-to-mld-proxy instance on page 1043

**Output Fields**

Table 38 on page 1042 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 38: 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**. |
| ***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
```

CHAPTER 34

# IGMP Snooping Operational Commands

## clear igmp snooping membership

**Syntax**

clear igmp snooping membership
<group | source *address*>
<instance *instance-name*>
<interface *interface-name*>
<learning-domain *learning-domain-name*>
<vlan-id *vlan-identifier*>

**Release Information**

Command introduced in Junos OS Release 8.5.

**Description**

Clear IP IGMP snooping membership information.

**Options**

**none**—Clear IGMP snooping membership for all supported address families on all interfaces.

**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.

**vlan-id *vlan-identifier***—(Optional) Perform this operation on a particular VLAN.

**Required Privilege Level**

clear

**Related Documentation**

- show igmp snooping membership on page 1051

**List of Sample Output**

clear igmp snooping membership on page 1046

**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 (all | *logical-system-name*)>

**Release Information**
Command introduced in Junos OS Release 8.5.

**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 (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 snooping statistics on page 1055

**List of Sample Output**
clear igmp snooping statistics on page 1047

**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
```

## show igmp snooping interface

**Syntax** show igmp snooping interface *interface-name*
<brief | detail>
<bridge-domain *bridge-domain-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) Display the specified level of output.

**bridge-domain *bridge-domain-name***—(Optional) Display information about a particular bridge domain.

**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 1051
- show igmp snooping statistics on page 1055

**List of Sample Output** show igmp snooping interface on page 1050
show igmp snooping interface (Group Limit Configured) on page 1050

**Output Fields** Table 39 on page 1048 lists the output fields for the **show igmp snooping interface** command. Output fields are listed in the approximate order in which they appear.

### Table 39: show igmp snooping interface Output Fields

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Routing-instance** | Routing instance for IGMP snooping. | All levels |
| **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. | **detail** |
| **IGMP Query Response Interval** | Time (in seconds) that the router waits for a response to a general query. | **detail** |

Table 39: show igmp snooping interface Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **IGMP Last Member Query Interval** | Time (in seconds) that the router waits for a report in response to a group-specific query. | **detail** |
| **IGMP Robustness Count** | Number of times the router retries a query. | **detail** |
| **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. | none |
| **State** | State of the interface: **Up** or **Down**. | none |
| **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. | none |
| **IGMP Other Querier Present Timeout** | Time that the router waits for the IGMP querier to send a query. | none |

## Sample Output

### show igmp snooping interface

```
user@host> show igmp snooping interface
Instance: bridge-domain bar

Learning-Domain: default
Interface: ge-0/1/0.200
    State:         Up Groups:      0
    Immediate leave: Off
    Router interface: yes
Interface: ge-0/1/2.200
    State:         Up Groups:      2
    Immediate leave: On
    Router interface: no
Interface: ge-0/1/3.200
    State:         Up Groups:      1
    Immediate leave: Off
    Router interface: no

Configured Parameters:
IGMP Query Interval: 130.0
IGMP Query Response Interval: 15.0
IGMP Last Member Query Interval: 2.0
IGMP Robustness Count: 3

Derived Parameters:
IGMP Membership Timeout: 405.0
IGMP Other Querier Present Timeout: 397.500
```

## Sample Output

### 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>
<bridge-domain *bridge-domain-name*>
<group *group-name*>
<virtual-switch *virtual-switch-name*>
<vlan-id *vlan-identifier*>

**Release Information**
Command introduced in Junos OS Release 8.5.

**Description**
Display IGMP snooping membership information.

**Options**
**none**—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.

**group *group-name*** —(Optional) Display information about this group address.

**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 1048
- show igmp snooping statistics on page 1055
- clear igmp snooping membership on page 1046

**List of Sample Output**
show igmp snooping membership on page 1053
show igmp snooping membership (Exclude Mode) on page 1053
show igmp snooping membership interface ge-0/1/2.200 on page 1053
show igmp snooping membership vlan-id 1 on page 1054

**Output Fields**
Table 40 on page 1051 lists the output fields for the **show igmp snooping membership** command. Output fields are listed in the approximate order in which they appear.

### Table 40: show igmp snooping membership Output Fields

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Instance** | Routing instance for IGMP snooping. | All levels |
| **Learning Domain** | Learning domain for snooping. | All levels |

Table 40: show igmp snooping membership Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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:<br>• **Dynamic**—Group membership was learned by the IGMP protocol.<br>• **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: 225.10.10.1
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 100.6.85.2
        Group timeout:     173 Type: Dynamic
```

### 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: 225.10.10.1
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 100.6.85.2
        Group timeout:     173 Type: Dynamic
```

### 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: 225.1.1.1
        Source: 0.0.0.0
        Timeout:     391 Type: Static
    Group: 232.1.1.1
        Source: 192.168.1.1
        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: 225.10.10.1
        Group mode: Exclude
        Source: 0.0.0.0
        Last reported by: 100.6.85.2
        Group timeout:     209 Type: Dynamic
```

## show igmp snooping statistics

**Syntax**
show igmp snooping statistics
<brief | detail>
<bridge-domain *bridge-domain-name*>
<virtual-switch *virtual-switch-name*>
<vlan-id *vlan-identifier*>

**Release Information**
Command introduced in Junos OS Release 8.5.

**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.

**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 1048
- show igmp snooping membership on page 1051
- clear igmp snooping statistics on page 1047

**List of Sample Output**
show igmp snooping statistics on page 1057

**Output Fields**
Table 41 on page 1055 lists the output fields for the **show igmp snooping statistics** command. Output fields are listed in the approximate order in which they appear.

Table 41: 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 |

Table 41: show igmp snooping statistics Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **IGMP Message type** | Summary of IGMP statistics:<br>• **Membership Query**—Number of membership queries sent and received.<br>• **V1 Membership Report**—Number of version 1 membership reports sent and received.<br>• **DVMRP**—Number of DVMRP messages sent or received.<br>• **PIM V1**—Number of PIM version 1 messages sent or received.<br>• **Cisco Trace**—Number of Cisco trace messages sent or received.<br>• **V2 Membership Report**—Number of version 2 membership reports sent or received.<br>• **Group Leave**—Number of group leave messages sent or received.<br>• **Domain Wide Report**—Number of domain-wide reports sent or received.<br>• **V3 Membership Report**—Number of version 3 membership reports sent or received.<br>• **Other Unknown types**—Number of unknown message types received.<br>• **IGMP v3 unsupported type**—Number of messages received with unknown and unsupported IGMP version 3 message types.<br>• **IGMP v3 source required for SSM**—Number of IGMP version 3 messages received that contained no source.<br>• **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.<br>• **Bad Length**—Number of messages received with length errors so severe that further classification could not occur.<br>• **Bad Checksum**—Number of messages received with a bad IP checksum. No further classification was performed.<br>• **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
```

CHAPTER 35

# Multicast Snooping Operational Commands

## clear multicast snooping statistics

**Syntax** clear multicast snooping statistics
<instance *instance-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

**Release Information** Command introduced in Junos OS Release 8.5.

**Description** Clear IP multicast snooping statistics.

**Options**

**none**—Clear multicast snooping statistics for all supported address families on all interfaces.

**instance *instance-name***—(Optional) Clear multicast snooping statistics for the specified instance.

**interface *interface-name***—(Optional) Clear multicast snooping 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 snooping statistics on page 1064

**List of Sample Output** clear multicast snooping statistics on page 1060

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear multicast snooping statistics

```
user@host> clear multicast snooping statistics
```

## show multicast snooping route

**Syntax**

show multicast snooping route
<brief | detail | extensive>
<active | all | inactive>
<bridge-domain *bridge-domain-name*>
<group *group*>
<instance *instance-name*>
<mesh-group *mesh-group-name*>
<*regular-expression*>
<source-prefix *source-prefix*>

**Release Information**

Command introduced in Junos OS Release 8.5.

**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.

**brief | detail | extensive**—(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 snooping table.

**bridge-domain *bridge-domain***—(Optional) Display the entries for a particular bridge domain.

**group *group***—(Optional) Display the entries for a particular group.

**instance *instance-name***—(Optional) Display the entries for a multicast instance.

**mesh-group *mesh-group-name***—(Optional) Display the entries for a particular mesh group.

***regular-expression***—(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.

**Required Privilege Level**

view

**List of Sample Output**

show multicast snooping route bridge-domain on page 1063
show multicast snooping route instance vs on page 1063

**Output Fields**

Table 42 on page 1062 describes the output fields for the **show multicast snooping route** command. Output fields are listed in the approximate order in which they appear.

### Table 42: 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. | 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** |
| **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 statistics

**Syntax**

show multicast snooping statistics
<instance *instance-name*>
<interface *interface-name*>
<logical-system (all | *logical-system-name*)>

**Release Information**

Command introduced in Junos OS Release 8.5.

**Description**

Display IP multicast snooping statistics.

**Options**

**none**—Display multicast snooping statistics for all supported address families for all routing instances.

**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 snooping 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 snooping statistics on page 1060

**List of Sample Output**

show multicast snooping statistics on page 1066

**Output Fields**

Table 43 on page 1064 describes the output fields for the **show multicast snooping statistics** command. Output fields are listed in the approximate order in which they appear.

### Table 43: show multicast snooping statistics Output Fields

| Field Name | Field Description |
|---|---|
| **Routing-instance** | Name of the routing instance. (Displayed when multicast is configured within a routing instance.) |
| **Family** | Protocol family for which multicast statistics are displayed: **INET** or **INET6**. |
| **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. |

Table 43: show multicast snooping statistics Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **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 snooping statistics** command was issued. |
| **Out Kbytes** | Total accumulated outgoing packets (in KB) since the last time the **clear multicast snooping 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 snooping statistics** command was issued. |
| **Out packets** | Total number of outgoing packets since the last time the **clear multicast snooping statistics** command was issued. |

## Sample Output

### show multicast snooping statistics

```
user@host> show multicast snooping statistics
Routing-instance: foo
 Family: INET
 Interface: fe-0/0/2.200
     Routing protocol:          PIM   Mismatch error:              0
     Mismatch:                    0   Mismatch no route:           0
     Kernel resolve:             22   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

Routing-instance: bar
 Family: INET
 Interface: fe-0/1/2.200
     Routing protocol:          PIM   Mismatch error:              0
     Mismatch:                    0   Mismatch no route:           0
     Kernel resolve:             22   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 route table

**Syntax**
show route table *routing-table-name*
<brief | detail | extensive | terse>
<logical-system (all | *logical-system-name*)>

**Syntax (EX 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.

**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 name begins 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 inetflow detail on page 1076

**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 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)

10.255.245.12:1:4.0.0.0/8 (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

```
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
```

Target Filtering Is Configured)

```
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 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 111.222.5.254 via fxp0.0
1.0.0.1/32          *[Direct/0] 00:51:58
                     > via at-5/3/0.0
1.0.0.2/32          *[Local/0] 00:51:58
                       Local
12.12.12.21/32      *[Local/0] 00:51:57
                       Reject
13.13.13.13/32      *[Direct/0] 00:51:58
                     > via t3-5/2/1.0
13.13.13.14/32      *[Local/0] 00:51:58
                       Local
13.13.13.21/32      *[Local/0] 00:51:58
                       Local
13.13.13.22/32      *[Direct/0] 00:33:59
                     > via t3-5/2/0.0
127.0.0.1/32         [Direct/0] 00:51:58
                     > via lo0.0
111.222.5.0/24      *[Direct/0] 00:51:58
                     > via fxp0.0
111.222.5.81/32     *[Local/0] 00:51:58
                       Local
```

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: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 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: 65002
                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 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: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:45:09, metric 1
                      Receive
1                  *[MPLS/0] 00:45:09, metric 1
                      Receive
2                  *[MPLS/0] 00:45:09, metric 1
                      Receive
100000             *[L2VPN/7] 00:43:04
                    > via so-0/1/0.1, Pop
100001             *[L2VPN/7] 00:43:03
                    > via so-0/1/0.2, Pop       Offset: 4
100002             *[LDP/9] 00:43:22, metric 1
                      via so-0/1/2.0, Pop
                    > via so-0/1/3.0, Pop
100002(S=0)        *[LDP/9] 00:43:22, metric 1
                      via so-0/1/2.0, Pop
                    > via so-0/1/3.0, Pop
100003             *[LDP/9] 00:43:22, metric 1
                    > via so-0/1/2.0, Swap 100002
                      via so-0/1/3.0, Swap 100002
100004             *[LDP/9] 00:43:16, metric 1
                      via so-0/1/2.0, Swap 100049
                    > via so-0/1/3.0, Swap 100049
so-0/1/0.1         *[L2VPN/7] 00:43:04
                    > via so-0/1/2.0, Push 100001, Push 100049(top)
                      via so-0/1/3.0, Push 100001, Push 100049(top)
so-0/1/0.2         *[L2VPN/7] 00:43:03
                      via so-0/1/2.0, Push 100000, Push 100049(top) Offset: -4
                    > via so-0/1/3.0, Push 100000, Push 100049(top) Offset: -4
```

### 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 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300352(S=0)        *[RSVP/7/1] 00:08:00, metric 1
                    > to 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300384             *[RSVP/7/2] 00:05:20, metric 1
                    > to 8.64.1.106 via ge-1/0/0.0, Pop
300384(S=0)        *[RSVP/7/2] 00:05:20, metric 1
                    > to 8.64.1.106 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

1.1.1.11:1000:1:1/96 (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

1:10.255.2.202:65535:10.255.2.202/432
```

```
                    *[BGP/170] 00:02:37, localpref 100, from 10.255.2.202
                      AS path: I
                     > via so-0/1/3.0
1:10.255.2.203:65535:10.255.2.203/432
                    *[BGP/170] 00:02:37, localpref 100, from 10.255.2.203
                      AS path: I
                     > via so-0/1/0.0
1:10.255.2.204:65535:10.255.2.204/432
                    *[MVPN/70] 00:57:23, metric2 1
                      Indirect
5:10.255.2.202:65535:128:::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
6:10.255.2.203:65535:65000:128:::10.12.53.12:128:ffff::1/432
                    *[PIM/105] 00:02:37
                      Multicast (IPv6)
7:10.255.2.202:65535:65000:128:::192.168.90.2:128:ffff::1/432
                    *[MVPN/70] 00:02:37, metric2 1
                      Indirect
```

### 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 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: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 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: 65002
                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@PE1> **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

1.1.1.2:100:1.1.1.2/96 AD
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
1.1.1.4:100:1.1.1.4/96 AD
                   *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
1.1.1.2:100:1:0/96 MH
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
1.1.1.4:100:1:0/96 MH
                   *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.2:1.1.1.4/176
                   *[VPLS/7] 1d 03:11:02, metric2 1
                    >    via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.4:1.1.1.2/176
                   *[LDP/9] 1d 03:11:02
                         Discard
```

CHAPTER 36

# MBGP MVPNs Operational Commands

## show bgp group

**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.
**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**

show bgp group on page 1085
show bgp group brief on page 1085
show bgp group detail on page 1085
show bgp group rtf detail on page 1086
show bgp group summary on page 1086

**Output Fields** Table 44 on page 1081 describes the output fields for the **show bgp group** command. Output fields are listed in the approximate order in which they appear.

### Table 44: 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 |
| **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 |
| **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 |
| **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 |

Table 44: show bgp group Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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.<br>• If a peer is not established, the field shows the state of the peer session: **Active**, **Connect**, or **Idle**.<br>• 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:<br>◦ 8 active routes, 10 received routes, 10 accepted routes, and 2 damped routes from a BGP peer appear in the **inet.0** routing table.<br>◦ 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:<br>• 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 44: show bgp group Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Table inet.*number*** | Information about the routing table.<br>• **Received prefixes**—Total number of prefixes from the peer, both active and inactive, that are in the routing table.<br>• **Active prefixes**—Number of prefixes received from the peer that are active in the routing table.<br>• **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.<br>• **Advertised prefixes**—Number of prefixes advertised to a peer.<br>• **Received external prefixes**—Total number of prefixes from the external BGP (EBGP) peers, both active and inactive, that are in the routing table.<br>• **Active external prefixes**—Number of prefixes received from the EBGP peers that are active in the routing table.<br>• **Externals suppressed**—Number of routes received from EBGP peers currently inactive because of damping or other reasons.<br>• **Received internal prefixes**—Total number of prefixes from the IBGP peers, both active and inactive, that are in the routing table.<br>• **Active internal prefixes**—Number of prefixes received from the IBGP peers that are active in the routing table.<br>• **Internals suppressed**—Number of routes received from IBGP peers currently inactive because of damping or other reasons.<br>• **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 44: 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
 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 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
  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 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 1087

**Output Fields** Table 45 on page 1087 lists the output fields for the **show ingress-replication mvpn** command. Output fields are listed in the approximate order in which they appear.

### Table 45: show ingress-replication mvpn

| 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 mpls lsp

**Syntax**

show mpls lsp
<brief | detail | extensive | terse>
<bidirectional | unidirectional>
<bypass>
<count-active-routes>
<defaults>
<descriptions>
<down | up>
<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>
<*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.

**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.

**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.

**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.
- **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.

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

**List of Sample Output**

show mpls lsp defaults on page 1096
show mpls lsp descriptions on page 1096
show mpls lsp detail on page 1096
show mpls lsp extensive on page 1096
show mpls lsp ingress extensive on page 1097

show mpls lsp p2mp on page 1098
show mpls lsp p2mp detail on page 1098
show mpls lsp detail count-active-routes on page 1099
show mpls lsp statistics extensive on page 1099

**Output Fields** Table 46 on page 1090 describes the output fields for the **show mpls lsp** command. Output fields are listed in the approximate order in which they appear.

### Table 46: 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** |

Table 46: show mpls lsp Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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** |
| **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 **Static configured** or dynamic **Dynamic configured**. 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** |
| **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 |

Table 46: show mpls lsp Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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 node 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** |
| **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** |
| **AdjustTimer** | (Ingress LSP) Configured value of the bandwidth adjustment timer, indicating the total amount of time allowed before bandwidth adjustment will take place, in seconds. | **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.<br>***Primary long**<br>**Standby short** | **detail extensive** |

Table 46: show mpls lsp Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **Primary** | (Ingress LSP) Name of the primary path. | **detail extensive** |
| **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 46: 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:<br>• **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.<br>• **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).<br>• **0x03**—Combination of **0x01** and **0x02**.<br>• **0x04**—Bandwidth protection. The downstream routing device has a backup path providing the same bandwidth guarantee as the protected LSP for the protected section.<br>• **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.<br>• **0x09**—Detour is established. Combination of **0x01** and **0x08**.<br>• **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.<br>• **0xb**—Detour is in use. Combination of **0x01**, **0x02**, and **0x08**. | **detail 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** |
| **Labelin** | Incoming label for this LSP. | **brief detail** |
| **Labelout** | Outgoing label for this LSP. | **brief detail** |
| **LSPname** | Name of the LSP. | **brief detail** |

Table 46: show mpls lsp Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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** |
| **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** |
| **MPLS-TE LSP Defaults** | Default settings for MPLS traffic engineered LSPs:<br>• **LSP Holding Priority**—Determines the degree to which an LSP holds on to its session reservation after the LSP has been set up successfully.<br>• **LSP Setup Priority**—Determines whether a new LSP that preempts an existing LSP can be established.<br>• **Hop Limit**—Specifies the maximum number of routers the LSP can traverse (including the ingress and egress).<br>• **Bandwidth**—Specifies the bandwidth in bits per second for the LSP.<br>• **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 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
  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 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 Configured
```

```
  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):
          1.1.1.2 4.4.4.1 5.5.5.2
   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 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: 60.2599Mbps, Bandwidth Adjustment in 0 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 mvpn c-multicast

**Syntax** show mvpn c-multicast
<extensive | summary>
<instance-name *instance-name*>

**Release Information** Command introduced in Junos OS Release 8.4.

**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.

**Required Privilege Level** view

**List of Sample Output**
show mvpn c-multicast on page 1102
show mvpn c-multicast summary on page 1102
show mvpn c-multicast extensive on page 1102

**Output Fields** Table 47 on page 1101 lists the output fields for the **show mvpn c-multicast** command. Output fields are listed in the approximate order in which they appear.

Table 47: 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:<br>• **DS**—Represents (S,G) and is created due to (*,G)<br>• **RM**—Remote VPN route learned from the remote PE router<br>• **St** display blank—SSM group join | **extensive** none |
| **MVPN instance** | Name of the multicast VPN routing instance | **extensive** none |
| **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
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         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
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

### show mvpn c-multicast summary

```
user@host> show mvpn c-multicast summary
MVPN Summary:
Instance: VPN-A
  C-multicast IPv4 route count: 1
Instance: VPN-B
  C-multicast IPv4 route count: 2
```

### 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
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         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
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

## show mvpn instance

**Syntax** show mvpn instance
<extensive | summary>
<instance *instance-name*>

**Release Information** Command introduced in Junos OS Release 8.4.

**Description** Display the multicast VPN routing instance information.

**Options** **extensive | summary**—(Optional) Display the specified level of output.

**instance *instance-name***—(Optional) Display statistics for the specified routing instance.

**Required Privilege Level** view

**List of Sample Output**
show mvpn instance on page 1105
show mvpn instance on page 1105
show mvpn instance summary on page 1105
show mvpn instance extensive on page 1106
show mvpn instance summary (IPv6) on page 1106

**Output Fields** Table 48 on page 1103 lists the output fields for the **show mvpn instance** command. Output fields are listed in the approximate order in which they appear.

### Table 48: 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 |
| **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:<br>• **DS**—Represents (S,G) and is created due to (*,G)<br>• **RM**—Remote VPN route learned from the remote PE router<br>• **St** display blank—SSM group join | **extensive** none |

Table 48: show mvpn instance Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         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, 239.2.0.0
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

### 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-1
  MVPN Mode : SPT-ONLY
  Provider tunnel: I-P-tnl:LDP-P2MP:10.255.72.162, lsp-id 16777217
  Neighbor                              I-P-tnl
  10.255.72.160                         LDP-P2MP:10.255.72.160, lsp-id 16777217
  10.255.72.166                         RSVP-TE P2MP:10.255.72.166,
13054,10.255.72.166
  10.255.72.168
```

## Sample Output

### show mvpn instance summary

```
user@host> show mvpn instance summary
MVPN Summary:
Instance: VPN-A
  Neighbor count: 2
  C-multicast IPv4 route count: 1
Instance: VPN-B
  Neighbor count: 4
```

```
  C-multicast IPv4 route count: 2
```

## Sample Output

### show mvpn instance extensive

```
user@host> show mvpn instance 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
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                               I-P-tnl
  10.255.14.160                          PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                           PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         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, 239.2.0.0
  Neighbor                               I-P-tnl
  10.255.14.160                          PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                           PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

### 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 1108
show mvpn neighbor extensive on page 1108
show mvpn neighbor extensive on page 1108
show mvpn neighbor instance-name on page 1109
show mvpn neighbor neighbor-address on page 1109
show mvpn neighbor neighbor-address summary on page 1109
show mvpn neighbor neighbor-address extensive on page 1109
show mvpn neighbor neighbor-address instance-name on page 1110

**Output Fields**

Table 49 on page 1107 lists the output fields for the **show mvpn neighbor** command. Output fields are listed in the approximate order in which they appear.

Table 49: 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 |
| **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, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
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, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
```

## 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, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
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, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
```

### 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, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
```

## 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, 239.1.1.1
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, 239.2.0.0
```

## 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

```
user@host> show mvpn neighbor neighbor-address 10.255.70.17 extensive
```

### show mvpn neighbor neighbor-address 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, 239.1.1.1
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, 239.2.0.0
```

## 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, 239.1.1.1
```

## show route forwarding-table

**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 (Routing Matrix)**

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.

**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: **fibre-channel**, **fmembers**, **inet**, **inet6**, **iso**, **mpls**, **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***—(Routing Matrix only) (Optional) On a routing matrix composed of a TX Matrix Plus router and T640 routers configured in the routing matrix, 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 routers configured in the routing matrix, display information for the specified T1600 router (or line-card chassis) connected to the TX Matrix Plus router. Replace ***number*** with a value from 0 through 3.

**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 (default | *logical-system-name/routing-instance-name* | *routing-instance-name*)**—(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 on page 1116
show route forwarding-table detail on page 1116
show route forwarding-table destination extensive (Weights and Balances) on page 1117
show route forwarding-table extensive on page 1118
show route forwarding-table extensive (RPF) on page 1119
show route forwarding-table family mpls on page 1121
show route forwarding-table family vpls on page 1121
show route forwarding-table family vpls extensive on page 1121
show route forwarding-table table default on page 1122
show route forwarding-table table logical-system-name/routing-instance-name on page 1123
show route forwarding-table vpn on page 1124

**Output Fields** Table 50 on page 1113 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 50: 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 |
| **Address family** | Address family (for example, **IP**, **IPv6**, **ISO**, **MPLS**, and **VPLS**). | All levels |
| **Destination** | Destination of the route. | **detail extensive** |

Table 50: 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):<br>• **cloned** (**clon**)—(TCP or multicast only) Cloned route.<br>• **destination** (**dest**)—Remote addresses directly reachable through an interface.<br>• **destination down** (**iddn**)—Destination route for which the interface is unreachable.<br>• **interface cloned** (**ifcl**)—Cloned route for which the interface is unreachable.<br>• **route down** (**ifdn**)—Interface route for which the interface is unreachable.<br>• **ignore** (**ignr**)—Ignore this route.<br>• **interface** (**intf**)—Installed as a result of configuring an interface.<br>• **permanent** (**perm**)—Routes installed by the kernel when the routing table is initialized.<br>• **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:<br>• **none**—No flags are enabled.<br>• **accounting**—Route has accounting enabled.<br>• **cached**—Cache route.<br>• **incoming-iface** ***interface-number***—Check against incoming interface.<br>• **prefix load balance**—Load balancing is enabled for this prefix.<br>• **rt nh decoupled**—Route has been decoupled from the next hop to the destination.<br>• **sent to PFE**—Route has been sent to the Packet Forwarding Engine.<br>• **static**—Static route. | **extensive** |
| **Next hop** | IP address of the next hop to the destination. | **detail extensive** |

Table 50: 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):<br>• **broadcast (bcst)**—Broadcast.<br>• **deny**—Deny.<br>• **discard (dscd)** —Discard.<br>• **hold**—Next hop is waiting to be resolved into a unicast or multicast type.<br>• **indexed (idxd)**—Indexed next hop.<br>• **indirect (indr)**—Indirect next hop.<br>• **local (locl)**—Local address on an interface.<br>• **routed multicast (mcrt)**—Regular multicast next hop.<br>• **multicast (mcst)**—Wire multicast next hop (limited to the LAN).<br>• **multicast discard (mdsc)**—Multicast discard.<br>• **multicast group (mgrp)**—Multicast group member.<br>• **receive (recv)**—Receive.<br>• **reject (rjct)**—Discard. An ICMP unreachable message was sent.<br>• **resolve (rslv)**—Resolving the next hop.<br>• **unicast (ucst)**—Unicast.<br>• **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
1.1.1.0/24         ifdn     0                    rslv   608     1 ge-2/0/1.0
1.1.1.0/32         iddn     0 1.1.1.0            recv   606     1 ge-2/0/1.0
1.1.1.1/32         user     0                    rjct    46     4
1.1.1.1/32         intf     0 1.1.1.1            locl   607     2
1.1.1.1/32         iddn     0 1.1.1.1            locl   607     2
1.1.1.255/32       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

```
user@host> show route forwarding-table destination 3.4.2.1 extensive
Routing table: inet [Index 0]
Internet:
```

(Weights and Balances)

```
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: 4.4.4.4
  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: 0:90:69:8e:b1:1b
  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 15.95.1.2/30;
        }
    }
}

user@host> show route forwarding-table extensive
Routing table: inet [Index 0]
Internet:
...
...
Destination:  15.95.1.3/32
  Route type: destination
  Route reference: 0                    Route interface-index: 67
```

```
Flags: sent to PFE
Nexthop: 15.95.1.3
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 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 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
00:90:69:0c:20:1f/48             <<<<<Remote CE

                   dynm     0                   indr   351     4
                                                Push 800000, Push 100002(top)
so-0/0/0.0
00:90:69:85:b0:1f/48             <<<<<<Local CE

                   dynm     0                   ucst   354     2 fe-0/1/0.0
```

### 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:  10:00:00:01:01:01/48
  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:  10:00:00:01:01:04/48
  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:  10:00:00:01:03:05/48
  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
1.0.0.1/32         user     0                    dscd   561     2
2.0.2.0/24         intf     0                    rslv   771     1 ge-1/2/0.3
2.0.2.0/32         dest     0 2.0.2.0            recv   769     1 ge-1/2/0.3
2.0.2.1/32         intf     0 2.0.2.1            locl   770     2
2.0.2.1/32         dest     0 2.0.2.1            locl   770     2
2.0.2.2/32         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
2.0.2.255/32       dest     0 2.0.2.255          bcst   768     1 ge-1/2/0.3
224.0.0.0/4        perm     1                    mdsc   562     1
```

```
224.0.0.1/32       perm     0 224.0.0.1          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
      224.0.0.0/4        perm     2                   mdsc     5     3
      224.0.0.1/32       perm     0 224.0.0.1         mcst     1     8
      224.0.0.5/32       user     1 224.0.0.5         mcst     1     8
      255.255.255.255/32 perm     0                   bcst     2     3
```

## show route label

**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

**List of Sample Output**

show route label on page 1126
show route label detail on page 1126
show route label extensive on page 1126
show route label terse on page 1126

**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

```
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 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 1126.

### 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 table

**Syntax**
show route table *routing-table-name*
<brief | detail | extensive | terse>
<logical-system (all | *logical-system-name*)>

**Syntax (EX 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.

**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 name begins 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 inetflow detail on page 1136

**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 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)

10.255.245.12:1:4.0.0.0/8 (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

```
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
```

Target Filtering Is Configured)

```
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 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 111.222.5.254 via fxp0.0
1.0.0.1/32         *[Direct/0] 00:51:58
                    > via at-5/3/0.0
1.0.0.2/32         *[Local/0] 00:51:58
                      Local
12.12.12.21/32     *[Local/0] 00:51:57
                      Reject
13.13.13.13/32     *[Direct/0] 00:51:58
                    > via t3-5/2/1.0
13.13.13.14/32     *[Local/0] 00:51:58
                      Local
13.13.13.21/32     *[Local/0] 00:51:58
                      Local
13.13.13.22/32     *[Direct/0] 00:33:59
                    > via t3-5/2/0.0
127.0.0.1/32        [Direct/0] 00:51:58
                    > via lo0.0
111.222.5.0/24     *[Direct/0] 00:51:58
                    > via fxp0.0
111.222.5.81/32    *[Local/0] 00:51:58
                      Local
```

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: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 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: 65002
                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 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: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:45:09, metric 1
                      Receive
1                  *[MPLS/0] 00:45:09, metric 1
                      Receive
2                  *[MPLS/0] 00:45:09, metric 1
                      Receive
100000             *[L2VPN/7] 00:43:04
                    > via so-0/1/0.1, Pop
100001             *[L2VPN/7] 00:43:03
                    > via so-0/1/0.2, Pop       Offset: 4
100002             *[LDP/9] 00:43:22, metric 1
                      via so-0/1/2.0, Pop
                    > via so-0/1/3.0, Pop
100002(S=0)        *[LDP/9] 00:43:22, metric 1
                      via so-0/1/2.0, Pop
                    > via so-0/1/3.0, Pop
100003             *[LDP/9] 00:43:22, metric 1
                    > via so-0/1/2.0, Swap 100002
                      via so-0/1/3.0, Swap 100002
100004             *[LDP/9] 00:43:16, metric 1
                      via so-0/1/2.0, Swap 100049
                    > via so-0/1/3.0, Swap 100049
so-0/1/0.1         *[L2VPN/7] 00:43:04
                    > via so-0/1/2.0, Push 100001, Push 100049(top)
                      via so-0/1/3.0, Push 100001, Push 100049(top)
so-0/1/0.2         *[L2VPN/7] 00:43:03
                      via so-0/1/2.0, Push 100000, Push 100049(top) Offset: -4
                    > via so-0/1/3.0, Push 100000, Push 100049(top) Offset: -4
```

### 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 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300352(S=0)        *[RSVP/7/1] 00:08:00, metric 1
                    > to 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300384             *[RSVP/7/2] 00:05:20, metric 1
                    > to 8.64.1.106 via ge-1/0/0.0, Pop
300384(S=0)        *[RSVP/7/2] 00:05:20, metric 1
                    > to 8.64.1.106 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

1.1.1.11:1000:1:1/96 (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

1:10.255.2.202:65535:10.255.2.202/432
```

```
                    *[BGP/170] 00:02:37, localpref 100, from 10.255.2.202
                      AS path: I
                     > via so-0/1/3.0
1:10.255.2.203:65535:10.255.2.203/432
                    *[BGP/170] 00:02:37, localpref 100, from 10.255.2.203
                      AS path: I
                     > via so-0/1/0.0
1:10.255.2.204:65535:10.255.2.204/432
                    *[MVPN/70] 00:57:23, metric2 1
                      Indirect
5:10.255.2.202:65535:128:::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
6:10.255.2.203:65535:65000:128:::10.12.53.12:128:ffff::1/432
                    *[PIM/105] 00:02:37
                      Multicast (IPv6)
7:10.255.2.202:65535:65000:128:::192.168.90.2:128:ffff::1/432
                    *[MVPN/70] 00:02:37, metric2 1
                      Indirect
```

### 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 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: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 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: 65002
                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@PE1> 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

1.1.1.2:100:1.1.1.2/96 AD
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
1.1.1.4:100:1.1.1.4/96 AD
                   *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
1.1.1.2:100:1:0/96 MH
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
1.1.1.4:100:1:0/96 MH
                   *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.2:1.1.1.4/176
                   *[VPLS/7] 1d 03:11:02, metric2 1
                    >    via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.4:1.1.1.2/176
                   *[LDP/9] 1d 03:11:02
                         Discard
```

CHAPTER 37

# Draft Rosen MVPN Operational Commands

## show pim mdt

**Syntax**

show pim mdt instance *instance-name*
<brief | detail | extensive>
<incoming | outgoing>
<logical-system (all | logical-system-name)>
<*range*>

**Release Information**

Command introduced before Junos OS Release 7.4.

**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.

**incoming | outgoing**—(Optional) Display incoming or outgoing multicast data tunnels, respectively.

**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 instance on page 1142
show pim mdt instance detail on page 1142
show pim mdt instance extensive on page 1142
show pim mdt instance incoming on page 1142
show pim mdt instance outgoing on page 1143
show pim mdt instance (SSM Mode) on page 1143

**Output Fields**

Table 51 on page 1140 describes the output fields for the **show pim mdt** command. Output fields are listed in the approximate order in which they appear.

### Table 51: 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 |

Table 51: show pim mdt Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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 instance

```
user@host> show pim mdt instance VPN-A
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

Instance: PIM.VPN-A
Tunnel direction: Incoming
Default group address: 239.1.1.1
Default tunnel interface: mt-1/1/0.49152
```

### 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-1/1/0.49152
```

### 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-1/1/0.49152
```

### 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-1/1/0.49152
```

### 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-1/3/0.49153

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-1/3/0.49153
```

## 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 525
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

**List of Sample Output** show pim mdt data-mdt-joins on page 1145

**Output Fields** Table 52 on page 1144 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 52: 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. |

Table 52: show pim mdt data-mdt-joins Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **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 525
- Example: Configuring Data MDTs and Provider Tunnels Operating in Source-Specific Multicast Mode on page 527
- Example: Configuring Data MDTs and Provider Tunnels Operating in Any-Source Multicast Mode on page 537

**List of Sample Output** show pim mdt data-mdt-limit on page 1147

**Output Fields** Table 53 on page 1146 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 53: 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 1148

**Output Fields** Table 54 on page 1148 describes the output fields for the **show pim mvpn** command. Output fields are listed in the approximate order in which they appear.

### Table 54: 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).<br><br>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
```

CHAPTER 38

# AMT Operational Commands

## 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 1152

**List of Sample Output**
clear amt statistics on page 1150

**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 1157

**List of Sample Output**

clear amt tunnel on page 1151
clear amt tunnel statistics gateway-address on page 1151

**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
```

## 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 1150
- show amt summary on page 1155
- show amt tunnel on page 1157

**List of Sample Output**
show amt statistics on page 1154

**Output Fields**
Table 55 on page 1152 describes the output fields for the **show amt statistics** command. Output fields are listed in the approximate order in which they appear.

### Table 55: show amt statistics Output Fields

| Field Name | Field Description |
|---|---|
| **AMT receive message count** | Summary of AMT statistics for messages received on all interfaces.<br>• **AMT relay discovery**—Number of AMT relay discovery messages received.<br>• **AMT membership request**—Number of AMT membership request messages received.<br>• **AMT membership update**—Number of AMT membership update messages received. |
| **AMT send message count** | Summary of AMT statistics for messages sent on all interfaces.<br>• **AMT relay advertisement**—Number of AMT relay advertisement messages sent.<br>• **AMT membership query**—Number of AMT membership query messages sent. |

Table 55: 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.<br>• **AMT incomplete packet**—Number of messages received with length errors so severe that further classification could not occur.<br>• **AMT invalid mac**—Number of messages received with an invalid message authentication code (MAC).<br>• **AMT unexpected type**—Number of messages received with an unknown message type specified.<br>• **AMT invalid relay discovery address**—Number of AMT relay discovery messages received with an address other than the configured anycast address.<br>• **AMT invalid membership request address**—Number of AMT membership request messages received with an address other than the configured AMT local address.<br>• **AMT invalid membership update address**—Number of AMT membership update messages received with an address other than the configured AMT local address.<br>• **AMT incomplete relay discovery messages**—Number of AMT relay discovery messages received that are not fully formed.<br>• **AMT incomplete membership request messages**—Number of AMT membership request messages received that are not fully formed.<br>• **AMT incomplete membership update messages**—Number of AMT membership update messages received that are not fully formed.<br>• **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.<br>• **AMT invalid inner header checksum**—Number of AMT membership update messages received with an invalid IP checksum.<br>• **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 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 1151
- show amt statistics on page 1152
- show amt tunnel on page 1157

**List of Sample Output**
show amt summary on page 1156

**Output Fields**
Table 56 on page 1155 describes the output fields for the **show amt summary** command. Output fields are listed in the approximate order in which they appear.

### Table 56: 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 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 1151
- show amt statistics on page 1152
- show amt summary on page 1155

**List of Sample Output**

show amt tunnel on page 1159
show amt tunnel detail on page 1159
show amt tunnel tunnel-interface on page 1159
show amt tunnel gateway-address on page 1159
show amt tunnel gateway-address detail on page 1160

**Output Fields**

Table 57 on page 1157 describes the output fields for the **show amt tunnel** command. Output fields are listed in the approximate order in which they appear.

Table 57: 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 |

Table 57: show amt tunnel Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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**.<br>• **Active**—The tunnel is active.<br>• **Pending**—The tunnel creation is pending. This is a transient state.<br>• **Down**—The tunnel is in the down state.<br>• **Graceful restart pending**—Graceful restart is in progress.<br>• **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:<br>• **AMT Request**—Number of AMT relay tunnel request messages received.<br>• **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
```

CHAPTER 39

# Session Announcement Protocol Operational Commands

## 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 1162
show sap listen brief on page 1162
show sap listen detail on page 1162

**Output Fields**

Table 58 on page 1162 describes the output fields for the **show sap listen** command. Output fields are listed in the approximate order in which they appear.

Table 58: 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 1162.

**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 1162.

CHAPTER 40

# MSDP Operational Commands

## 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.

**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 1166
- show msdp source-active on page 1168
- show msdp statistics on page 1171

**List of Sample Output**

show msdp on page 1165
show msdp brief on page 1165
show msdp detail on page 1165

**Output Fields**

Table 59 on page 1164 describes the output fields for the **show msdp** command. Output fields are listed in the approximate order in which they appear.

Table 59: 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 |
| **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 |

Table 59: show msdp Output Fields *(continued)*

| Field Name | Field Description | Level of Output |
|---|---|---|
| **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 1165.

### 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.

**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 1164
- show msdp source-active on page 1168
- show msdp statistics on page 1171

**List of Sample Output**

show msdp source on page 1167

**Output Fields** Table 60 on page 1167 describes the output fields for the **show msdp source** command. Output fields are listed in the approximate order in which they appear.

### Table 60: 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:<br>• **Configured**—Source-active limit explicitly configured for this source.<br>• **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.

**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 1164
- show msdp source on page 1166
- show msdp statistics on page 1171

**List of Sample Output**

show msdp source-active on page 1170
show msdp source-active brief on page 1170
show msdp source-active detail on page 1170
show msdp source-active source on page 1170

**Output Fields**

Table 61 on page 1169 describes the output fields for the **show msdp source-active** command. Output fields are listed in the approximate order in which they appear.

### Table 61: 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 1170.

### 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 1170.

### 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.

**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

**List of Sample Output**

show msdp statistics on page 1174
show msdp statistics peer on page 1174

**Output Fields**

Table 62 on page 1171 describes the output fields for the **show msdp statistics** command. Output fields are listed in the approximate order in which they appear.

### Table 62: 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). |
| **Global active source limit log interval** | Time (in seconds) between consecutive log messages. |
| **Peer** | Address of peer. |

Table 62: show msdp statistics Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **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. |
| **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. |
| **Keepalive messages sent** | Number of keepalive messages sent. |
| **Keepalive messages received** | Number of keepalive messages received. |
| **Unknown messages received** | Number of unknown messages received. |

Table 62: show msdp statistics Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **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 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 usage

**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.

**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 1177
show multicast usage brief on page 1177
show multicast usage instance on page 1177
show multicast usage detail on page 1177

**Output Fields**

Table 63 on page 1175 describes the output fields for the **show multicast usage** command. Output fields are listed in the approximate order in which they appear.

### Table 63: show multicast usage Output Fields

| Field Name | Field Description |
|---|---|
| **Instance** | Name of the routing instance. (Displayed when multicast is configured within a routing instance.) |

Table 63: show multicast usage Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **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
228.0.0.0       1       52847               4439148
239.1.1.1       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 1177.

### show multicast usage instance

```
user@host> show multicast usage instance VPN-A
Group           Sources Packets             Bytes
224.2.127.254   1       5538                509496
224.0.1.39      1       13                  624
224.0.1.40      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
228.0.0.0       1       53159               4465356
    Source: 10.255.14.144   /32 Packets: 53159 Bytes: 4465356
239.1.1.1       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: 228.0.0.0       Packets: 53159 Bytes: 4465356
    Group: 239.1.1.1       Packets: 13407 Bytes: 1122156
10.255.70.15    /32  1      43                  3374
    Group: 239.1.1.1       Packets: 43 Bytes: 3374
```

## show route table

**Syntax**

show route table *routing-table-name*
<brief | detail | extensive | terse>
<logical-system (all | *logical-system-name*)>

**Syntax (EX 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.

**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 name begins 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 inetflow detail on page 1187

**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 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)

10.255.245.12:1:4.0.0.0/8 (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

```
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
```

Target Filtering Is Configured)

```
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 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 111.222.5.254 via fxp0.0
1.0.0.1/32         *[Direct/0] 00:51:58
                    > via at-5/3/0.0
1.0.0.2/32         *[Local/0] 00:51:58
                      Local
12.12.12.21/32     *[Local/0] 00:51:57
                      Reject
13.13.13.13/32     *[Direct/0] 00:51:58
                    > via t3-5/2/1.0
13.13.13.14/32     *[Local/0] 00:51:58
                      Local
13.13.13.21/32     *[Local/0] 00:51:58
                      Local
13.13.13.22/32     *[Direct/0] 00:33:59
                    > via t3-5/2/0.0
127.0.0.1/32        [Direct/0] 00:51:58
                    > via lo0.0
111.222.5.0/24     *[Direct/0] 00:51:58
                    > via fxp0.0
111.222.5.81/32    *[Local/0] 00:51:58
                      Local
```

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: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 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: 65002
                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 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: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 00:45:09, metric 1
                      Receive
1                  *[MPLS/0] 00:45:09, metric 1
                      Receive
2                  *[MPLS/0] 00:45:09, metric 1
                      Receive
100000             *[L2VPN/7] 00:43:04
                    > via so-0/1/0.1, Pop
100001             *[L2VPN/7] 00:43:03
                    > via so-0/1/0.2, Pop       Offset: 4
100002             *[LDP/9] 00:43:22, metric 1
                      via so-0/1/2.0, Pop
                    > via so-0/1/3.0, Pop
100002(S=0)        *[LDP/9] 00:43:22, metric 1
                      via so-0/1/2.0, Pop
                    > via so-0/1/3.0, Pop
100003             *[LDP/9] 00:43:22, metric 1
                    > via so-0/1/2.0, Swap 100002
                      via so-0/1/3.0, Swap 100002
100004             *[LDP/9] 00:43:16, metric 1
                      via so-0/1/2.0, Swap 100049
                    > via so-0/1/3.0, Swap 100049
so-0/1/0.1         *[L2VPN/7] 00:43:04
                    > via so-0/1/2.0, Push 100001, Push 100049(top)
                      via so-0/1/3.0, Push 100001, Push 100049(top)
so-0/1/0.2         *[L2VPN/7] 00:43:03
                      via so-0/1/2.0, Push 100000, Push 100049(top) Offset: -4
                    > via so-0/1/3.0, Push 100000, Push 100049(top) Offset: -4
```

### 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 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300352(S=0)        *[RSVP/7/1] 00:08:00, metric 1
                    > to 8.64.0.106 via ge-1/0/1.0, label-switched-path lsp1_p2p
300384             *[RSVP/7/2] 00:05:20, metric 1
                    > to 8.64.1.106 via ge-1/0/0.0, Pop
300384(S=0)        *[RSVP/7/2] 00:05:20, metric 1
                    > to 8.64.1.106 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

1.1.1.11:1000:1:1/96 (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

1:10.255.2.202:65535:10.255.2.202/432
```

```
                    *[BGP/170] 00:02:37, localpref 100, from 10.255.2.202
                      AS path: I
                     > via so-0/1/3.0
1:10.255.2.203:65535:10.255.2.203/432
                    *[BGP/170] 00:02:37, localpref 100, from 10.255.2.203
                      AS path: I
                     > via so-0/1/0.0
1:10.255.2.204:65535:10.255.2.204/432
                    *[MVPN/70] 00:57:23, metric2 1
                      Indirect
5:10.255.2.202:65535:128:::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
6:10.255.2.203:65535:65000:128:::10.12.53.12:128:ffff::1/432
                    *[PIM/105] 00:02:37
                      Multicast (IPv6)
7:10.255.2.202:65535:65000:128:::192.168.90.2:128:ffff::1/432
                    *[MVPN/70] 00:02:37, metric2 1
                      Indirect
```

### 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 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: 65002 Peer AS: 65000
                Age: 4
                Task: BGP_65000.10.12.99.5+3792
                Announcement bits (1): 0-Flow
                AS path: 65000 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: 65002
                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@PE1> 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

1.1.1.2:100:1.1.1.2/96 AD
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
1.1.1.4:100:1.1.1.4/96 AD
                   *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
1.1.1.2:100:1:0/96 MH
                   *[VPLS/170] 1d 03:11:03, metric2 1
                         Indirect
1.1.1.4:100:1:0/96 MH
                   *[BGP/170] 1d 03:11:02, localpref 100, from 1.1.1.4
                      AS path: I, validation-state: unverified
                    >    via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.2:1.1.1.4/176
                   *[VPLS/7] 1d 03:11:02, metric2 1
                    >    via ge-1/2/1.5
1.1.1.4:NoCtrlWord:5:100:100:1.1.1.4:1.1.1.2/176
                   *[LDP/9] 1d 03:11:02
                         Discard
```

CHAPTER 41

# PGM Operational Commands

## clear pgm negative-acknowledgments

**Syntax** clear pgm negative-acknowledgments

**Release Information** Command introduced before Junos OS Release 7.4.

**Description** Clear the Pragmatic General Multicast (PGM) negative acknowledgment (NAK) state received.

**Options** This command has no options.

**Required Privilege Level** clear

**Related Documentation**
- show pgm negative-acknowledgments on page 1193

**List of Sample Output** clear pgm negative-acknowledgments on page 1190

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear pgm negative-acknowledgments

```
user@host> clear pgm negative-acknowledgments
```

## clear pgm source-path-messages

**Syntax** clear pgm source-path-messages

**Release Information** Command introduced before Junos OS Release 7.4.

**Description** Clear Pragmatic General Multicast (PGM) source-path messages.

**Options** This command has no options.

**Required Privilege Level** clear

**Related Documentation**
- show pgm source-path-messages on page 1195

**List of Sample Output** clear pgm source-path-messages on page 1191

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear pgm source-path-messages

```
user@host> clear pgm source-path-messages
```

## clear pgm statistics

**Syntax** clear pgm statistics

**Release Information** Command introduced before Junos OS Release 7.4.

**Description** Clear Pragmatic General Multicast (PGM) statistics.

**Options** This command has no options.

**Required Privilege Level** clear

**Related Documentation**
- show pgm statistics on page 1196

**List of Sample Output** clear pgm statistics on page 1192

**Output Fields** When you enter this command, you are provided feedback on the status of your request.

### Sample Output

clear pgm statistics

```
user@host> clear pgm statistics
```

## show pgm negative-acknowledgments

**Syntax** show pgm negative-acknowledgments

**Release Information** Command introduced before Junos OS Release 7.4.

**Description** Display the sent or received Pragmatic General Multicast (PGM) negative acknowledgments (NAKs), the source-path message (SPM) sequence number being negatively acknowledged, and the current state of repair.

**Options** This command has no options.

**Required Privilege Level** view

**List of Sample Output** show pgm negative-acknowledgments on page 1194

**Output Fields** Table 64 on page 1193 describes the output fields for the **show pgm negative-acknowledgments** command. Output fields are listed in the approximate order in which they appear.

### Table 64: show pgm negative-acknowledgments Output Fields

| Field Name | Field Description |
|---|---|
| **Global source id** | Global source identifier (GSI), which combines with the source port to determine the transport session identifier (TSI). |
| **Network layer address** | Network layer address of the local system. |
| **Source port** | Source port number, which is combined with the GSI to determine the TSI. |
| **SPM sequence number** | Numeric sequence identifier of the source-path message. |
| **Window (trailing/leading sequence)** | Range of sequence numbers used by the source for sequentially numbering and transmitting the most recent packets. The trailing (or left) edge of the transmit window is the sequence number of the oldest data packet available for repair from a source. The leading (or right) edge of the transmit window is defined as the sequence number of the most recent data packet a source has transmitted. |
| **Outstanding NAKS** | Total number of outstanding negative acknowledgments sent or received by the local system. NAK packets indicate that a packet in the expected original data sequence has been detected as missing.<br>• **Sequence number**—Numeric sequence identifier of the source-path message.<br>• **Group**—Group address.<br>• **Source**—Multicast source.<br>• **Interface**—Interface name.<br>• **Receiver**—IP address receiving the multicast. |

## Sample Output

### show pgm negative-acknowledgments

```
user@host> show pgm negative-acknowledgments
Global source ID: 010203040506 Source port: 1111
      Network layer address: 10.38.0.1
      SPM sequence number: 1
      Window (trailing/leading sequence): 0/1
      Outstanding NAKs:
              Sequence number: 1
              Group: 225.1.1.1
              Source: 192.168.195.121
              Interface: t3-0/2/0:0 Receiver: 10.38.0.10
```

## show pgm source-path-messages

**Syntax** show pgm source-path-messages

**Release Information** Command introduced before Junos OS Release 7.4.

**Description** Display the Pragmatic General Multicast (PGM) source-path messages received.

**Options** This command has no options.

**Required Privilege Level** view

**List of Sample Output** show pgm source-path-messages on page 1195

**Output Fields** Table 65 on page 1195 describes the output fields for the **show pgm source-path-messages** command. Output fields are listed in the approximate order in which they appear.

### Table 65: show pgm source-path-messages Output Fields

| Field Name | Field Description |
|---|---|
| **Global source ID** | Global source identifier (GSI), which combines with the source port to determine the transport session identifier (TSI). |
| **Port** | Source port number, which combines with the GSI to determine the TSI. |
| **SPM number** | Numeric sequence identifier of the source-path message. |
| **Trail number** | Sequence number of the oldest data packet available for repair from a source. |
| **Lead number** | Sequence number of the most recent data packet a source has transmitted. |
| **Network layer address** | Network layer address of the local system. |

## Sample Output

**show pgm source-path-messages**

```
user@host> show pgm source-path-messages
Global source ID  Port SPM number Trail number Lead number Network layer address
010203040506      1111          1            0           1 10.38.0.1
```

## show pgm statistics

**Syntax** show pgm statistics

**Release Information** Command introduced before Junos OS Release 7.4.

**Description** Display Pragmatic General Multicast (PGM) packet statistics, including general loss and repair statistics.

**Options** This command has no options.

**Required Privilege Level** view

**List of Sample Output** show pgm statistics on page 1198

**Output Fields** Table 66 on page 1196 describes the output fields for the **show pgm statistics** command. Output fields are listed in the approximate order in which they appear.

### Table 66: show pgm statistics Output Fields

| Field Name | Field Description |
|---|---|
| **PGM type, # received, # sent** | Number of packets received and sent for the following PGM packet types:<br>• **SPM**—Number of total source path messages received and sent by the local system. Source path messages (SPMs) are sent by a source to establish the source path state in network elements and to provide the transmit-window state to receivers.<br>• **POLL**—Total number of poll requests received and sent by the local system.<br>• **POLR**—Total number of poll responses received and sent by the local system.<br>• **ODATA**—Total number of original data packets received and sent by the local system.<br>• **RDATA**—Total number of repair data packets received and sent by the local system. RDATA packets are generated in response to negative acknowledgments (NAKs), which indicate a missing packet from the original data sequence.<br>• **NAK**—Total number of negative acknowledgments received and sent by the local system. NAK packets indicate that a packet in the expected original data sequence has been detected as missing.<br>• **NULLNAK**—Total number of null negative acknowledgments received and sent by the local system. NULLNAKs are transmitted by a designated local repairer that receives NAKs redirected to it by either receivers or network elements to provide flow-control feedback to a source.<br>• **NCF**—Total number of NAK confirmations received and sent by the local system. NAK confirmations are generated in response to NAK packets that are received.<br>• **SPMR**—Total number of source path message requests (SPMRs) received and sent by the local system. SPMRs are used to solicit a source path message from a source in a nonimplosive way. The typical application is for late-joining receivers to solicit source path messages directly from a source in order to be able to send NAKs for missing packets, without having to wait for a regularly scheduled source path message from that source.<br>• **OTHER**—Total number of other PGM packets received and sent by the local system. |
| **packets shorter than minimum PGM header length** | Total number of packets received with headers that are shorter than the minimum required PGM header length. |

Table 66: show pgm statistics Output Fields *(continued)*

| Field Name | Field Description |
|---|---|
| **packets received with incorrect check sum** | Total number of packets received with an incorrect checksum. The checksum field is the 1's complement of the 1's complement sum of the entire PGM packet, including the header. |
| **packets received with zero check sum** | Total number of packets received with a zero checksum. If the computed checksum is zero, it is transmitted as all ones. A value of zero in this field means that the transmitter generated no checksum. |
| **packets received with TSDU length incorrect** | Total number of packets received with an incorrect Transport Service Data Unit (TSDU) length (16 bits). |
| **packets received with SPM length incorrect** | Total number of packets received with an incorrect source path message length. |
| **packets received with unknown SPM address family** | Total number of packets received with an unknown source path message address family indicator (AFI). |
| **packets received with NAK length incorrect** | Total number of packets received with an incorrect NAK length. |
| **packets received with unknown NAK address family** | Total number of packets received with an unknown NAK address family indicator (AFI). |
| **packets received with NAK for unknown TSI** | Total number of NAK packets received with an unknown transport session identifier (TSI). |
| **packets received when NAK throttled** | Total number of packets received when NAK is throttled. |
| **packets received with NCF length incorrect** | Total number of packets received with an incorrect NAK confirmation length. |
| **packets received with unknown NCF address family** | Total number of packets received with an unknown NAK confirmation address family indicator (AFI). |
| **packets received with NCF for unknown TSI** | Total number of NAK confirmation packets received with an unknown transport session identifier (TSI). |
| **packets received with RDATA length incorrect** | Total number of packets received with an incorrect RDATA length. |
| **packets received with RDATA for unknown TSI** | Total number of RDATA packets received with an unknown transport session identifier (TSI). |

## Sample Output

### show pgm statistics

```
user@host> show pgm statistics
PGM type     # received     # sent
SPM                   0          0
POLL                  0          0
POLR                  0          0
ODATA                 0          0
RDATA                 0          0
NAK                   0          0
NULLNAK               0          0
NCF                   0          0
SPMR                  0          0
OTHER                 0          0

packets shorter than minimum PGM header length  :          0
packets received with incorrect check sum       :          0
packets received with zero check sum            :          0
packets received with TSDU length incorrect     :          0
packets received with SPM length incorrect      :          0
packets received with unknown SPM address family:          0
packets received with NAK length incorrect      :          0
packets received with unknown NAK address family:          0
packets received with NAK for unknown TSI       :          0
packets received when NAK throttled             :          0
packets received with NCF length incorrect      :          0
packets received with unknown NCF address family:          0
packets received with NCF for unknown TSI       :          0
packets received with RDATA length incorrect    :          0
packets received with RDATA for unknown TSI     :          0
```

CHAPTER 42

# DVMRP Operational Commands

## show dvmrp interfaces

**Syntax** show dvmrp interfaces
<logical-system (all | *logical-system-name*)>

**Release Information** 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 1201

**Output Fields** Table 67 on page 1200 describes the output fields for the **show dvmrp interfaces** command. Output fields are listed in the approximate order in which they appear.

### Table 67: 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. |
| **Mode** | DVMRP mode:<br>• **Forwarding**—DVMRP does both the routing and the multicast data forwarding.<br>• **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** 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 1203

**Output Fields** Table 68 on page 1202 describes the output fields for the **show dvmrp neighbors** command. Output fields are listed in the approximate order in which they appear.

### Table 68: 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:<br>• **1**—One way. The local router has seen the neighbor, but the neighbor has not seen the local router.<br>• **G**—Neighbor supports generation ID.<br>• **L**—Neighbor is a leaf router.<br>• **M**—Neighbor supports mtrace.<br>• **N**—Neighbor supports netmask in prune messages and graft messages.<br>• **P**—Neighbor supports pruning.<br>• **S**—Neighbor supports SNMP. |
| **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** 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 1205
show dvmrp prefix brief on page 1205
show dvmrp prefix detail on page 1205

**Output Fields** Table 69 on page 1204 describes the output fields for the **show dvmrp prefix** command. Output fields are listed in the approximate order in which they appear.

### Table 69: 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** |
| **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** 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 1207

**Output Fields** Table 70 on page 1206 describes the output fields for the **show dvmrp prunes** command. Output fields are listed in the approximate order in which they appear.

### Table 70: 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
...
```

PART 4

# Troubleshooting

CHAPTER 43

# Knowledge Base



## Verifying a Multicast Configuration

To verify a multicast configuration, 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 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.

### 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.

## 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.

**Related Documentation**

- *Junos OS Feature Support Reference for SRX Series and J Series Devices*
- Multicast Overview on page 3
- show sap listen on page 1162 in the Junos OS Operational Mode Commands
- show igmp interface on page 1020 in the Junos OS Operational Mode Commands
- show pim interfaces on page 945 in the Junos OS Operational Mode Commands
- show pim rps on page 962 in the Junos OS Operational Mode Commands
- show multicast rpf on page 1001 in the Junos OS Operational Mode Commands

PART 5

# Index

# Index

## Symbols



## A

## B



## C

## D

## E



## F



## G

## H



## I

## M

## N

## Q



## R

## S

## T

## U



## V



## W