JUNOS™ Internet Software

Multicast Protocols Configuration Guide

Release 8.2

Juniper Networks, Inc.

1194 North Mathilda Avenue

Sunnyvale, CA 94089

USA

408-745-2000

www.juniper.net

Part Number: 530-017599-01 Revision 1

JUNOS Internet Software Multicast Protocols Configuration Guide, Release 8.2
Writing: Walter Goralski
Editing: Stella Hackell
Illustration: Faith Bradford, Nathaniel Woodward
Cover design: Edmonds Design

Revision History
12 January 2007—Revision 1

The information in this document is current as of the date listed in the revision history.

Year 2000 Notice

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

Software License

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

End User License Agreement

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller. "Embedded Software" means Software which Juniper has embedded in the Juniper equipment.

3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use the Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius software on multiple computers requires multiple licenses, regardless of whether such computers are physically contained on a single chassis.

c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.

e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Embedded Software on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Interface Information. To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. Third Party Software. Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at http://www.gnu.org/licenses/gpl.html, and a copy of the LGPL at http://www.gnu.org/licenses/lgpl.html.

15. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattaché, soient redigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Abbreviated Table of Contents

Table of Contents

About This Guide

This preface provides the following guidelines for using the *JUNOS Internet Software Multicast Protocols Configuration Guide* and related Juniper Networks, Inc., technical documents:

Objectives

This guide provides an overview of the multicast protocols for the JUNOS Internet software and describes how to configure multicast protocols on the router.

NOTE: This guide documents Release 8.2 of the JUNOS Internet software. For additional information about the JUNOS software—either corrections to or information that might have been omitted from this guide—see the software release notes at http://www.juniper.net/.

Supported Routing Platforms

For the features described in this manual, the JUNOS software currently supports the following routing platforms:

- J-series
- M-series
- MX-series
- T-series

Audience

This guide is designed for network administrators who are configuring and monitoring a Juniper Networks routing platform.

To use this guide, you need a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. You must also be familiar with one or more of the following Internet routing protocols:

- Border Gateway Protocol (BGP)
- Distance Vector Multicast Routing Protocol (DVMRP)
- Intermediate System-to-Intermediate System (IS-IS)
- Internet Control Message Protocol (ICMP) router discovery
- Internet Group Management Protocol (IGMP)
- Multiprotocol Label Switching (MPLS)
- Open Shortest Path First (OSPF)
- Protocol-Independent Multicast (PIM)
- Resource Reservation Protocol (RSVP)
- Routing Information Protocol (RIP)
- Simple Network Management Protocol (SNMP)

Personnel operating the equipment must be trained and competent; must not conduct themselves in a careless, willfully negligent, or hostile manner; and must abide by the instructions provided by the documentation.

Using the Indexes

This guide contains two indexes: a complete index that includes topic entries, and an index of statements and commands only.

In the index of statements and commands, an entry refers to a statement summary section only. In the complete index, the entry for a configuration statement or command contains at least two parts:

- The primary entry refers to the statement summary section.
- The secondary entry, *usage guidelines*, refers to the section in a configuration guidelines chapter that describes how to use the statement or command.

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. 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 *JUNOS CLI User Guide*.

Documentation Conventions

Table 1 defines notice icons used in this guide.

Table 1: Notice Icons

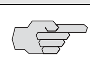
Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.

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

Table 2: Text and Syntax Conventions

Convention	Element	Example
Bold sans serif typeface	Represents text that you type.	To enter configuration mode, type the configure command: user@host> **configure**
`Fixed-width typeface`	Represents output on the terminal screen.	`user@host>` **`show chassis alarms`** `No alarms currently active`
Italic typeface	■ Introduces important new terms. ■ Identifies book names. ■ Identifies RFC and Internet draft titles.	■ A policy *term* is a named structure that defines match conditions and actions. ■ *JUNOS System Basics Configuration Guide* ■ RFC 1997, *BGP Communities Attribute*
Italic sans serif typeface	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Sans serif typeface	Represents names of configuration statements, commands, files, and directories; IP addresses; configuration hierarchy levels; or labels on routing platform components.	■ To configure a stub area, include the stub statement at the [edit protocols ospf area *area-id*] hierarchy level. ■ The console port is labeled CONSOLE.
< > (angle brackets)	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 (*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]
		routing-options {
		    static {
		        route default {
		            nexthop *address*;
		            retain;
		        }
		    }
		}

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

Related Juniper Networks Documentation

Table 3 lists the software and hardware guides and release notes for the supported Juniper Networks routing platforms and describes the contents of each document. Table 4 lists the books included in the *Network Operations Guide* series.

Table 3: Technical Documentation for Supported Routing Platforms (1 of 3)

Document	Description
JUNOS Internet Software Configuration Guides	
Class of Service	Provides an overview of the class-of-service (CoS) functions of the JUNOS software and describes how to configure CoS features, including configuring multiple forwarding classes for transmitting packets, defining which packets are placed into each output queue, scheduling the transmission service level for each queue, and managing congestion through the random early detection (RED) algorithm.
CLI User Guide	Describes how to use the JUNOS command-line interface (CLI) to configure, monitor, and manage Juniper Networks routing platforms. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
Feature Guide	Provides a detailed explanation and configuration examples for several of the most complex features in the JUNOS software.
MPLS Applications	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
Multicast Protocols	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
Network Interfaces	Provides an overview of the network interface functions of the JUNOS software and describes how to configure the network interfaces on the routing platform.
Network Management	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP and accounting options.
Policy Framework	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, forwarding options, and cflowd.
Routing Protocols	Provides an overview of routing concepts and describes how to configure routing, routing instances, and unicast routing protocols.
Secure Configuration Guide for Common Criteria and JUNOS-FIPS	Provides an overview of secure Common Criteria and JUNOS-FIPS protocols for the JUNOS Internet software and describes how to install and configure secure Common Criteria and JUNOS-FIPS on a routing platform.
Services Interfaces	Provides an overview of the services interfaces functions of the JUNOS software and describes how to configure the services interfaces on the routing platform.
Software Installation and Upgrade Guide	Provides a description of JUNOS software components and packaging, and includes detailed information about how to initially configure, reinstall, and upgrade the JUNOS system software. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
System Basics	Describes Juniper Networks routing platforms, and provides information about how to configure basic system parameters, supported protocols and software processes, authentication, and a variety of utilities for managing your router on the network.
VPNs	Provides an overview and describes how to configure Layer 2 and Layer 3 virtual private networks (VPNs), virtual private LAN service (VPLS), and Layer 2 circuits. Provides configuration examples.

Table 3: Technical Documentation for Supported Routing Platforms (2 of 3)

Document	Description
JUNOS References	
Hierarchy and RFC Reference	Describes the JUNOS configuration mode commands. Provides a hierarchy reference that displays each level of a configuration hierarchy, and includes all possible configuration statements that can be used at that level. This material was formerly covered in the *JUNOS System Basics Configuration Guide*.
Interfaces Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot interfaces.
Routing Protocols and Policies Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot routing protocols and policies, including firewall filters.
System Basics and Services Command Reference	Describes the JUNOS software operational mode commands you use to monitor and troubleshoot system basics, including commands for real-time monitoring and route (or path) tracing, system software management, and chassis management. Also describes commands for monitoring and troubleshooting services such as CoS, IP Security (IPSec), stateful firewalls, flow collection, and flow monitoring.
System Log Messages Reference	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
J-Web User Guide	
J-Web Interface User Guide	Describes how to use the J-Web GUI to configure, monitor, and manage Juniper Networks routing platforms.
JUNOS API and Scripting Documentation	
JUNOScript API Guide	Describes how to use the JUNOScript application programming interface (API) to monitor and configure Juniper Networks routing platforms.
JUNOS XML API Configuration Reference	Provides reference pages for the configuration tag elements in the JUNOS XML API.
JUNOS XML API Operational Reference	Provides reference pages for the operational tag elements in the JUNOS XML API.
JUNOS Configuration and Diagnostic Automation Guide	Describes how to use the commit script and self-diagnosis features of the JUNOS software. This guide explains how to enforce custom configuration rules defined in scripts, how to use commit script macros to provide simplified aliases for frequently used configuration statements, and how to configure diagnostic event policies.
NETCONF API Guide	Describes how to use the NETCONF API to monitor and configure Juniper Networks routing platforms.
JUNOS Comprehensive Index and Glossary	
Comprehensive Index and Glossary	Provides a complete index of all JUNOS software books, the *JUNOScript API Guide*, and the *NETCONF API Guide*. Also provides a comprehensive glossary.
JUNOScope Documentation	
JUNOScope Software User Guide	Describes the JUNOScope software GUI, how to install and administer the software, and how to use the software to manage routing platform configuration files and monitor routing platform operations.
J-series Services Router Documentation	
Getting Started Guide	Provides an overview, basic instructions, and specifications for J-series Services Routers. The guide explains how to prepare your site for installation, unpack and install the router and its components, install licenses, and establish basic connectivity. Use the *Getting Started Guide* for your router model.
Basic LAN and WAN Access Configuration Guide	Explains how to configure the interfaces on J-series Services Routers for basic IP routing with standard routing protocols, ISDN backup, and digital subscriber line (DSL) connections.

Table 3: Technical Documentation for Supported Routing Platforms (3 of 3)

Document	Description
Advanced WAN Access Configuration Guide	Explains how to configure J-series Services Routers in virtual private networks (VPNs) and multicast networks, configure data link switching (DLSw) services, and apply routing techniques such as policies, stateless and stateful firewall filters, IP Security (IPSec) tunnels, and class-of-service (CoS) classification for safer, more efficient routing.
Administration Guide	Shows how to manage users and operations, monitor network performance, upgrade software, and diagnose common problems on J-series Services Routers.
Hardware Documentation	
Hardware Guide	Describes how to install, maintain, and troubleshoot routing platforms and components. Each platform has its own hardware guide.
PIC Guide	Describes the routing platform PICs. Each platform has its own PIC guide.
Release Notes	
JUNOS Release Notes	Summarize new features and known problems for a particular software release, provide corrections and updates to published JUNOS, JUNOScript, and NETCONF manuals, provide information that might have been omitted from the manuals, and describe upgrade and downgrade procedures.
Hardware Release Notes	Describe the available documentation for the routing platform and the supported PICs, and summarize known problems with the hardware and accompanying software. Each platform has its own release notes.
JUNOScope Software Release Notes	Contain corrections and updates to the published JUNOScope manual, provide information that might have been omitted from the manual, and describe upgrade and downgrade procedures.
J-series Services Router Release Notes	Briefly describe the J-series Services Router features, identify known hardware problems, and provide upgrade and downgrade instructions.

Table 4: JUNOS Internet Software Network Operations Guides

Book	Description
Baseline	Describes the most basic tasks for running a network using Juniper Networks products. Tasks include upgrading and reinstalling JUNOS software, gathering basic system management information, verifying your network topology, and searching log messages.
Interfaces	Describes tasks for monitoring interfaces. Tasks include using loopback testing and locating alarms.
MPLS	Describes tasks for configuring, monitoring, and troubleshooting an example MPLS network. Tasks include verifying the correct configuration of the MPLS and RSVP protocols, displaying the status and statistics of MPLS running on all routers in the network, and using the layered MPLS troubleshooting model to investigate problems with an MPLS network.
MPLS Log Reference	Describes MPLS status and error messages that appear in the output of the show mpls lsp extensive command. The guide also describes how and when to configure Constrained Shortest Path First (CSPF) and RSVP trace options, and how to examine a CSPF or RSVP failure in a sample network.
Hardware	Describes tasks for monitoring M-series and T-series routing platforms.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. Send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at http://www.juniper.net/techpubs/docbug/docbugreport.html. If you are using e-mail, be sure to include the following information with your comments:

- Document name
- Document part number
- Page number
- Software release version

Requesting Support

For technical support, open a support case using the Case Manager link at http://www.juniper.net/support/ or call 1-888-314-JTAC (from the United States, Canada, or Mexico) or 1-408-745-9500 (from elsewhere).

Part 1

Introduction to Multicast

Chapter 1
Multicast Overview

The JUNOS software 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.

This chapter discusses the following topics:

What Is Multicast?

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 Web site server, to a single destination such as a client PC. This 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 public 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 Web site replication, and distributed interactive simulation (DIS) such as war games 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.

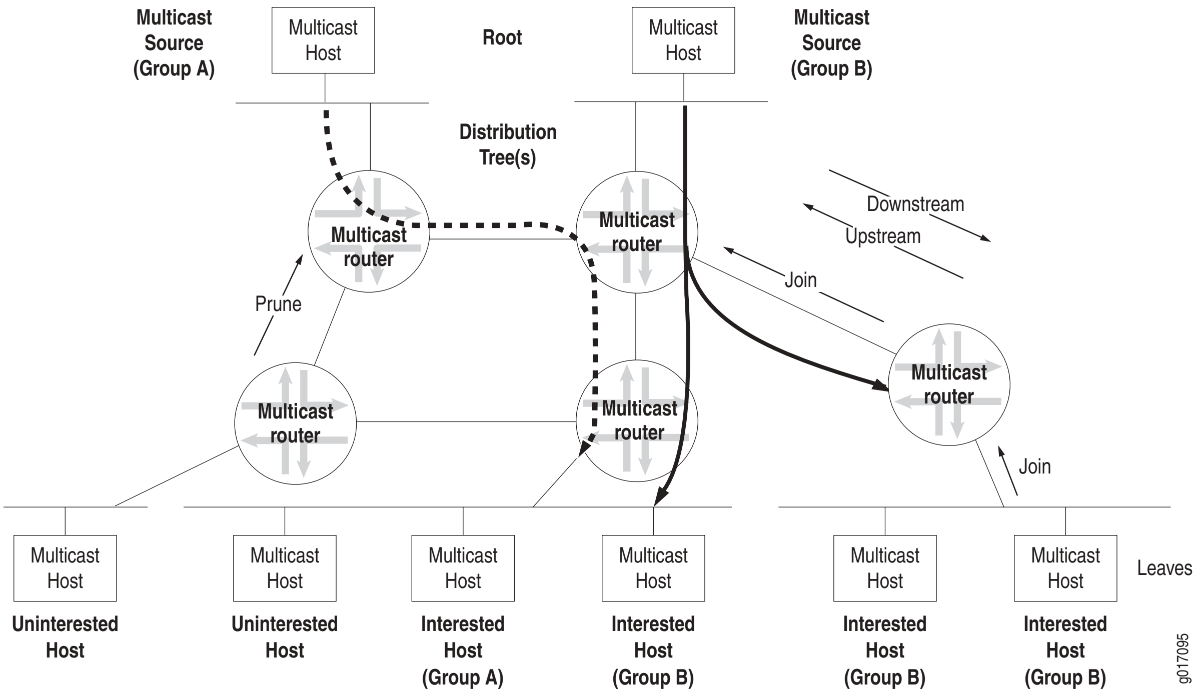
For radio station or news ticker traffic, multicast provides the most efficient and effective outcome, with none of the drawbacks and all of the advantages of the other methods. A single source of multicast packets finds its way to every *interested* receiver. As with broadcast, the transmitting host 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 shows a general view of some of the terms commonly used in an IP multicast network.

In a multicast network, the key component is the *router,* able to replicate packets and 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* to connect 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, there should be only one upstream interface on the router receiving 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 should 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.

Multicast terminology includes two more complex concepts:

- Leaf and Branch on page 7
- Protocols for Multicast Networks on page 8

Leaf and Branch

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 now 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.

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.

Protocols for Multicast Networks

The actions of receivers suggest two basic strategies for protocols to handle joining and pruning branches among a collection of multicast routers:

- Dense-mode multicast—The assumption could be made 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.

- Sparse-mode multicast—Alternatively, the assumption could be made that very few of the possible receivers want packets from this 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. This is the *sparse mode* of multicast operation. 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.

Some multicast routing protocols, especially older ones, support only dense-mode operation, which makes them inappropriate for use on the public Internet. Others allow sparse mode as well. If *sparse-dense mode* is supported, the multicast routing protocol allows some multicast groups to be sparse and other groups to be dense.

There is also a 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 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.

IP Multicast Building Blocks

Implementing an IP multicast network is made simpler by using a number of standardized building blocks. These IP multicast building blocks include a special IP multicast address space, administrative scoping to prevent large-scale routing loops, upstream and downstream interface lists, reverse path forwarding (RPF) to prevent small-scale routing loops, a shortest-path tree (SPT) algorithm to build a minimal distribution tree, and a rendezvous point (RP) and associated rendezvous-point tree (RPT) to allow sparse mode receivers to find sources.

There are six major IP multicast building blocks:

- IP Multicast Addressing on page 9
- Administrative Scoping on page 10
- Interface Lists on page 10
- Reverse-Path Forwarding on page 11
- Shortest-Path Tree on page 12
- Rendezvous Point, Shared Trees, and the Rendezvous-Point Tree on page 13

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 are concerned more with content than with physical devices. Receivers are not assigned their own multicast addresses, but need to know only 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.

Administrative Scoping

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 time-to-live (TTL) field in the IP packet header, but TTL scoping depends on intimate knowledge of the network topology by the network administrator. 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 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 should not 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 Open Shortest Path First (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.

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 should never 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 wild card 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.

For more information about the use of multicast forwarding state notations in different types of distribution trees, see "Rendezvous Point, Shared Trees, and the Rendezvous-Point Tree" on page 13. For more information about the use of multicast notations in different multicast routing protocols, see "Protocols for Multicast" on page 16.

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.

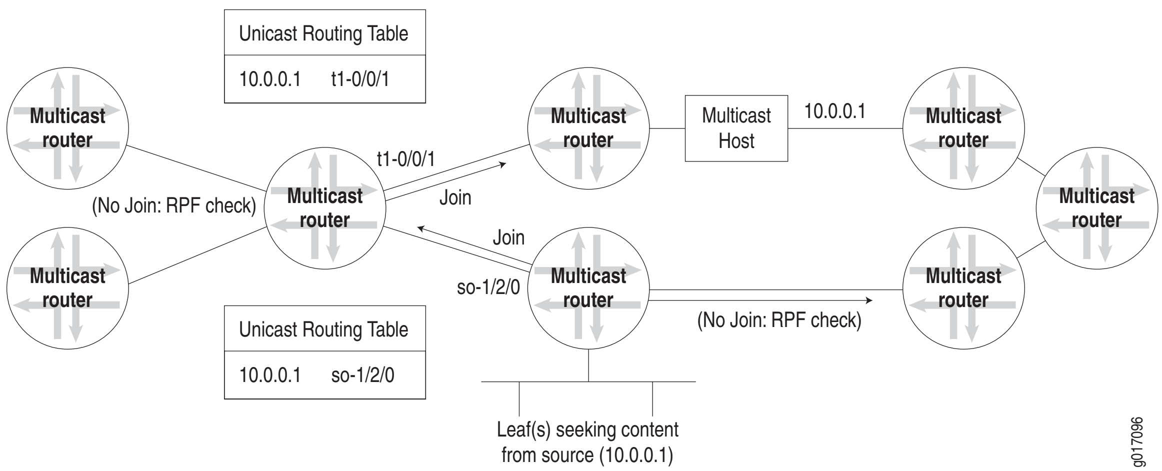
In multicast, the router forwards the packet *away* from the source to make progress along the distribution tree and prevent routing loops. The router's multicast forwarding state runs more logically by organizing tables based on the reverse path, from the receiver back to the root of the distribution 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 2 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 2: 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. The distribution tree should follow the shortest-path tree topology for efficiency. The RPF check helps to construct this tree.

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 in the router, or as complex as a set of protocols.

To build the SPT for that group, the router executes an 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 should 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 wishes 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 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 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 the RPF interface, building the SPT as it goes. The process stops when the join message:

- Reaches the router directly connected to the host that is the source, or
- 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 could be located more centrally in the network, on the backbone. This sharing of distribution trees with roots in the core network is accomplished by a multicast rendezvous point. For more information about RPs, see "Rendezvous Point, Shared Trees, and the Rendezvous-Point Tree" on page 13.

Rendezvous Point, Shared Trees, and the Rendezvous-Point Tree

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 knows 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 RP, not to the actual source of the content.

To join the shared tree, or *rendezvous-point tree (RPT)* as it 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 knows that multicast packets from this RP for this group should 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 wishes to join the shared tree for that group. This message is an (*,G) join message because S is not known, only the RP, 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 OIL for the group and also performs an RPF check on the RP address. The upstream router then sends an (*,G) join message out 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 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 that group

  - A router along the RPT that already has a multicast forwarding state for this group

  In either case, the branch is created, and packets can flow from the source to the RP and from the RP to the receiver. Note that there is no guarantee that the shared tree (RPT) is the shortest path tree to the source; most likely it is not. However, there are ways to "migrate" a shared tree to an SPT once the flow of packets begins. In other words, the forwarding state can transition from (*,G) to (S,G). The formation of both types of tree depends heavily on the operation of the RPF check and the *RPF table*. For more information about the RPF table, see "RPF Checks and the RPF Table" on page 14.

RPF Checks and the 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.

RPF Checks

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.

Populating the RPF Table

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 Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), OSPF, and 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 Border Gateway Protocol (MBGP) and multitopology routing in IS-IS (M-ISIS). 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-ISIS 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.

Protocols for Multicast

The protocols used among a collection of multicast-capable IP routers fall into two major categories:

- Multicast group membership protocols that are used between host and router (and host to host)
- Multicast routing protocols that are used between routers

The following sections describe:

- Multicast Group Membership Protocols on page 16
- Multicast Routing Protocols on page 17

Multicast Group Membership Protocols

Multicast group membership protocols allow a router to know 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 has to send only one copy of each packet for that multicast group out on that interface because of the inherent broadcast nature of LANs. Only when the router is informed by the multicast group membership protocol that there are no interested hosts on the subnet can the packets be withheld and that leaf pruned from the distribution tree.

There is one standard IP multicast group membership protocol: the Internet Group Management Protocol (IGMP). However, IGMP has several versions that are supported by hosts and routers. There are currently three versions of IGMP:

- 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)*. (RFC 1112 supported both many-to-many and one-to-many multicast, but one-to-many is considered the more viable model for the Internet at large.)

Although the various versions of IGMP are backward compatible, it is common for a router to run multiple versions of IGMP on LAN interfaces because 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 drops 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.

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.

There are five multicast routing protocols:

- 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).

- 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).

- PIM dense mode—This is PIM operating in dense mode (PIM DM), but the differences from PIM sparse mode are profound enough to consider the two modes separately. PIM also supports sparse-dense mode, with mixed sparse and dense groups, but there is no special notation for that operational mode. 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 sparse mode—Allows a router to use any unicast routing protocol and performs RPF checks using the unicast routing table. However, 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 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. However, 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.

- 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.

The differences among the five multicast routing protocols are summarized in Table 5.

Table 5: 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
CBT	No	Yes	No	Yes	No	Yes

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 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.

Chapter 2

IP Multicast Overview

The JUNOS software implements the following protocols to support IP multicast routing:

- Internet Group Management Protocol (IGMP), versions 1 and 2—Used to learn whether group members are present, for IP version 4 (IPv4) routers.
- Multicast Listener Discovery (MLD), versions 1 and 2—Used to learn whether group members are present, for IP version 6 (IPv6) routers.
- Distance Vector Multicast Routing Protocol (DVMRP)—Dense-mode multicast routing protocol.
- Protocol Independent Multicast (PIM)—Multicast routing protocol that routes to multicast groups that might span wide-area and interdomain internetworks. Both dense mode and sparse mode are supported.
- Multicast Source Discovery Protocol (MSDP)—Multicast routing protocol that discovers active sources of multicast messages. PIM sparse mode uses these sources.
- Session Announcement Protocol (SAP) and Session Description Protocol (SDP)—Handle conference session announcements.

This chapter discusses the following topics:

IP Multicast Standards

The protocols related to IP multicast are defined in the following documents:

- 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 2362, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification*
- RFC 2365, *Administratively Scoped IP Multicast*
- RFC 2547, *BGP/MPLS VPNs*
- RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*
- RFC 2858, *Multiprotocol Extensions for BGP-4*
- RFC 2974, *Session Announcement Protocol*
- RFC 3208, *PGM Reliable Transport Protocol Specification*
- RFC 3376, *Internet Group Management Protocol, Version 3* (source-specific multicast [SSM] include mode only)
- RFC 3446, *Anycast Rendezvous Point (RP) Mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*
- RFC 3569, *An Overview of Source-Specific Multicast (SSM)*
- RFC 3590 (SSM include mode only), *Source Address Selection for Multicast Listener Discovery Protocol*
- RFC 3618, *Multicast Source Discovery Protocol (MSDP)*
- RFC 3973, *Protocol Independent Multicast —Dense Mode (PIM-DM)*
- RFC 4601, *Protocol Independent Multicast —Sparse Mode (PIM-SM)*
- Internet draft draft-ietf-pim-sm-bsr-05.txt, *Bootstrap Router (BSR) Mechanism for PIM Sparse Mode* (expired August 2005) (no support for draft’s scoping mechanism)
- Internet draft draft-ietf-idmr-dvmrp-v3-11.txt, *Distance Vector Multicast Routing Protocol* (expired April 2004)
- Internet draft draft-raggarwa-l3vpn-2547-mvpn-00.txt, *Base Specification for Multicast in BGP/MPLS VPNs,* Section 2 (expired December 2004)
- Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs,* Section 2 (expired October 2003)

- Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP VPNs,* data MDTs only (expired April 2004)
- Internet draft draft-ietf-ssm-arch-06.txt, *Source-Specific Multicast for IP* (expired March 2005)
- Internet draft draft-ietf-mboned-ssm232-08.txt, *Source-Specific Protocol Independent Multicast in 232/8* (expired September 2004)
- Internet draft draft-holbrook-idmr-igmpv3-ssm-07.txt, *Using IGMPv3 and MLDv2 for Source-Specific Multicast* (expired December 2004)
- Internet draft draft-raggarwa-l3vpn-2547-mvpn-00.txt, *Base Specification for Multicast in BGP/MPLS VPNs* (expired December 2004)

NOTE: The implementation of the Distance Vector Multicast Routing Protocol (DVMRP) is based on a series of expired Internet drafts, as are all current implements of DVMRP. None are based on RFC 1075, *Distance Vector Multicast Routing Protocol*.

To access Internet RFCs and drafts, go to the IETF Web site at http://www.ietf.org.

Multicast Overview

IPv4 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 and 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.

The Internet Multicast Backbone (MBone) is an interconnected set of subnetworks and routers that support the delivery of IP multicast traffic. The MBone is a virtual network that is layered on top of sections of the physical Internet. The MBone is composed of islands of multicast routing capability that are connected to other islands by virtual point-to-point links called tunnels. The tunnels allow multicast traffic to pass undisturbed through the parts of the Internet that are not multicast-capable. Because the MBone and the Internet have different topologies, multicast routers execute a separate routing protocol to decide how to forward multicast packets.

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 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.

Multicast Redundancy

The JUNOS software supports nondisruptive, graceful Routing Engine switchover and graceful restart for some routing protocols in the case of Routing Engine or routing process failure. You can configure many routing protocols to continue to forward packets during the Routing Engine switchover or routing process restart. To support graceful Routing Engine switchover, the router must have two Routing Engines installed and be configured properly. For more information about graceful Routing Engine switchover, see the *JUNOS System Basics Configuration Guide*.

Graceful restart of routing protocol processes is required for graceful switchover. By default, graceful restart is disabled and must be enabled at the [edit routing-options graceful-restart] hierarchy level.

PIM sparse mode uses a mechanism called a *generation identifier* to indicate the need for graceful restart. Generation identifiers are included by default in PIM hello messages, as specified in the Internet draft draft-ietf-pim-sm-v2-new-10.txt. An initial generation identifier is created by each PIM neighbor to establish device capabilities. When one of the PIM neighbors restarts, it sends a new generation identifier to its neighbors. All neighbors that support graceful restart and are connected by point-to-point links assist by sending multicast updates to the restarting neighbor.

The restart phase is completed either when the PIM state becomes stable or when the restart interval timer expires. If the neighbors do not support graceful restart or if they connect to each other using multipoint interfaces, the restarting router uses a restart interval timer to define the restart period.

Graceful restart is compatible with the use of all multicast protocols. However, only PIM benefits from this feature. The router does not forward multicast packets for protocols other than PIM during graceful restart or switchover, because all other multicast protocols must completely restart after a routing process failure.

To configure graceful restart for PIM, include the graceful-restart statement at the [edit routing-options] hierarchy level, and include the pim statement at the [edit protocols pim] hierarchy level:

```
[edit routing-options]
graceful-restart;

[edit protocols]
pim {...};
```

For more information about graceful restart for PIM sparse mode, see “Configuring PIM Sparse Mode Graceful Restart” on page 234 and the *JUNOS Feature Guide*. For more information about graceful restart and other routing protocols, see the *JUNOS Routing Protocols Configuration Guide*.

Replicating Multicast Packets

The Juniper Networks T-series routing platforms 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.

Part 2

IP Multicast Configuration

Chapter 3

Introduction to PIM

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.

This chapter provides an overview of the features and capabilities of PIM.

This chapter discusses the following topics:

PIM Background

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 a bottlenecked core. PIM was designed to avoid the dense-mode scaling issues of DVMRP and the potential performance problems 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 or 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.

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 everything 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. So 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* running the PIM *bootstrap router (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 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).

PIM Modes of Operation

PIM operates in two basic modes: sparse mode and dense mode. In addition, PIM can operate in sparse-dense mode, with some multicast groups configured as dense mode (flood-and-prune, (S,G) state) and others 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.

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 need not introduce and manage a separate routing protocol just for RPF checks. Unicast routing protocols extended as multiprotocol Border Gateway Protocol (MBGP) and Multitopology Routing in Intermediate System-to-Intermediate System (M-ISIS) were later employed to build special tables to perform RPF checks, but PIM dense mode does not require them.

PIM dense mode can use the unicast routing table populated by Open Shortest Path First (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-ISIS when performing RPF checks.

PIM Sparse Mode

These are the major characteristics of PIM sparse mode:

- 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 then get content directly from the source.
- This transitional aspect of PIM sparse mode from shared to source-based tree is one of the major attractions of PIM. This feature 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 each other.
- 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 CBT.

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

PIM SSM

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 knows 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 are responsible for knowing 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 makes perfect sense. However, by ignoring the many-to-many model and focusing attention on the one-to-many 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 6.

Table 6: 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.

Mixing Modes

It is possible to mix PIM dense mode, PIM sparse mode, and PIM 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.

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 the scaling issues make dense modes inappropriate for Internet use beyond very restricted uses.

PIM sparse mode was capable of forming 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.

Chapter 4
Complete Multicast Configuration Statements

This chapter shows the complete configuration statement hierarchy for the portions of the configuration discussed in this manual, listing all possible configuration statements and showing their level in the configuration hierarchy. When you are configuring the JUNOS software, your current hierarchy level is shown in the banner on the line preceding the user@host# prompt.

For a list of the complete configuration statement hierarchy, see the *JUNOS System Basics Configuration Guide*.

This chapter is organized as follows:

[edit protocols] Hierarchy Level

```
[edit]
protocols {
```

Distance Vector Multicast Routing Protocol (DVMRP)

```
    dvmrp {
        disable;
        export [ policy-names ];
        import [ policy-names ];
        interface interface-name {
            disable;
            hello-interval seconds;
            hold-time seconds;
            metric metric;
            mode (forwarding | unicast-routing);
        }
        rib-group group-name;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

Internet Group Management Protocol (IGMP)

```
    igmp {
        interface interface-name {
            disable;
            ssm-map ssm-map-name;
            static {
                    group group {
                        source source;
                }
            }
            version version;
        }
        query-interval seconds;
        query-last-member-interval seconds;
        query-response-interval seconds;
        robust-count number;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

Multicast Listener Discovery (MLD)

```
    mld {
        interface interface-name {
            disable;
            ssm-map ssm-map-name;
            static {
                group group {
                    source source;
                }
            }
            version version;
        }
```

```
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Multicast Source Discovery Protocol (MSDP)

```
msdp {
    active-source-limit {
        maximum number;
        threshold number;
    }
    data-encapsulation <disable | enable>;
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    rib-group group-name;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    peer address {
        active-source-limit {
            maximum number;
            threshold number;
        }
        authentication-key peer-key;
        default-peer;
        disable;
        export [ policy-names ];
        import [ policy-names ];
        local-address address;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <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>;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
        peer address {
            authentication-key peer-key;
```

```
            default-peer;
            disable;
            export [ policy-names ];
            import [ policy-names ];
            local-address address;
            traceoptions {
                file name <replace> <size size> <files number> <no-stamp>
                    <world-readable | no-world-readable>;
                flag flag <flag-modifier> <disable>;
            }
        }
    }
}
```

Pragmatic General Multicast (PGM)

```
pgm {
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier>;
    }
}
```

Protocol Independent Multicast (PIM)

```
pim {
    assert-timeout seconds;
    dense-groups {
        addresses;
    }
    disable;
    graceful-restart {
        disable;
        restart-duration seconds;
    }
    import [ policy-names ];
    interface interface-name {
        disable;
        bfd-liveness-detection {
                minimum-interval milliseconds;
                minimum-receive-interval milliseconds;
                minimum-transmit-interval milliseconds;
                multiplier number;
                version (0 | 1 | automatic);
            }
        hello-interval seconds;
        mode (dense | sparse | sparse-dense);
        neighbor-policy policy-name;
        priority number;
        version version;
    }
    rib-group group-name;
    rp {
        auto-rp {
            (announce | discovery | mapping);
            (mapping-agent-election | no-mapping-agent-election);
        }
        bootstrap {
            family (inet | inet6) {
                priority number;
```

```
                    import [ policy-names ];
                    export [ policy-names ];
                }
            }
            bootstrap-export [ policy-names ];
            bootstrap-import [ policy-names ];
            bootstrap-priority number;
            dr-register-policy [ policy-names ];
            embedded-rp {
                maximum-rps limit;
                group-ranges {
                    destination-mask;
                }
            }
            local {
                family (inet | inet6) {
                    disable;
                    address address;
                    anycast-pim {
                        rp-set {
                            address address [forward-msdp-sa];
                        }
                        local-address address;
                    }
                    group-ranges {
                        destination-mask;
                    }
                    hold-time seconds;
                    priority number;
                }
            }
            rp-register-policy [ policy-names ];
            spt-threshold {
                infinity [ spt-threshold-infinity-policies ];
            }
            static {
                address address {
                    version version;
                    group-ranges {
                        destination-mask;
        }
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

Session Announcement Protocol/Session Description Protocol (SAP/SDP)

```
    sap {
        disable;
        listen [ address <port port>];
    }
}
```

[edit routing-instances] Hierarchy Level

```
[edit]
routing-instances {
    routing-instance-name {
        interface interface-name;
        instance-type (forwarding | no-forwarding | virtual-router | vpls | vrf);
        protocols {
            pim {
                mdt {
                    group-range multicast-prefix;
                    threshold {
                        group group-address {
                            source source-address {
                                rate threshold-rate;
                            }
                        }
                    }
                    tunnel-limit limit;
                }
                vpn-group-address class-D-address; /* only for the vrf instance type */
            }
        }
    }
}
```

NOTE: All other statements at the [edit protocols pim] and [edit protocols msdp] hierarchy levels can be configured at the [edit routing-instances *routing-instance-name* protocols pim] and [edit routing-instances *routing-instance-name* protocols msdp] hierarchy levels. The listed statements are only valid in a routing instance.

[edit routing-options] Hierarchy Level

```
[edit]
routing-options {
    multicast {
        flow-map flow-map-name {
            forwarding-cache {
                timeout (never | minutes);
                }
            policy policy-name;
            }
        }
        forwarding-cache {
            threshold suppress value <reuse value>;
            timeout minutes
        }
        rpf-check-policy [ policy-names ];
        scope scope-name {
            interface [ interface-names ];
            prefix destination-prefix;
        }
        scope-policy policy-name;
```

```
        ssm-groups {
            address;
        }
        ssm-map ssm-map-name {
            policy policy-name;
            source addresses;
        }
    }
}
```

[edit logical-routers *logical-router-name* protocols] Hierarchy Level

```
[edit logical-routers logical-router-name]
protocols {
```

DVMRP

```
    dvmrp {
        disable;
        export [ policy-names ];
        import [ policy-names ];
        interface interface-name {
            disable;
            hello-interval seconds;
            hold-time seconds;
            metric metric;
            mode (forwarding | unicast-routing);
        }
        rib-group group-name;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

IGMP

```
    igmp {
        interface interface-name {
            disable;
            ssm-map ssm-map-name;
            static {
                group group {
                    source source;
                }
            }
            version version;
        }
        query-interval seconds;
        query-last-member-interval seconds;
        query-response-interval seconds;
        robust-count number;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

MLD

```
mld {
    interface interface-name {
        disable;
        ssm-map ssm-map-name;
        static {
            group group {
                source source;
            }
        }
        version version;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

MSDP

```
msdp {
  active-source-limit {
    maximum number;
    threshold number;
  }
  data-encapsulation <disable | enable>;
  disable;
  export [ policy-names ];
  import [ policy-names ];
  local-address address;
  rib-group group-name;
  traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
  peer address {
    authentication-key peer-key;
    default-peer;
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
      file name <replace> <size size> <files number> <no-stamp>
        <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>;
    traceoptions {
      file name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
    peer address; {
      authentication-key peer-key;
      default-peer;
      disable;
      export [ policy-names ];
      import [ policy-names ];
      local-address address;
      traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
          <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
      }
    }
  }
}
```

PGM

```
pgm {
  traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier>;
  }
}
```

PIM

```
pim {
  assert-timeout seconds;
  dense-groups {
    addresses;
  }
  disable;
  graceful-restart {
    disable;
    restart-duration seconds;
  }
  import [ policy-names ];
  interface interface-name {
    disable;
    bfd-liveness-detection {
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        minimum-transmit-interval milliseconds;
        multiplier number;
        version (0 | 1 | automatic);
      }
    hello-interval seconds;
    mode (dense | sparse | sparse-dense);
    neighbor-policy policy-name;
    priority number;
    version version;
  }
  rib-group group-name;
  rp {
    auto-rp {
      (discovery | mapping);
      (mapping-agent-election | no-mapping-agent-election);
    }
    bootstrap {
      family (inet | inet6) {
        priority number;
        import [ policy-names ];
        export [ policy-names ];
      }
    }
    bootstrap-export [ policy-names ];
    bootstrap-import [ policy-names ];
    bootstrap-priority number;
    dr-register-policy [ policy-names ];
    embedded-rp {
      maximum-rps limit;
      group-ranges {
        destination-mask;
      }
    }
```

```
        local {
            family (inet | inet6) {
                disable;
                address address;
                anycast-pim {
                    rp-set {
                        address address [forward-msdp-sa];
                    }
                    local-address address;
                }
                group-ranges {
                    destination-mask;
                }
                hold-time seconds;
                priority number;
            }
        }
        rp-register-policy [ policy-names ];
        spt-threshold {
            infinity [ spt-threshold-infinity-policies ];
        }
        static {
            address address {
                version version;
                group-ranges {
                    destination-mask;
    }
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Session Announcement Protocol/Session Description Protocol (SAP/SDP)

```
    sap {
        disable;
        listen [ address <port port>];
    }
}
```

[edit logical-routers *logical-router-name* routing-instances] Hierarchy Level

```
[edit logical-routers logical-router-name]
routing-instances {
    routing-instance-name {
        interface interface-name;
        instance-type (forwarding | no-forwarding | virtual-router | vpls | vrf);
        protocols {
            pim {
                mdt {
                    group-range multicast-prefix;
                    threshold {
                        group group-address {
                            source source-address {
                                rate threshold-rate;
                            }
                        }
                    }
                    tunnel-limit limit;
                }
                vpn-group-address class-D-address; /* needed only for vrf instance type */
            }
        }
    }
}
```

NOTE: Almost all other statements at the [edit protocols pim] and [edit protocols msdp] hierarchy levels can be configured at the [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim] and [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols msdp] hierarchy level. For restrictions, see "PIM Configuration Guidelines" on page 223. The listed statements are valid in a routing instance only.

[edit logical-routers *logical-router-name* routing-options] Hierarchy Level

```
[edit logical-routers logical-router-name]
routing-options {
    multicast {
      flow-map flow-map-name {
        forwarding-cache {
          timeout (never | minutes);
          }
        policy policy-name;
        }
      }
      forwarding-cache {
        threshold suppress value <reuse value>;
        timeout minutes
      }
      rpf-check-policy [ policy-names ];
      scope scope-name {
        interface [ interface-names ];
        prefix destination-prefix;
      }
      scope-policy policy-name;
      ssm-groups {
        address;
      }
      ssm-map ssm-map-name {
        policy policy-name;
        source addresses;
      }
    }
}
```

Part 3

IGMP

Chapter 5
IGMP Overview

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 want to receive IP multicasts.

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 lists. Inclusion lists provide the ability 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 like an inclusion list, allowing 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. The following membership-report record types are supported for IGMPv3: mode is allowed, allow new sources, and block old sources.

For information about supported standards for IGMP, see “IP Multicast Standards” on page 20.

Chapter 6
IGMP Configuration Guidelines

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

```
igmp {
    interface interface-name {
        disable;
        ssm-map ssm-map-name;
        static {
            group group {
                source source;
            }
        }
        version version;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

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

This chapter describes the following tasks for configuring IGMP:

Minimum IGMP Configuration

IGMP is automatically enabled on all interfaces on which you configure PIM and all broadcast interfaces when you configure DVMRP. All IGMP configuration statements are optional.

Enabling IGMP

IGMP is automatically enabled on all interfaces on which you configure PIM and all broadcast interfaces when you configure DVMRP.

Optionally, you can specify the interface or interfaces on which to enable IGMP. If you do not specify any interfaces, IGMP is enabled on all interfaces on which you configure PIM and all broadcast interfaces when you configure DVMRP. To enable IGMP explicitly, include the igmp statement:

```
igmp {
    interface interface-name;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For information about specifying interface names, see the *JUNOS Network Interfaces Configuration Guide*.

Modifying the IGMP Host-Query Message Interval

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.

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

To modify this interval, include the query-interval statement:

query-interval *seconds*;

You can include this statement at the following hierarchy levels:

- [edit protocols igmp]
- [edit logical-routers *logical-router-name* protocols igmp]

The query interval value can be from 1 through 1024 seconds.

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. Varying this interval allows you to adjust the burst peaks of IGMP messages on the subnet.

By default, the query response interval is 10 seconds. To modify this interval, include the query-response-interval statement:

query-response-interval *seconds*;

You can include this statement at the following hierarchy levels:

- [edit protocols igmp]
- [edit logical-routers *logical-router-name* protocols igmp]

The query response interval can be from 1 through 1024 seconds. It must be less than the host-query message interval.

Modifying the 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 lower this interval to reduce the amount of time it takes a router to detect the loss of the last member of a group.

The default last-member query interval is 1 second. To modify this interval, include the query-last-member-interval statement:

```
query-last-member-interval seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols igmp]
- [edit logical-routers *logical-router-name* protocols igmp]

The last-member query interval can be from 1 through 1024 seconds.

Modifying the Robustness Variable

The IGMP robustness variable provides fine-tuning to allow for expected packet loss on a subnet. The value of the robustness variable is used in calculating the following IGMP message intervals:

- Group member interval—Amount of time that must pass before a multicast router decides 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 decides 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. You might want to increase this value if you expect a subnet to lose packets. To change the value of the robustness variable, include the robust-count statement:

```
robust-count number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols igmp]
- [edit logical-routers *logical-router-name* protocols igmp]

The number can be from 2 through 10.

Changing the IGMP Version

By default, the router runs IGMP version 2. To change to version 3 (for source-specific multicast [SSM] functionality), include the version statement:

```
version 3;
```

You can include this statement at the following hierarchy levels:

- [edit protocols igmp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols igmp interface *interface-name*]

To enable SSM functionality, you must configure version 3 on the host and the host's directly connected router.

NOTE: Routers running different versions of IGMP negotiate the lowest common version of IGMP that is supported by hosts on their subnet and operate in that version.

If you have already configured the router to use IGMP version 1 and then configure it to use IGMP version 2, the router continues to use IGMP version 1 for up to 6 minutes and then uses IGMP version 2.

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 receiving membership reports from host members.

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

To configure IGMP static membership, include the static statement. Then specify the group, or the group and its source or sources:

```
static {
    group group {
      source source;
    }
}
```

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

You can include this statement at the following hierarchy levels:

- [edit protocols igmp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols igmp interface *interface-name*]

Example: IGMP Static Group Membership

Configure IGMP static membership on the interface where the data is to be forwarded, and specify the groups 239.255.0.1 and 232.1.1.1 with the sources 10.1.1.1 and 10.1.1.2:

```
[edit]
protocols {
    igmp {
      interface ge-1/1/1.0 {
        static {
          group 239.255.0.1;
          group 232.1.1.1 {
            source 10.1.1.1;
            source 10.1.1.2;
          }
        }
      }
    }
}
```

Tracing IGMP Protocol Traffic

To trace IGMP protocol traffic, specify options to the traceoptions statement at the [edit routing-options] or [edit logical-routers *logical-router-name* routing-options] hierarchy level. Options applied at the routing options level trace all packets, and options applied at the protocol level trace only IGMP traffic.

You can specify IGMP-specific options by including the traceoptions statement:

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols igmp]
- [edit logical-routers *logical-router-name* protocols igmp]

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

- leave—Trace leave-group messages (for IGMP version 2 only).
- mtrace—Trace mtrace packets. Use the mtrace command to troubleshoot the software.
- packets—Trace all IGMP packets.
- query—Trace IGMP membership query messages, including general and group-specific queries.
- report—Trace membership report messages.

To trace the paths of multicast packets, use the mtrace command, as described in the *JUNOS System Basics and Services Command Reference*.

For information about tracing and global tracing options, see the *JUNOS Routing Protocols Configuration Guide*.

Example: Tracing IGMP Protocol Traffic

Trace only unusual or abnormal operations to the file routing-log, and trace all IGMP packets to the file igmp-log:

```
[edit]
routing-options {
    traceoptions {
        file routing-log;
        flag errors;
    }
}
protocols {
    igmp {
        traceoptions {
            file igmp-log;
            flag packets;
        }
    }
}
```

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-routers *logical-router-name* protocols igmp interface *interface-name*]

Chapter 7

Summary of IGMP Configuration Statements

The following sections explain each of the Internet Group Management Protocol (IGMP) configuration statements. The statements are organized alphabetically.

disable

Syntax disable;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Disable IGMP on the system.

Usage Guidelines See "Disabling IGMP" on page 57.

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

group

Syntax

```
group group {
    source source;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp interface *interface-name* static],
[edit protocols igmp interface *interface-name* static]

Release Information Statement introduced before JUNOS Release 7.4.

Description IGMP multicast group address that receives data on an interface.

Options *group*—Name of group.

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

The remaining statement is explained separately.

Usage Guidelines See “Enabling IGMP Static Group Membership” on page 55.

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

igmp

Syntax

```
igmp {
    interface interface-name {
        disable;
        static {
            group group {
                source source;
            }
        }
        version version;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols],
[edit protocols]

Release Information Statement introduced before JUNOS Release 7.4.

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

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).

Options The statements are explained separately.

Usage Guidelines See “Enabling IGMP” on page 52.

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

interface

Syntax

```
interface interface-name {
    disable;
    static {
        group group {
            source source;
        }
    }
    version version;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before JUNOS Release 7.4.

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. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

The remaining statements are explained separately.

Usage Guidelines See “Enabling IGMP” on page 52.

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

query-interval

Syntax query-interval *seconds*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description How often the querier router sends general host-query messages.

Options *seconds*—Time interval, in seconds.
Range: 1 through 1024
Default: 125 seconds

Usage Guidelines See “Modifying the IGMP Host-Query Message Interval” on page 53.

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

See Also query-last-member-interval on page 63, query-response-interval on page 63

query-last-member-interval

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

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description 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 1024
Default: 1 second

Usage Guidelines See "Modifying the Last-Member Query Interval" on page 54.

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

See Also query-interval on page 62, query-response-interval on page 63

query-response-interval

Syntax query-response-interval *seconds*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before JUNOS Release 7.4.

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

Options *seconds*—Time interval, in seconds. This interval must be less than the interval between general host-query messages.
Range: 1 through 1024
Default: 10 seconds

Usage Guidelines See "Modifying the IGMP Query Response Interval" on page 53.

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

See Also query-interval on page 62, query-last-member-interval on page 63

robust-count

Syntax robust-count *number* ;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Dimensionless factor used to calculate several IGMP message intervals. Raised for higher expected packet loss on a subnet.

Options *number*—Robustness variable.

Range: 2 through 10

Default: 2

Usage Guidelines See "Modifying the Robustness Variable" on page 54.

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

source

Syntax source *source*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp interface *interface-name* static group],
[edit protocols igmp interface *interface-name* static group]

Release Information Statement introduced before JUNOS Release 7.4.

Description IP version 4 (IPv4) unicast address that sends data on an interface.

Options *source*—IPv4 unicast address.

Usage Guidelines See "Enabling IGMP Static Group Membership" on page 55.

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

ssm-map

Syntax ssm-map *ssm-map-name*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced in JUNOS Release 7.4.

Description Apply an SSM map to an IGMP interface.

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

Usage Guidelines See “Example: Configuring SSM Mapping” on page 148.

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

static

Syntax

```
static {
    group group {
        source source;
    }
}
```

Hierarchy [edit logical-routers *logical-router-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Test multicast forwarding on an interface without a receiver host.

Options The remaining statements are explained separately.

Usage Guidelines See “Enabling IGMP Static Group Membership” on page 55.

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.

traceoptions

Syntax

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp],
[edit protocols igmp]

Release Information Statement introduced before JUNOS Release 7.4.

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, as described in the *JUNOS System Basics and Services Command Reference*.

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 *name*—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 also must specify a maximum file size with the size option.

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 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.

Usage Guidelines See “Tracing IGMP Protocol Traffic” on page 56.

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.

version

Statement version *version*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols igmp interface *interface-name*],
[edit protocols igmp interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify the version of IGMP.

Options *version*—IGMP version number.

Range: 1, 2, or 3
Default: IGMP version 2

NOTE: Routers running different versions of IGMP negotiate the lowest common version of IGMP that is supported by hosts on their subnet and operate in that version.

If you have already configured the router to use IGMP version 1 and then configure it to use IGMP version 2, the router continues to use IGMP version 1 for up to 6 minutes and then use IGMP version 2.

Usage Guidelines See "Changing the IGMP Version" on page 55.

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

Part 4

MLD

Chapter 8

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 subnet, 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 subnets 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 want to receive IP multicasts. The JUNOS software supports MLD versions 1and 2. Version 2 is supported for source-specific multicast (SSM) include mode only.

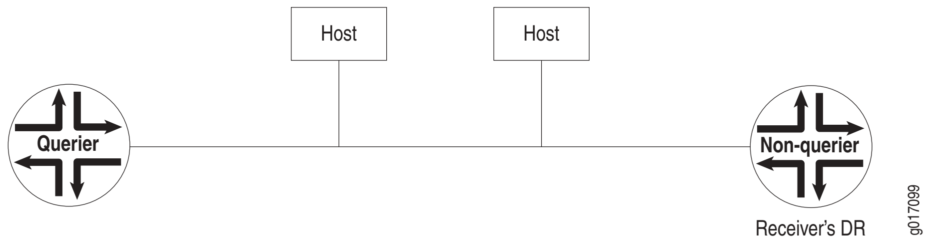
For information about supported standards for MLD, see "IP Multicast Standards" on page 20.

MLD Operation

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. For more information about PIM distribution trees, see "PIM Sparse Mode" on page 196. Nonquerier routers do not transmit MLD queries on a subnet but can do so if the querier router goes down.

All MLD-configured routers start up as querier routers on each attached subnet (see Figure 3 on page 72). The querier router on the right is the receiver's DR.

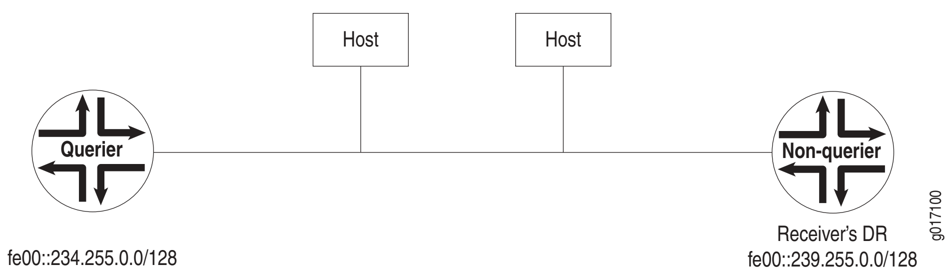
Figure 3: Routers Start Up on a Subnet



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 4, the router on the left has a source address numerically lower than the one on the right and therefore becomes the querier router.

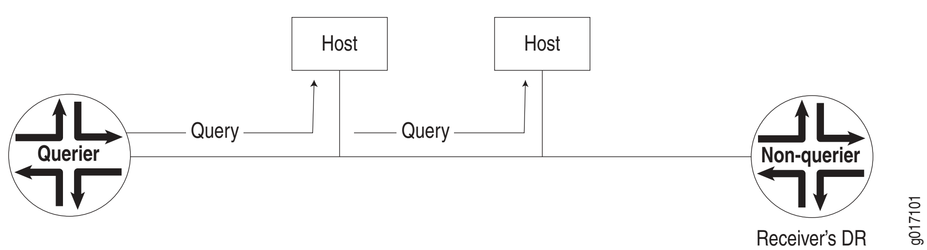
NOTE: In the practical application of MLD, several routers on a subnet are nonqueriers. If the elected querier router goes down, query messages are exchanged among the remaining routers. The router with the lowest IPv6 source address then becomes the new querier router.

Figure 4: Querier Router Is Determined



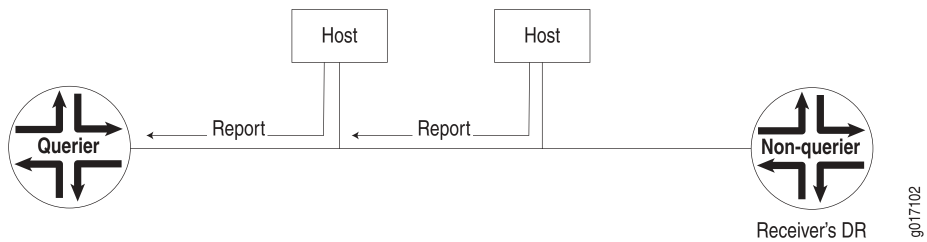
The querier router sends general MLD queries on the link-scope all-nodes multicast address FF02::1 at short intervals to all attached subnets to solicit group membership information (see Figure 5). Within the query message is the *maximum response delay* value, specifying the maximum allowed delay for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

Figure 5: General Query Message Is Issued



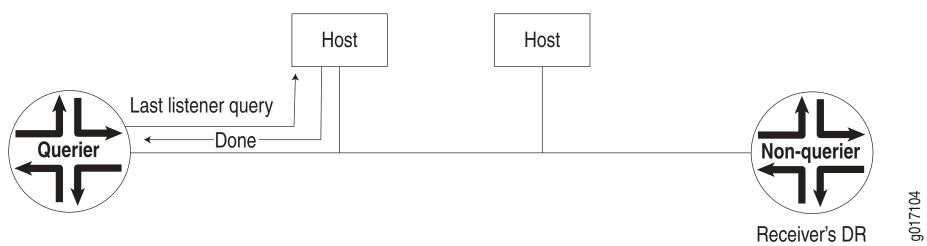
If interested listeners are attached to the host receiving the query, the host sends a report containing the host's IPv6 address to the router (see Figure 6). If the reported address is not yet in the router's list of multicast addresses with interested listeners, the address is added to the list and 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 6: Reports Are Received by the Querier Router



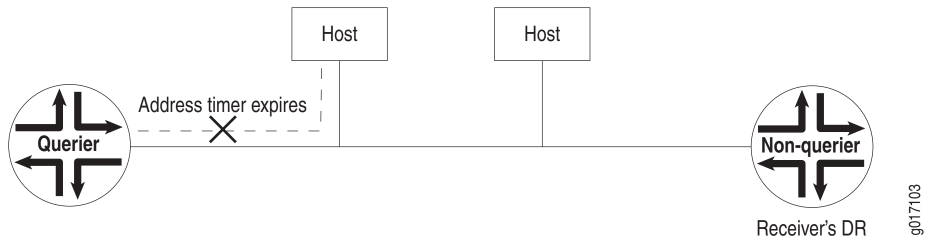
If the host has no interested multicast listeners, it sends a done message to the querier router. On receipt, the querier router issues a multicast-address-specific query containing the last listener query interval value to the multicast address of the host. If the router does not receive a report from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

Figure 7: 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 present on that subnet, removes the multicast address from the list, and notifies the RP in the PIM domain of its removal (see Figure 8 on page 74).

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



Chapter 9

MLD Configuration Guidelines

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

```
mld {
    interface interface-name {
        disable;
        ssm-map ssm-map-name;
        static {
            group group {
                source source;
            }
        }
        version version;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

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

This chapter describes the following tasks for configuring MLD:

- Minimum MLD Configuration on page 76
- Enabling MLD on page 76

Minimum MLD Configuration

MLD is automatically enabled on all broadcast interfaces when you configure PIM or DVMRP. All MLD configuration statements are optional.

Enabling MLD

MLD is automatically enabled on all broadcast interfaces when you configure PIM or DVMRP.

To enable MLD explicitly, include the mld statement. Optionally, you can specify the interface or interfaces on which to enable MLD. If you do not specify any interfaces, MLD is enabled on all interfaces.

```
mld {
    interface interface-name;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For information about specifying interface names, see the sections about interface naming in the *JUNOS Network Interfaces Configuration Guide*.

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.

```
version 2;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mld]
- [edit logical-routers *logical-router-name* protocols mld]

For more information about SSM, see "Source-Specific Multicast" on page 143.

Modifying the MLD Host-Query Message Interval

The MLD querier router periodically sends general host-query messages. These messages solicit group membership information and are sent to the link-scope all-nodes address FF02::1.

By default, host-query messages are sent every 125 seconds. You can change the number of MLD messages sent on the subnet by changing the query interval value. The larger the value, the less often MLD queries are sent.

To modify the interval value, include the query-interval statement:

```
query-interval seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mld]
- [edit logical-routers *logical-router-name* protocols mld]

The query interval value can range from 1 through 1024 seconds.

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. Varying this interval allows you to adjust the burst peaks of MLD messages on the subnet. Larger intervals create more widely spaced node responses and result in less bursty traffic.

By default, the query response interval is 10 seconds. To modify this interval, include the query-response-interval statement:

```
query-response-interval seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mld]
- [edit logical-routers *logical-router-name* protocols mld]

The query response interval can range from 1 through 1024 seconds. It must be less than the host-query message interval.

Modifying the 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 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.

The default last-member query interval is 1 second. To modify this interval, include the query-last-member-interval statement:

```
query-last-member-interval seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mld]
- [edit logical-routers *logical-router-name* protocols mld]

The last-member query interval can range from 1 through 1024 seconds.

Modifying the Robustness Variable

The MLD robustness variable provides fine-tuning to allow for expected packet loss on a subnet. 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 decides 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 decides 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. You might want to increase this value if you expect a subnet to lose packets. To change the value of the robustness variable, include the robust-count statement:

```
robust-count number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols mld]
- [edit logical-routers *logical-router-name* protocols mld]

The number can be from 2 through 10.

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 receiving membership reports from host members.

To configure MLD static membership, include the static statement. Then specify the group, or the group and its source or sources:

```
static {
    group group {
        source source;
    }
}
```

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

You can include this statement at the following hierarchy levels:

- [edit protocols mld interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols interface *interface-name*]

Example: MLD Static Group Membership

Configure MLD static membership on the interface where the data is to be forwarded, and specify the groups ff02::1:ff05:1a8d and ff02::1:ffa8:c34a with the source fe80::2e0:81ff:fe05:1a8d:

```
[edit ]
protocols {
    mld {
      interface fe-1/0/1.0 {
        static {
          group ff02::1:ff05:1a8d;
          group ff02::1:ffa8:c34a {
            source fe80::2e0:81ff:fe05:1a8d;
          }
        }
      }
    }
}
```

Tracing MLD Protocol Traffic

To trace MLD protocol traffic, you can specify options in the global traceoptions statement at the [edit routing-options] or [edit logical-routers *logical-router-name* routing-options] hierarchy level. Options applied at the routing options level trace all packets, and options applied at the protocol level trace only IGMP traffic. You can specify MLD-specific options by including the traceoptions statement:

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols mld]
- [edit logical-routers *logical-router-name* protocols mld]

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

- leave—Trace leave-group messages (for version 2 only).
- mtrace—Trace mtrace packets. Use the mtrace command to troubleshoot the software.
- packets—Trace all MLD packets.

- query—Trace MLD membership query messages, including general and group-specific queries.
- report—Trace membership report messages.

To trace the paths of multicast packets, use the mtrace command, as described in the *JUNOS System Basics and Services Command Reference*.

For information about tracing and global tracing options, see the *JUNOS Routing Protocols Configuration Guide.*

Example: Tracing MLD Protocol Traffic

Trace only unusual or abnormal operations to the file routing-log, and trace all MLD packets to the file mld-log:

```
[edit]
routing-options {
    traceoptions {
        file routing-log;
        flag errors;
    }
}
protocols {
    mld {
        traceoptions {
            file mld-log;
            flag packets;
        }
    }
}
```

Disabling MLD

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

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

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For information about specifying interface names, see the sections about interface naming in the *JUNOS Network Interfaces Configuration Guide*.

Chapter 10

Summary of MLD Configuration Statements

The following sections explain each of the Multicast Listener Discovery (MLD) configuration statements. The statements are organized alphabetically.

disable

Syntax	disable;
Hierarchy Level	[edit logical-routers *logical-router-name* protocols mld interface *interface-name*], [edit protocols mld interface *interface-name*]
Release Information	Statement introduced before JUNOS Release 7.4.
Description	Disable MLD on the system.
Usage Guidelines	See "Disabling MLD" on page 81.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

group

Syntax

```
group group {
    source source;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld interface *interface-name* static],
[edit protocols mld interface *interface-name* static]

Release Information Statement introduced before JUNOS Release 7.4.

Description MLD multicast group address that receives data on an interface.

Options *group*—Address of group.

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

The remaining statement is explained separately.

Usage Guidelines See "Enabling MLD Static Group Membership" on page 79.

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

interface

Syntax

```
interface interface-name {
    disable;
    static {
        group group {
            source source;
        }
    }
    version version;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before JUNOS 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. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

The remaining statements are explained separately.

Usage Guidelines See "Enabling MLD" on page 76.

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

mld

Syntax

```
mld {
    interface interface-name {
        disable;
        static {
            group group {
                source source;
            }
        }
        version version;
    }
    query-interval seconds;
    query-last-member-interval seconds;
    query-response-interval seconds;
    robust-count number;
    }
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols],
[edit protocols]

Release Information Statement introduced before JUNOS 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.

Usage Guidelines See “Enabling MLD” on page 76.

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

query-interval

Syntax query-interval *seconds*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before JUNOS Release 7.4.

Description How often the querier router sends general host-query messages.

Options *seconds*—Time interval, in seconds.
Range: 1 through 1024
Default: 125 seconds

Usage Guidelines See "Modifying the MLD Host-Query Message Interval" on page 77.

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

See Also query-last-member-interval on page 86, query-response-interval on page 87

query-last-member-interval

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

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before JUNOS Release 7.4.

Description 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

Usage Guidelines See "Modifying the Last-Member Query Interval" on page 78.

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

See Also query-interval on page 86, query-response-interval on page 87

query-response-interval

Syntax query-response-interval *seconds*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before JUNOS Release 7.4.

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

Options *seconds*—Time interval, in seconds. This interval must be less than the interval between general host-query messages.
Range: 1 through 1024
Default: 10 seconds

Usage Guidelines See "Modifying the MLD Query Response Interval" on page 78.

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

See Also query-interval on page 86, query-last-member-interval on page 86

robust-count

Syntax robust-count *number*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before JUNOS Release 7.4.

Description Tune for the expected packet loss on a subnet.

Options *number*—Time interval, in seconds. This interval must be less than the interval between general host-query messages.
Range: 2 through 10
Default: 2 seconds

Usage Guidelines See "Modifying the Robustness Variable" on page 79.

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

source

Syntax source *source*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld interface *interface-name* static group *group*],
[edit protocols mld interface *interface-name* static group *group*]

Release Information Statement introduced before JUNOS Release 7.4.

Description IP version 6 (IPv6) unicast address that sends data on an interface.

Options *source*—One or more IPv6 unicast addresses.

Usage Guidelines See "Enabling MLD Static Group Membership" on page 79.

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

ssm-map

Syntax ssm-map *ssm-map-name*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced in JUNOS Release 7.4.

Description Apply an SSM map to an MLD interface.

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

Usage Guidelines See "Example: Configuring SSM Mapping" on page 148.

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

static

Syntax

```
static {
    group group {
        source source;
    }
}
```

Hierarchy [edit logical-routers *logical-router-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Test multicast forwarding on an interface.

Options The remaining statements are explained separately.

Usage Guidelines See “Enabling MLD Static Group Membership” on page 79.

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.

traceoptions

Syntax

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols mld],
[edit protocols mld]

Release Information Statement introduced before JUNOS 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, as described in the *JUNOS System Basics and Services Command Reference*.

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 *name*—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 also must specify a maximum file size with the size option.

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 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.

Usage Guidelines See “Tracing MLD Protocol Traffic” on page 80.

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.

version

Syntax version *version*;

Hierarchy [edit logical-routers *logical-router-name* protocols mld interface *interface-name*],
[edit protocols mld interface *interface-name*]

Release Information Statement introduced before JUNOS 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)

Usage Guidelines See “Modifying the MLD Version” on page 77.

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.

Part 5

SAP and SDP

Chapter 11

SAP Overview

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. Only SAP has configuration parameters that users can change.

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 "IP Multicast Standards" on page 20.

Chapter 12

SAP Configuration Guidelines

The Session Announcement Protocol (SAP) and Session Description Protocol (SDP) 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. 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-routers *logical-router-name* protocols]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

SAP listens on one or more addresses and ports. By default, SAP always listens to the address and port 224.2.127.254:9875 for session advertisements. To add other addresses and ports, specify other address and port numbers.

Sessions learned by SDP, SAP's higher layer protocol, time out after 60 minutes.

Chapter 13

Summary of SAP Configuration Statements

The following sections explain each of the SAP multicast configuration statements. The statements are organized alphabetically.

disable

Syntax disable;

Hierarchy Level [edit logical-routers *logical-router-name* protocols sap],
[edit protocols sap]

Release Information Statement introduced before JUNOS Release 7.4.

Description Explicitly disable SAP.

Usage Guidelines See “SAP Configuration Guidelines” on page 97.

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

listen

Syntax listen [*address* <port *port*>];

Hierarchy Level [edit logical-routers *logical-router-name* protocols sap],
[edit protocols sap]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify one or more addresses and ports on which SAP and SDP listen.

SAP and SDP always listen on the default SAP address and port, 224.2.127.254:9875. To listen on additional addresses or address ranges, specify one or more addresses with the *address* and *port* options.

Options address—(Optional) Address where the router should listen for session advertisements.
Default: 224.2.127.254

port *port*—(Optional) Port where the router should listen for session advertisements.
Default: 9875

Usage Guidelines See "SAP Configuration Guidelines" on page 97.

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

sap

Syntax

```
sap {
    disable;
    listen [ address <port port> ];
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols],
[edit protocols]

Release Information Statement introduced before JUNOS 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 listen on additional addresses or address ranges, specify one or more addresses and ports with the listen statement.

Options The statements are explained separately.

Usage Guidelines See "SAP Configuration Guidelines" on page 97.

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

See Also listen on page 100

Part 6
PGM

Chapter 14
PGM Overview

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, leaves much to be desired in its reliability 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 IP protocol number 113.

Although PGM is mainly concerned 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.

For information about supported standards for PGM, see "IP Multicast Standards" on page 20.

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 source and receivers. All end devices must be PGM-enabled, although there can be non-PGM routers between 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 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 long enough or the transmit window small enough so that the source does not maintain a long session history, the receiver cannot get all required information.

This section describes in more detail the behavior of the three PGM elements in a multicast network:

- PGM-Enabled Source on page 105
- PGM-Enabled Receivers on page 105
- PGM-Enabled Routers on page 106

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 an NCF 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 knows that its NAK was received by the PGM next hop when it gets 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 get 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 DLR for a subnet. The receiver now has complete information or knows for certain 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.

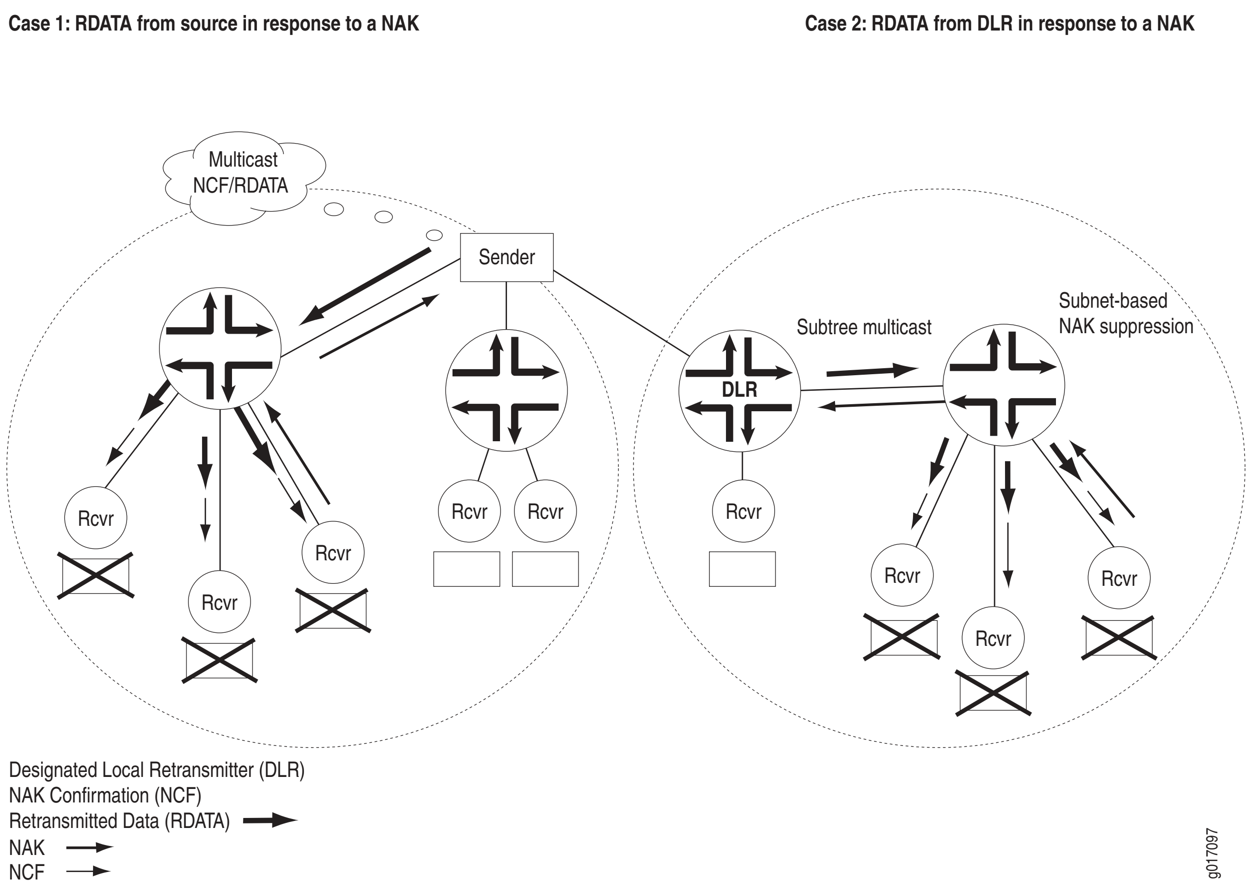
The PGM router multicasts NCFs in response to received NAKs on the downstream interfaces that received the NAKs. NCFs are not multicast on interfaces that have not received NAKs.

PGM routers must multicast all ODATA and RDATA packets 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 9 shows the overall PGM architecture and the role of PGM-enabled routers.

Figure 9: 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.

Chapter 15

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 is not directly concerned with 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-routers *logical-router-name* protocols]

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

```
pgm {
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier>;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

You can specify the following PGM-specific options in the PGM 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.

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

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

Chapter 16

Summary of PGM Configuration Statements

The following sections explain each PGM configuration statement. The statements are organized alphabetically.

pgm

Syntax

```
pgm {
    traceoptions {
      file name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier>;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols],
[edit protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure PGM globally and set tracing options.

To specify more than one tracing operation, include multiple flag statements.

Default The default PGM trace options are inherited from the routing protocol traceoptions statement included at the [edit routing-options] hierarchy level.

Options The remaining statement is explained separately.

traceoptions

Syntax

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier>;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pgm],
[edit protocols pgm]

Release Information Statement introduced before JUNOS 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.

file *name*—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 pgm-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 also must specify a maximum file size with the size option.

Range: 2 through 1000 files
Default: 2 files

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

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 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.

Usage Guidelines See "PGM Configuration Guidelines" on page 109.

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.

Part 7

Multicast Routing Instances

Chapter 17
Multicast Data MDT Overview

Protocol Independent Multicast (PIM) version 2 supports multicast over Layer 3 virtual private networks (VPNs) based on RFC 2547, *BGP/MPLS VPNs*, and Section 2 (Multicast Domains) of Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs*. This implementation does not require the provider (P) routers to maintain any VPN-specific PIM information, but this lack of VPN-specific information is not optimal. The issue is that a single multicast group is defined for each VPN to carry multicast control and data traffic inside the provider core and all VPNs are mapped to this single group in the provider's space. This mapping results in the delivery of packets to each provider edge (PE) router attached to the P router even if the PE router has no receivers for traffic from a multicast group in that VPN. Each PE router must process the encapsulated VPN traffic even if the multicast packets are then dropped. This is a waste of resources, especially in environments characterized by low bandwidth links in the core or a multicast source in the VPN sending a very high volume of information (for example, high-definition television [HDTV] packets) through the core.

A data multicast distribution tree (MDT), based on section 7 of Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs,* solves the problem of P routers flooding unnecessary multicast information to PE routers that have no interested receivers for a particular VPN multicast group. The multicast data MDT solution requires the creation of a new tunnel by the PE router if the source exceeds a configured rate threshold parameter. All other PE routers join the new tunnel only if the PE router has receivers in the VPN for that multicast group.

This chapter provides the following information about data MDTs:

Data MDT Creation Overview

Initially, the PE routers discover each other in a VPN routing and forwarding (VRF) instance using the default MDT. Each PE router configuration includes in its VRF instance various parameters to control the creation of a data MDT, such as when the source traffic in the VRF instance exceeds the configured threshold rate. The PE router monitors the rate during its periodic statistics-collection cycles. If the source locally attached to the PE router in the VPN exceeds this limit, the source PE advertises the new data MDT group and new MDT with a User Datagram Protocol (UDP) type-length-vector (TLV) packet called an *MDT join TLV*. The MDT join TLV describes the source and group pair (S,G) in the VRF instance and the new data MDT group address used in the provider space. The source PE periodically announces the MDT join TLV over the default MDT for that VRF instance as long as the source is active.

All PE routers receive the MDT join TLV because it is sent over the default MDT. Only the PE routers with receivers in the VRF instance for that multicast group can join the new group, and the PE routers must join the new group to receive the multicast traffic now sent over the new MDT by the source PE. PE routers without interested receivers listed in the VRF instance ignore the MDT join TLVs.

When remote PE routers join the new data MDT group, they send a PIM join message for the new group in the provider space. The PIM join message for the new group is sent directly to the source PE router from the remote PE routers by means of PIM source-specific multicast (SSM). SSM using (S,G) is possible with data MDT instead of the default MDT any-source multicast (ASM) (*,G) because the source address is known from the UDP signaling used with data MDT.

The source PE router starts encapsulating the multicast traffic for the entries in the VRF instance using the new data MDT group after 3 seconds, allowing time for the remote PE routers to switch to the new group. The source PE router then halts the flow of multicast packets over the default MDT, and the packet flow for the entries in the VRF instance source shifts to the newly created data MDT, joined only by PE routers with interested receivers.

When the preconfigured conditions, such as the rate threshold, are no longer met by the source because the source stops sending or the rate falls below the threshold, the source PE stops announcing the MDT join TLVs and the PE router switches to sending on the default MDT for that VRF instance again.

Data MDT Characteristics

The maximum number of data MDTs for all VPNs on a PE router is limited to 8000, 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 this limit is reached in the VRF instance, and all traffic for other sources that exceed the configured limits is still sent on the default MDT.

Creation of data MDTs depends on the monitoring of the multicast source data rate. This rate is checked once per minute, so the creation of data MDTs can be delayed up to 1 minute after a source exceeds a configured limit. In the same way, 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 MDT limit value do not affect existing tunnels that exceed the new limit. MDTs that are already active remain in place until the threshold conditions are no longer met.

To remove active MDTs no longer included in a newly configured group address range, you must restart the PIM routing instance. This restart clears all remnants of the former group addresses but disrupts routing and therefore requires a maintenance window for the change.

Multicast tunnel (mt) interfaces created because of exceeded thresholds are not recreated 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.

Chapter 18

MDT Configuration Guidelines

This chapter provides the following information about data multicast distribution trees (MDTs):

Configuring Data MDTs

To configure multicast data MDTs, include the mdt statement:

```
mdt {
      group-range multicast-prefix;
      threshold {
         group group-address {
            source source-address {
               rate threshold-rate;
            }
         }
      }
      tunnel-limit limit;
}
```

You can include the statement at the following hierarchy levels:

- [edit routing-instances *routing-instance-name* protocols pim]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim]

NOTE: Because MDT applies to virtual private networks (VPNs) and VPN routing and forwarding (VRF) instances, you cannot configure MDT statements in the master routing instance. If you configure MDT in the master routing instance, the configuration commit fails.

For an overview of routing instances and a detailed example of routing instance configuration, see the routing instances chapter of the *JUNOS Feature Guide*.

By default, creation of data MDTs is disabled.

This section describes the following tasks for configuring data MDTs:

- Configuring the Data MDT Group Range on page 122
- Configuring the Data MDT Threshold Parameters on page 122
- Configuring the Data MDT Limit on page 123

Data MDTs require a correctly configured Layer 3 VPN for multicast. For more information about configuring Layer 3 VPNs for multicast, see "Configuring Multicast for Layer 3 VPNs" on page 251.

Configuring the Data MDT Group Range

The provider edge (PE) router implementing data MDTs for a local multicast source must establish the group range to use for data MDTs created in this VRF instance. This address range cannot overlap with any of the default MDT addresses for all VPNs on the router. If you configure overlapping group ranges, the configuration commit fails.

To configure the data MDT group range, include the group-range statement:

```
group-range multicast-prefix;
```

You can include this statement at the following hierarchy levels:

- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim mdt]
- [edit routing-instances *routing-instance-name* protocols pim mdt]

Any multicast address range can be used as the multicast prefix, for example, 227.0.0.0/8.

Configuring the Data MDT Threshold Parameters

The PE router implementing data MDTs for a local multicast source must establish threshold limits for a multicast group and source. A multicast group can have more than one source of traffic.

To configure the data MDT threshold, include the threshold statement:

```
threshold {
    group group-address {
      source source-address {
        rate threshold-rate;
      }
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim mdt]

- [edit routing-instances *routing-instance-name* protocols pim mdt]

The group address is the multicast group address to which the threshold limits apply. This could be a well-known address for a certain type of multicast traffic.

The source address is the unicast address of the source of multicast information. It can be a source locally attached to or reached through the PE router. A group can have more than one source.

The group and source addresses 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, where prefix and more explicit address forms are used for the same source or group address, are not supported. For examples of supported and unsupported configurations, see "Examples: Configuring Data MDTs" on page 124.

The rate is the threshold applied to the multicast source to create a data MDT. The range is from 10 kilobits per second (Kbps), the default, to 1 gigabit per second (Gbps) (1,000,000 Kbps).

Configuring the Data MDT Limit

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.

To configure the data MDT limit, include the tunnel-limit statement:

tunnel-limit *limit*;

You can include this statement at the following hierarchy levels:

- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim mdt]
- [edit routing-instances *routing-instance-name* protocols pim 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.

Data MDTs and Tunnel Services PIC Limits

When configuring multicast over VPNs according to Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs*, each Tunnel Services Physical Interface Card (PIC) supports 512 multicast tunnel (mt-) interfaces. Configuring a router to allow more than 512 multicast tunnels requires another Tunnel Services PIC. Both default and data MDTs contribute to this total.

There are typically two default multicast tunnels (one for encapsulation and the other for de-encapsulation). If a router with a single Tunnel Services PIC creates more than 512 default and data MDTs, no traffic will flow for multicast tunnels created in excess of 512.

For example, configuring a router to allow 500 data MDTs requires only a single Tunnel Services PIC (500 + 2 – 502). However, configuring a router to allow 1000 data MDTs requires *two* Tunnel Services PICs (1000 + 2 = 1002). Up to 1024 multicast tunnels are supported with two Tunnel Services PICs.

For more information about Tunnel Services PICs and multicast tunnels, see "Tunnel Services PICs and Multicast" on page 216.

Examples: Configuring Data MDTs

This section describes the following examples for configuring data MDTs:

Configuring Data MDTs with Explicit Addresses

Configure routing instance VPN-A on a PE router to use tunnel identifiers taken from the 227.0.0.0/8 multicast address range. Create a data MDT when traffic for the multicast group 224.4.4.4 from local source 10.10.20.43 exceeds a threshold rate of 10 Kbps. Only 10 tunnels can be in use for this routing instance at any one time:

```
[edit routing-instances VPN-A protocols pim]
mdt {
    group-range 227.0.0.0/8;
    threshold {
        group 224.4.4.4 {
            source 10.10.20.43 {
                rate 10;
            }
        }
    }
    tunnel-limit 10;
}
```

No tunnels are created if 10 tunnels already exist for this routing instance on the PE router. Tunnels are deleted when the rate of traffic from the source falls below 10 Kbps, as determined by the normal, 60-second multicast statistics-collection cycle.

Configuring Data MDTs with Prefixes

Configure routing instance VPN-A on a PE router to use tunnel identifiers taken from the 227.0.0.0/8 multicast address range. Create a data MDT when traffic for any multicast group matching the prefix 224.0.0.0/4 (224/4) from any local source matching the prefix 10.0.0.0/8 (10/8) exceeds a threshold rate of 10 Kbps. Only 10 tunnels can be in use for this routing instance at any one time:

```
[edit routing-instances VPN-A protocols pim]
mdt {
    group-range 227.0.0.0/8;
    threshold {
```

```
            group 224.0.0.0/4 {
                source 10.0.0.0/8 {
                    rate 10;
                }
            }
        }
        tunnel-limit 10;
    }
```

No tunnels are created if 10 tunnels already exist for this routing instance on the PE router. Tunnels are deleted when the rate of traffic from the source falls below 10 Kbps, as determined by the normal, 60-second multicast statistics-collection cycle.

Explicit and prefix address forms can be combined if they do not overlap:

```
group 224.0.0.0/4 {
    source 10.10.20.43 {
        rate 10;
    }
    source 10.10.30.0/24 {
        rate 20;
    }
}
```

However, overlapping configurations such as the following are not supported:

```
group 224.0.0.0/4 {
    source 0/0 {
        rate 100;                    /* every source at 100 kbps... */
    }
group 224.0.0.0/4 {
    source 10.10.20.43 {
        rate 10;                     /* ...but THIS source at 10 kbps */
    }
```

Displaying Data MDTs

To display the data MDTs have been created for a VPN (VPN-A) on a PE router, use the show pin mdt outgoing instance *VPN-name* command:

```
user@PE-router> show pim mdt outgoing instance VPN-A
Instance: PIM.VPN-A
Tunnel direction: Outgoing
Default group address: 239.1.1.1
Default tunnel interface: mt-3/2/0.32768

C-Group address   C-source address  P-group address   Data tunnel interface
224.1.1.1         10.10.20.43       226.1.1.0         mt-3/2/0.32769
224.1.1.2         10.10.30.27       226.1.1.1         mt-3/2/0.32770
```

To verify that a VPN's data MDTs have been created as specified by rate configuration parameters on a PE router, use the show multicast route extensive instance *VPN-name* command:

```
user@PE-router> show multicast route extensive instance VPN-A
Family: INET

Group: 224.1.1.1
    Source: 10.10.20.43
    Upstream interface: fe-3/0/2.0
    Downstream interface list:
        mt-3/2/0.32769
    Session Description: ST Multicast Groups
    Statistics: 2 kBps, 25 pps, 127559 packets
    Next-hop ID: 378
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0

Group: 224.1.1.2
    Source: 10.10.30.27
    Upstream interface: fe-3/0/2.0
    Downstream interface list:
        mt-3/2/0.32770
    Session Description: ST Multicast Groups
    Statistics: 4 kBps, 40 pps, 10149 packets
    Next-hop ID: 380
    Upstream protocol: PIM
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: 360 seconds
    Wrong incoming interface notifications: 0

Family: INET6
```

Chapter 19

Summary of MDT Configuration Statements

The following sections explain each data multicast distribution tree (MDT) configuration statement. The statements are organized alphabetically.

group

Syntax

```
group group-address {
    source source-address {
      rate threshold-rate;
    }
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* routing-instances *routing-instance-name*
    protocols pim mdt threshold],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold]

Release Information

Statement introduced before JUNOS Release 7.4.

Description

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.

Usage Guidelines

See "Configuring the Data MDT Threshold Parameters" on page 122.

Required Privilege Level

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

group-range

Syntax group-range *multicast-prefix*;

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

Release Information Statement introduced before JUNOS 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

Usage Guidelines See "Configuring the Data MDT Group Range" on page 122.

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

mdt

Syntax

```
mdt {
    group-range multicast-prefix;
    threshold {
        group group-address {
            source source-address {
                rate threshold-rate;
            }
        }
    tunnel-limit limit;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim]

Release Information Statement introduced before JUNOS Release 7.4.

Description Establish the group range for data MDTs and threshold limits for a multicast group and source. A multicast group can have more than one source of traffic.

Options The remaining statements are explained separately.

Usage Guidelines See "MDT Configuration Guidelines" on page 121.

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

rate

Syntax rate *threshold-rate*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address* source *source-address*],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address* source *source-address*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Rate threshold applied to a multicast source to create a data MDT.

Options *threshold-rate*—Rate in kilobytes per second (Kbps) to apply to source.
Range: 10 Kbps through 1 Gbps (1,000,000 Kbps)
Default: 10 Kbps

Usage Guidelines See “Configuring the Data MDT Threshold Parameters” on page 122.

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

source

Syntax

```
source source-address {
    rate threshold-rate;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address*],
[edit routing-instances *routing-instance-name* protocols pim mdt threshold group *group-address*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Unicast address or prefix of the source of multicast information.

Options *source-address*—Explicit unicast address of multicast source.

The remaining statement is explained separately.

Usage Guidelines See “Configuring the Data MDT Group Range” on page 122.

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

threshold

Syntax

```
threshold {
    group group-address {
        source source-address {
            rate threshold-rate;
        }
    }
}
```

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

Release Information Statement introduced before JUNOS Release 7.4.

Description Establish threshold limits for a multicast group and source. A multicast group can have more than one source of traffic.

Options The remaining statements are explained separately.

Usage Guidelines See "Configuring the Data MDT Threshold Parameters" on page 122.

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

tunnel-limit

Syntax tunnel-limit *limit*;

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

Release Information Statement introduced before JUNOS 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.)

Usage Guidelines See "Configuring the Data MDT Limit" on page 123.

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

Part 8
Multicast Routing Options

Chapter 20

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 joins upstream toward a source and data forwarding downstream. Scoping can relieve stress on scarce resources, such as bandwidth, and improve privacy or scaling properties.

This section discusses the following topics that provide information about configuring multicast scoping:

For multicast scoping configuration examples, see "Example: Configuring Scoping with the scope Statement" on page 135 and "Example: Configuring Scoping with the scope-policy Statement" on page 137.

Multicast Scoping Overview

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 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.

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 should allocate 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.

Configuring Multicast Scoping

You can configure multicast scoping with a scoping statement or with a scoping policy statement. To configure multicast address scoping with either option, include the multicast statement:

```
multicast {
    scope scope-name {
      interface [ interface-names ];
      prefix destination-prefix;
    }
    scope-policy policy-name;
}
```

For a list of the hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

NOTE: You cannot apply a scoping 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.

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

You cannot use the scope and scope-policy statements together (the configuration does not commit). The policy statement referenced by the scope-policy statement must be properly configured at the policy-options hierarchy level. For more information about configuring policy statements, see the *JUNOS Policy Framework Configuration Guide*.

If you configure multicast scoping with the scope statement, the names of the defined scopes, prefixes, and interfaces are displayed as part of the show multicast scope command output. If you configure multicast scoping with the scope-policy statement, only the name of the scope policy is displayed as part of the show multicast scope command output.

This section discusses the following topics, which provide information about the two ways to configuring multicast scoping:

Configuring Multicast Scoping with the scope Statement

To configure multicast scoping with the scope statement, specify a name for the scope, the set of router interfaces on which you are configuring scoping, and the scope's address range.

When you configure multicast scoping with the scope statement, 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.

For information about supported standards for multicast scoping, see "IP Multicast Standards" on page 20.

Example: Configuring Scoping with the scope Statement

This example configures multicast scoping with the scope statement, creating four scopes: local, organization, engineering, and marketing.

If you have a Tunnel Physical Interface Card (PIC) in your router and you configure a tunnel interface to use IP-IP encapsulation, you can configure the local scope. For more information about configuring tunnel interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Configure the organization scope on an IP-IP encapsulation tunnel interface and a SONET/SDH interface. Configure the engineering and marketing scopes on an IP-IP encapsulation tunnel interface and two SONET/SDH interfaces. The JUNOS software can scope any user-configurable IPv6 or IPv4 group.

```
[edit]
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;
            }
        }
    }
```

NOTE: Do not configure the same prefix under multiple scope statements. If multiple scope statements contain the same prefix, only the last scope statement is enforced. If you need to scope the same prefix on multiple interfaces, do not use a separate scope statement for each interface. List all interfaces under one scope statement instead.

If you configure multicast scoping with the scope statement, you cannot use the scope-policy statement on the same router and vice versa. Using both statements on the same router prevents you from committing the configuration. To verify that group scoping is in effect, use 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-2/1/0                        0
organization     239.192.0.0/14            gr-2/1/0 so-0/0/0               0
engineering      239.255.255.0/24          ip-2/1/0 so-0/0/1 so-0/0/2  0
marketing        239.255.254.0/24          gr-2/1/0 so-0/0/2 so-1/0/0  0
```

For more information, see the *JUNOS Routing Protocols Configuration Guide*.

Configuring Scoping with the scope-policy Statement

To configure multicast scoping with the scope-policy statement, specify a policy name for the scope. The referenced policy must be correctly configured and contain the set of router interfaces on which you are configuring scoping, and the scope's address range configured as a series of router filters.

Only the interface, route-filter, and prefix-list match conditions are supported for multicast scoping 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, scoping policies apply to all interfaces. The default action is accept.

For more information about configuring route filters and policies, see the *JUNOS Policy Framework Configuration Guide.*

In contrast to scoping with the scope statement, scoping with scope-policy does not automatically add the local scope at scope boundaries. You must explicitly configure the local scope boundaries when you use the scope-policy statement. The local scope limits the use of the multicast group 239.255.0.0/16 to an attached LAN.

For information about supported standards for multicast scoping, see "IP Multicast Standards" on page 20.

Example: Configuring Scoping with the scope-policy Statement

This example configures a scope-policy statement named allow-Auto-RP-on-backbone, allowing packets for Auto-RP groups 224.0.1.39/32 exact and 224.0.1.40/32 exact on backbone-facing interfaces, and rejecting all other addresses in the 224.0.1.0/24 or longer and 239.0.0.0/8 or longer address ranges.

First, configure the policy allow-Auto-RP-on-backbone at the [policy-options] hierarchy level:

```
[edit]
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;
        }
    }
}
```

By default, the scope policy applies to all interfaces. For more information about route filters, see the *JUNOS Policy Framework Configuration Guide*.

Then apply the scope policy allow-Auto-RP-on-backbone at the routing-options hierarchy level:

```
[edit]
routing-options {
    multicast {
        scope-policy allow-Auto-RP-on-backbone;
    }
}
```

If you configure multicast scoping with the scope-policy statement, you cannot use the scope statement on the same router and vice versa. Using both statements on the same router prevents you from committing the configuration. To verify that the scope policy is in effect, use the show multicast scope command:

```
user@host> show multicast scope
Scope policy: [ allow-Auto-RP-on-backbone ]
```

Chapter 21
Multicast Reverse Path Forwarding

You use multicast reverse path forwarding (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 should 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 "RPF Checks and the RPF Table" on page 14.

However, there are network router configurations where multicast packets that fail the RPF check *should* 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. For information on LSPs, see the *JUNOS MPLS Applications Configuration Guide*.

In cases such as these, you can configure policies on the PE router to decide which multicast groups and sources should be exempt from the default RPF check.

This chapter discusses the following topics that provide information about configuring multicast RPF policies:

For more information about policies, see the *JUNOS Policy Framework Configuration Guide*.

Configuring RPF Policies

You configure one or more multicast RPF policies to disable RPF checks for a particular multicast (S,G) pair. You usually disable RPF checks on egress routers of a point-to-multipoint LSP.

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).

NOTE: Be careful when disabling RPF checks on multicast traffic. If you disable RPF checks in some configurations, multicast loops can result.

To configure multicast RPF policies, include the rpf-check-policy statement with a correctly configured policy:

```
multicast {
    rpf-check-policy [ policy-names ];
}
```

For a list of the hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

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

Example: Configuring RPF Policies

This example configures an RPF check policy named disable-RPF-on-PE, disabling the RPF check on multicast packets for the configured (S,G) source-group pair. This policy will not perform RPF checks on packets arriving for group 228.0.0.0/8 or from source address 196.168.25.6.

First, configure the policy disable-RPF-on-PE at the [edit policy-options] hierarchy level:

```
[edit]
policy-options {
    policy-statement disable-RPF-on-PE {
      term first {
        from {
          route-filter 228.0.0.0/8 orlonger;
          source-address-filter 192.168.25.6/32 exact;
```

```
            }
            then {
                reject;
            }
        }
    }
}
```

For more information about route and source address filters, see the *JUNOS Policy Framework Configuration Guide*.

Then apply the policy disable-RPF-on-PE at the [edit routing-options] hierarchy level:

```
[edit]
routing-options {
    multicast {
        rpf-check-policy disable-RPF-on-PE;
    }
}
```

You can also configure each condition as a separate policy and reference both policies in the rpf-check-policy statement:

```
[edit]
policy-options {
    policy-statement disable-RPF-on-group {
        term first {
            from {
                route-filter 228.0.0.0/8 orlonger;
            }
            then {
                reject;
            }
        }
    }
}
    policy-statement disable-RPF-on-source {
        term first {
            from {
                source-address-filter 192.168.25.6/32 exact;
            }
            then {
                reject;
            }
        }
    }
}

[edit]
routing-options {
    multicast {
        rpf-check-policy [ disable-RPF-on-group disable-RPF-on-source ];
    }
}
```

This allows you to associate groups in one policy and source in the other.

Chapter 22

Source-Specific Multicast

Source-specific multicast (SSM) is a service model that identifies session traffic by both source and group address. SSM implemented in the JUNOS software 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 on 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) stack or configure SSM mapping from IGMPv1/IGMPv2 to IGMPv3. An IGMPv3 stack is the capability of a host operating system to use the IGMPv3 protocol. Most operating systems today use an IGMPv2 stack, but IGMPv3 is available for Windows XP, and upgrades are available on some 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.

For information about standards supported for source-specific multicast, see "IP Multicast Standards" on page 20.

This chapter discusses the following topics:

Source-Specific Multicast Groups Overview

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.

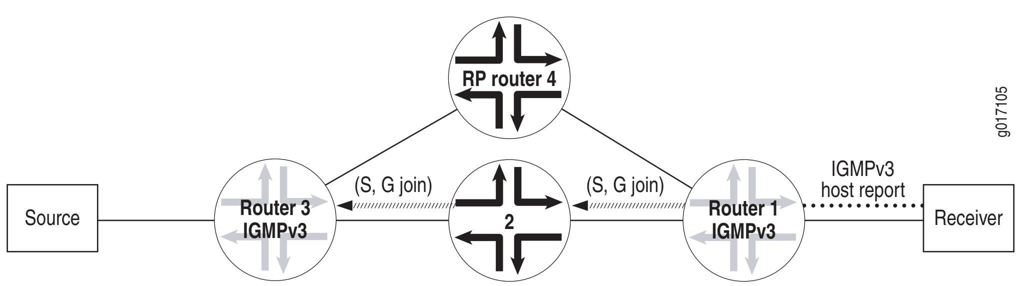
By default, the SSM group multicast address is limited to the IP address range 232.0.0.0 to 232.255.255.255. However, you can extend SSM operations into another Class D range by including the address statement at the [edit routing-options multicast ssm-groups] hierarchy level.

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).

Deploying SSM is easy. You need only 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 members interfaces, multicast packets are not forwarded. Source lists, supported in IGMPv3, are used in PIM SSM. Only sources that are specified send traffic to the SSM group.

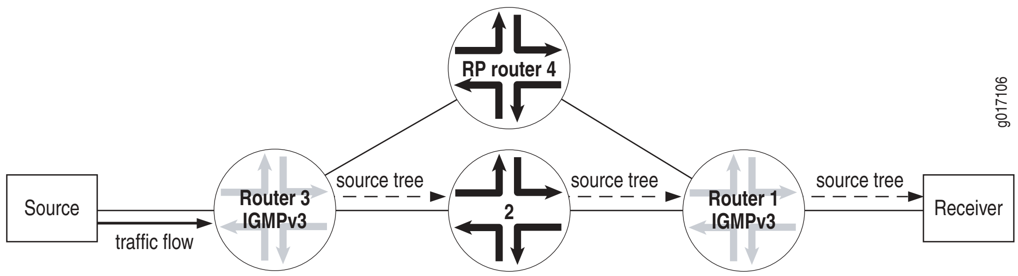
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3), announcing a desire to join group G and source S (see Figure 10). The directly connected PIM sparse-mode router, the receiver's designated router (DR), sends an (S,G) join message to its reverse-path forwarding (RPF) neighbor for the source. Notice in Figure 10 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

Figure 10: Receiver Announces Desire to Join Group G and Source S



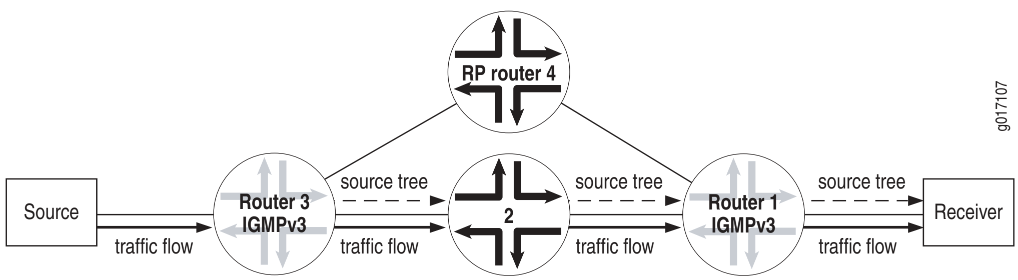
The (S,G) join message initiates the source tree, then builds it out hop by hop until it reaches the source. In Figure 11, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 11: Router 3 (Last-Hop Router) Joins the Source Tree



Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 12).

Figure 12: The (S,G) State Is Built Between the Source and the Receiver



To configure additional SSM groups, include the ssm-groups statement:

```
multicast {
    ssm-groups {
      address;
    }
}
```

For a list of the hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

Source-Specific Multicast Examples

This section discusses the following topics:

- Example: Configuring an SSM-Only Domain on page 146
- Example: Configuring PIM SSM on a Network on page 146
- Example: Configuring SSM Mapping on page 148

Example: Configuring an SSM-Only Domain

Deploying an SSM-only domain is much simpler than deploying an ASM domain; 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:

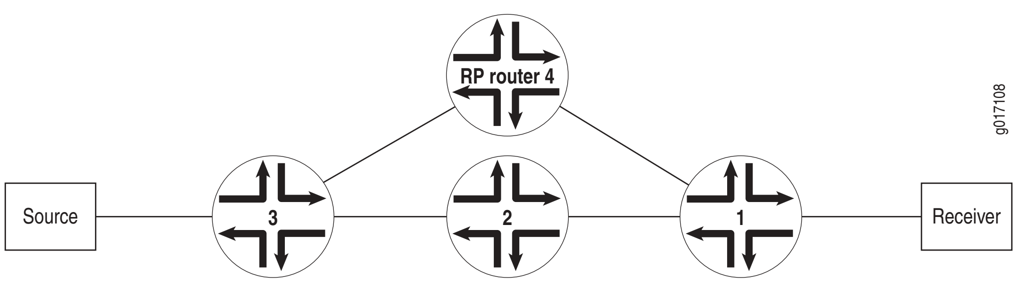
```
[edit]
protocols {
    pim {
      interface all {
        mode sparse;
          version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }

    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 13.

Figure 13: Network on Which to Configure PIM SSM



The configuration establishes IGMPv3 on all receiving host interfaces. You then can use the following show commands to verify the PIM SSM configuration:

- Issue the show igmp interface command to display IGMP interfaces, configurable parameters, and IGMP version.
- Issue the show pim join extensive command to display the PIM state.

This example discusses the following topics that provide information about configuring and verifying operation of PIM SSM:

Enabling IGMPv3 on all Host-Facing Interfaces

To enable IGMPv3 on all host-facing interfaces, include the version 3 statement under the interface all statement at the [edit protocols igmp] hierarchy level:

```
[edit protocols igmp]
interface all {
    version 3;
}
interface fxp0.0 {
    disable;
}
```

NOTE: When you configure IGMPv3 on a router, hosts on interfaces configured with IGMPv2 cannot join the source tree.

Displaying the IGMP State

To show IGMP information about the interfaces on Router 1, use the show igmp interface command:

```
user@router1> show igmp interface
Interface      State    Querier      Timeout Version Groups
fxp1.0         Up                          0       3      0
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
```

Displaying the PIM State

To show the PIM state on Router 2 and Router 3 (the upstream routers), use the show pim join extensive command:

```
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
```

To show the PIM state on Router 1 (the router connected to the receiver), use the show pim join extensive command:

```
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 all hosts to support IGMPv3. SSM mapping translates IGMPv1 or IGMPv2 membership reports to an IGMPv3 report. This allows hosts running IGMPv1/IGMPv2 to participate in SSM until the hosts transition to IGMPv3.

There are three steps to configuring SSM mapping. First, you create a policy to match the group addresses 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. 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). Finally, you apply the SSM map to one or more IGMP (for IPv4) or MLD (for IPv6) interfaces.

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.

- Creating the SSM Policy on page 149
- Defining the SSM Map on page 149
- Applying SSM Mapping to Interfaces on page 149
- Displaying the SSM Maps on page 150

Creating the SSM Policy

This example creates an SSM policy called 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.

```
[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.

Defining the SSM Map

This example defines 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.

```
[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.

Applying SSM Mapping to Interfaces

You should apply SSM maps for IPv4-to-IGMP interfaces and SSM maps for IPv6-to-MLD interfaces:

```
[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;
    }
}
```

Displaying the SSM Maps

To show the SSM maps that are applied to an interface, use the show igmp interface or the show mld interface command:

```
user@router1> 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@router1> 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
```

Chapter 23

Flow Maps

Multicast flow maps enable you to manage a subset of multicast forwarding table entries. For example, you can specify that certain forwarding cache entries be permanent or have a different timeout value than other multicast flows that are not associated with the flow map policy.

The three steps to create a flow map are:

1. Create a policy that contains source and group addresses of multicast flows.
2. Define a flow map that references the flow map policy.
3. Define the properties that the flow map applies to the flows that are selected by the flow map policy.

Creating a Flow Map

This section includes the following topics:

Creating the Flow Map Policy

The following example creates a flow map policy called policyForFlow1. In this example, the policy statement matches the source address using the source-address-filter statement and matches the group address using the prefix list.

```
[edit policy-options]
prefix-list permanentEntries1 {
    232.1.1.1/32;
}
policy-statement policyForFlow1 {
    from {
        source-address-filter 11.11.11.11/32 exact;
        prefix-list permanentEntries1;
    }
    then accept;
}
```

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

For additional information about creating policy statements, see the *JUNOS Policy Framework Configuration Guide*.

Defining the Flow Map

This example defines a flow map called flowMap1 with permanent forwarding entries (that is, the entries never time out).

```
[edit routing-options]
multicast {
    flow-map flowMap1 {
        policy policyForFlow1;
        forwarding-cache {
            timeout never;
        }
}
```

Defining Flow Map Properties

Flow maps enable you to configure multicast forwarding cache timeout values for entries defined by the flow map policy. The timeout value can range from 1 to 720 minutes, or you can set the value to never, making the entries permanent. For additional information about how to configure forwarding cache properties, see “Configuring General Multicast Forwarding Cache Properties” on page 153.

Displaying the Flow Maps

To display configured flow maps, the policies that they use, and their configuration settings, use the show multicast flow-map command:

```
user@router1> show multicast flow-map
Instance: master
Name            Policy                  Cache timeout
map1            policy1                    60 seconds
map2            policy1                         never
```

For additional information about this show command, including descriptions of the fields, see the *JUNOS Routing Protocols and Policies Command Reference*.

Chapter 24

Multicast Forwarding Cache Properties

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.

This section discusses the following topics, which provide information about configuring multicast forwarding cache properties:

Configuring General Multicast Forwarding Cache Properties

To configure multicast forwarding cache limits and timeout values, include the forwarding-cache statement at the [edit routing-options multicast] hierarchy level:

```
multicast {
    forwarding-cache {
        threshold suppress value <reuse value>;
        timeout minutes;
    }
}
```

You can configure this statement at the following hierarchy levels:

- [edit logical-routers logical-router-name routing-instances routing-instance-name routing-options multicast]
- [edit logical-routers logical-router-name routing-options multicast]
- [edit routing-instances routing-instance-name routing-options multicast]
- [edit routing-options multicast]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

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

You can specify a value for the threshold 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 also specify a timeout value for all multicast forwarding cache entries. The range for the timeout value is 1 through 720 minutes.

For information about supported standards for multicast scoping, see "IP Multicast Standards" on page 20.

Configuring Multicast Forwarding Cache Properties for Flow Maps

The forwarding-cache statement 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 1 through 720 minutes.

You can use longer timeouts or permanent forwarding cache entries in source redundancy scenarios to decrease the time delay inherent in switching from one source to another. With source redundancy, only one source (for example, s1) actively sends traffic at any given time, but the sources can switch. This means that the forwarding state for another source (for example, s2) can time out.

Even though the Routing Engine has the corresponding PIM states set up, when the first (s2,g) packet arrives on each router after a switchover from source s1 to source s2, the router must reinstall the forwarding path (s2,g) if the path has timed out. If many data streams exist, the switchover can take considerable time. Using longer timeouts or permanent cache entries helps reduce the inherent switchover delay.

To configure the multicast forwarding cache timeout for a specified flow map, include the forwarding-cache statement:

```
multicast {
    flow-map flow-map-name {
        forwarding-cache {
            timeout (never | minutes);
        }
    }
}
```

You can configure this statement at the following hierarchy levels:

- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*]
- [edit logical-routers *logical-router-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*]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

NOTE: The permanent forwarding state must exist on all routers in the path for fast source switchover to function properly.

For information about configuring multicast flow maps, see "Creating a Flow Map" on page 151.

Examples: Configuring Multicast Forwarding Cache Properties

Depending on the hierarchy level at which you modify the multicast forwarding cache properties, you can specify a threshold value, a timeout value, or both.

Configuring Forwarding Cache Properties at the Multicast Level

Configure multicast forwarding cache limits, establishing a suppression threshold of 10, a reuse threshold of 7, and no timeout value:

```
routing-options {
    multicast {
        forwarding-cache {
            threshold suppress 10 reuse 7;
        }
    }
}
```

Only 10 multicast forwarding cache entries are created. No new multicast forwarding cache entries are created until the number of multicast forwarding cache entries falls to 7.

Configure multicast forwarding cache limits, establishing a suppression threshold of 20, no reuse threshold, and no timeout value:

```
routing-options {
    multicast {
        forwarding-cache {
            threshold suppress 20;
        }
    }
}
```

Only 20 multicast forwarding cache entries are created. A new multicast forwarding cache entry is created when the number of multicast forwarding cache entries falls to 19.

Configure the multicast forwarding cache, establishing a timeout of 60 minutes, no suppression, and no reuse threshold:

```
routing-options {
    multicast {
        forwarding-cache {
            timeout 60;
        }
    }
}
```

Multicast forwarding cache entries are created as necessary. Forwarding cache entries are deleted after being idle for 60 minutes (1 hour).

Configuring Forwarding Cache Properties at the Flow Map Level

Configure the flow map forwarding cache, establishing a timeout of 120 minutes:

```
routing-options {
    multicast {
        flow-map flowMap1
            forwarding-cache {
                timeout 120;
            }
        }
    }
}
```

Multicast forwarding cache entries associated with flow map *flowMap1* are deleted after being idle for 120 minutes (2 hours).

Configure the flow map forwarding cache, establishing permanent forwarding cache entries:

```
routing-options {
    multicast {
        flow-map flowMap2
            forwarding-cache {
                timeout never;
            }
        }
    }
}
```

Multicast forwarding cache entries associated with flow map *flowMap2* are permanent (that is, they never time out).

Displaying the Cache Timeout

To display the cache timeout value for each multicast route, use the show multicast route extensive command:

```
user@router1> 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: never
Wrong incoming interface notifications: 0
```

For additional information about this show command, including descriptions of the fields, see the *JUNOS Routing Protocols and Policies Command Reference*.

Chapter 25

Summary of Multicast Routing Options Configuration Statements

The following sections explain each of the multicast routing options configuration statements. The statements are organized alphabetically.

flow-map

Syntax

```
flow-map flow-map-name {
    policy policy-name;
    forwarding-cache {
        timeout (never | minutes);
    }
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced in JUNOS Release 8.2.

Description Configure multicast flow maps.

Options The remaining statements are explained separately.

Usage Guidelines See “Creating a Flow Map” on page 151.

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

forwarding-cache

See the following sections:

- forwarding-cache (flow-map) on page 160
- forwarding-cache (multicast) on page 160

forwarding-cache (flow-map)

Syntax

```
forwarding-cache {
    timeout (never | minutes);
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit logical-routers *logical-router-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 Release 8.2.

Description Configure multicast forwarding cache properties for the flow-map.

Options The statement is explained separately.

Usage Guidelines See "Configuring Multicast Forwarding Cache Properties for Flow Maps" on page 154.

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 {
    threshold suppress value <reuse value>;
    timeout minutes;
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure multicast forwarding cache properties. These properties include threshold suppression limits and timeout values.

Options The remaining statements are explained separately.

Usage Guidelines See "Configuring General Multicast Forwarding Cache Properties" on page 153.

Required Privilege Level

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

interface

Syntax interface [*interface-names*];

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast scope *scope-name*],
[edit logical-routers *logical-router-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 Release 7.4.

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. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Usage Guidelines See “Configuring Multicast Scoping with the scope Statement” on page 135.

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

multicast

Syntax

```
multicast {
    flow-map flow-map-name {
        forwarding-cache {
            timeout (never | minutes);
            }
        policy policy-name;
        }
    }
    forwarding-cache {
        threshold suppress value <reuse value>;
        timeout minutes
    }
    rpf-check-policy [ policy-names ];
    scope scope-name {
        interface [ interface-names ];
        prefix destination-prefix;
    }
    scope-policy policy-name;
    ssm-groups {
        address;
    }
    ssm-map ssm-map-name {
        policy policy-name;
        source addresses;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options],
[edit logical-routers *logical-router-name* routing-options],
[edit routing-instances *routing-instance-name* routing-options],
[edit routing-options]

NOTE: You cannot apply a scoping 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 Release 7.4.

Description Configure multicast routing options properties.

Options The remaining statements are explained separately.

Usage Guidelines See "Configuring Multicast Scoping" on page 134, "Source-Specific Multicast Groups Overview" on page 144, "Configuring General Multicast Forwarding Cache Properties" on page 153, and "Configuring Multicast Forwarding Cache Properties for Flow Maps" on page 154.

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

policy

See the following sections:

- policy (flow-map) on page 162
- policy (ssm-map) on page 163

policy (flow-map)

Syntax policy *policy-name*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name*],
[edit logical-routers *logical-router-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 Release 8.2.

Description Configure a flow map policy.

Options *policy-name*—Name of the policy for flow mapping.

Usage Guidelines See "Creating the Flow Map Policy" on page 151.

Required Privilege Level routing—To view this statement in the configuration.

policy (ssm-map)

Syntax policy *policy-name*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast ssm-map *ssm-map-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description Apply a policy to an SSM map.

Options *policy-name*—Name of the policy for SSM mapping.

Usage Guidelines See “Example: Configuring SSM Mapping” on page 148.

Required Privilege Level routing—To view this statement in the configuration.

prefix

Syntax prefix *destination-prefix*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast scope *scope-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure the prefix for multicast scopes.

Options *destination-prefix*—Address range for the multicast scope.

Usage Guidelines See "Configuring Multicast Scoping with the scope Statement" on page 135.

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

rpf-check-policy

Syntax rpf-check-policy [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced in JUNOS Release 8.1.

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.

Usage Guidelines See "Configuring RPF Policies" on page 140.

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

scope

Syntax

```
scope scope-name {
    interface [ interface-names ];
    prefix destination-prefix;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure multicast scoping.

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

The remaining statements are explained separately.

Usage Guidelines See "Configuring Multicast Scoping with the scope Statement" on page 135.

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

scope-policy

Syntax scope-policy *policy-name*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-options multicast]

NOTE: You can configure a scoping policy at these two hierarchy levels only. You cannot apply a scoping 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.

Release Information Statement introduced before JUNOS Release 7.4.

Description Apply a policy for scoping. The policy must be correctly configured.

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

Usage Guidelines See "Configuring Scoping with the scope-policy Statement" on page 136.

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

source

Syntax source *addresses*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast ssm-map *ssm-map-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description Specify IPv4 or IPv6 source addresses for an SSM map.

Options *addresses*—IPv4 or IPv6 source addresses.

Usage Guidelines See "Example: Configuring SSM Mapping" on page 148.

Required Privilege Level routing—To view this statement in the configuration.

ssm-groups

Syntax

```
ssm-groups {
    address;
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure additional source-specific multicast (SSM) groups.

Options *address*—Address range of the additional SSM group.

Usage Guidelines See "Source-Specific Multicast Groups Overview" on page 144.

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

ssm-map

Syntax

```
ssm-map ssm-map-name {
    policy policy-name;
    source addresses;
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast],
[edit logical-routers *logical-router-name* routing-options multicast],
[edit routing-instances *routing-instance-name* routing-options multicast],
[edit routing-options multicast]

Release Information Statement introduced in JUNOS Release 7.4.

Description Configure SSM mapping.

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

The remaining statements are explained separately.

Usage Guidelines See "Example: Configuring SSM Mapping" on page 148.

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

threshold

Syntax threshold suppress *value* <reuse *value*>;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit logical-routers *logical-router-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 before JUNOS Release 7.4.

Description Configure the suppression and reuse thresholds for multicast forwarding cache limits.

Options suppress *value*—Value 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

reuse *value*—Value to begin creating new multicast forwarding cache entries. This value is optional. If configured, this number should be less than the suppress value.
Range: 1 through 200,000

Usage Guidelines See "Examples: Configuring Multicast Forwarding Cache Properties" on page 155.

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

timeout

See the following sections:

- timeout (multicast forwarding-cache) on page 169
- timeout (flow-map forwarding-cache) on page 169

timeout (multicast forwarding-cache)

Syntax timeout *minutes*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast forwarding-cache],
[edit logical-routers *logical-router-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 Release 8.2.

Description Configure the timeout value for multicast forwarding cache entries.

Options timeout *minutes*—Length of time, in minutes, that the forwarding cache limit remains active.
Range: 1 through 720

Usage Guidelines See "Configuring General Multicast Forwarding Cache Properties" on page 153.

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

timeout (flow-map forwarding-cache)

Syntax timeout (never | *minutes)*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name* forwarding-cache],
[edit logical-routers *logical-router-name* routing-options multicast flow-map *flow-map-name* forwarding-cache],
[edit routing-instances *routing-instance-name* routing-options multicast flow-map *flow-map-name* forwarding-cache],
[edit routing-options multicast flow-map *flow-map-name* forwarding-cache]

Release Information Statement introduced in JUNOS Release 8.2.

Description Configure the timeout value for multicast forwarding cache entries associated with the flow map.

Options timeout *minutes*—Length of time, in minutes, that the forwarding cache entry remains active.
Range: 1 through 720

never—Specifies that the forwarding cache entry always remains active

Usage Guidelines See “Configuring Multicast Forwarding Cache Properties for Flow Maps” on page 154.

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

Part 9

DVMRP

Chapter 26

DVMRP Overview

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.

For information about standards supported for DVMRP, see “IP Multicast Standards” on page 20.

Chapter 27

DVMRP Configuration Guidelines

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 name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

By default, DVMRP is disabled.

This chapter describes the following tasks for configuring DVMRP:

Minimum DVMRP Configuration

To enable DVMRP on an interface, include at least the following statements in the configuration. All other DVMRP configuration statements are optional.

```
routing-options {
    interface-routes {
        rib-group group-name1;
    }
    rib-groups {
        group-name1 {
            import-rib [ inet.0 inet.2 ];
        }
        group-name2 {
            import-rib inet.2;
            export-rib inet.2;
        }
    }
}
protocols {
    dvmrp {
        rib-group group-name2;
        interface interface-name;
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit]
- [edit logical-routers *logical-router-name*]

The port 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). All interfaces can be configured with a metric specifying cost for receiving packets on a given port. The default metric is 1.

Creating Routing Tables for DVMRP 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.

To create the necessary routing tables and routing table groups for DVMRP, include the following statements:

```
routing-options {
    interface-routes {
        rib-group group-name1;
    }
    rib-groups {
        group-name1 {
            import-rib [ inet.0 inet.2 ];
        }
        group-name2 {
            import-rib inet.2;
            export-rib inet.2;
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit]
- [edit logical-routers *logical-router-name*]

To associate the routing tables with DVMRP, include the rib-group statement at the [edit protocols dvmrp] hierarchy level, as described in "Enabling DVMRP" on page 177.

Enabling DVMRP

To enable DVMRP on the router, include the following statements:

```
dvmrp {
    interface interface-name;
    rib-group group-name;
    traceoptions;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]

The rib-group statement selects a routing table group. DVMRP exports routes from this group and imports routes to this group. The rib-group statement associates with DVMRP the routing table group that imports and exports routes into the specified routing table group. This is a group you defined with the rib-groups statement at the [edit routing-options] hierarchy level.

You must specify the interface or interfaces on which to enable DVMRP. Specify the full interface name, including the physical and logical address components. To configure all interfaces, specify the interface name all. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

NOTE: If you have configured Protocol Independent Multicast (PIM) on the interface, you can configure DVMRP in unicast-routing mode only. You cannot configure PIM and DVMRP in forwarding mode at the same time.

Modifying the DVMRP Hold-Time Period

The DVMRP hold-time period is the amount of time that a neighbor should consider the sending router (this router) to be operative (up). The default hold-time period is 35 seconds.

To modify the hold-time value for the local router, include the hold-time statement:

```
hold-time seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

The hold-time period can range from 1 through 255 seconds.

Modifying the Metric Value

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.

By default, a metric value of 1 is associated with each DVMRP route. To modify the metric value, include the metric statement:

```
metric metric;
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

The metric can range from 1 through 31.

Disabling DVMRP on an Interface

To disable DVMRP on an interface, include the disable statement:

```
disable;
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

Configuring DVMRP Routing Policy

All routing protocols use the routing table to store the routes that they learn and to determine which routes they should advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in and retrieve from the routing table.

When configuring DVMRP routing policy, you can apply routing policies. To do this, include the import and export statements at the [edit protocols dvmrp] or [edit logical-routers *logical-router-name* protocols dvmrp] hierarchy level.

To apply policies to routes imported into the routing table from DVMRP, include the import statement, listing the names of one or more policy filters to be evaluated. 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.

```
import [ policy-names ];
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp]
- [edit logical-routers *logical-router-name* protocols dvmrp]

To apply policies to routes exported from the routing table into DVMRP, include the export statement, listing the names of one or more policies to be evaluated. 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.

```
export [ policy-names ];
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp]
- [edit logical-routers *logical-router-name* protocols dvmrp]

Configuring DVMRP Routing Modes

You can configure DVMRP for either forwarding or unicast routing mode. In forwarding mode, DVMRP operates its protocol normally (for example, it does the routing as well as multicast data forwarding). In unicast routing mode, you can use DVMRP for unicast routing only; to forward multicast data, enable PIM on that interface. To configure the mode, include the mode statement.

To configure DVMRP for multicast forwarding, include the mode forwarding statement:

```
mode forwarding;
```

To configure DVMRP for unicast routing, include the mode unicast-routing statement:

```
mode unicast-routing;
```

You can include these statements at the following hierarchy levels:

- [edit protocols dvmrp interface *interface-name*]
- [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*]

The default mode is forwarding.

Tracing DVMRP Protocol Traffic

To trace DVMRP protocol traffic, you can specify options in the global traceoptions statement at the [edit routing-options] or [edit logical-routers *logical-router-name* routing-options] hierarchy level. Options applied at the routing options level trace all packets, and options applied at the protocol level trace only DVMRP traffic.

You can specify DVMRP-specific options by including the traceoptions statement:

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols dvmrp]
- [edit logical-routers *logical-router-name* protocols dvmrp]

You can specify the following DVMRP-specific options in the DVMRP traceoptions statement:

- all—Trace everything.
- general—Trace general events.
- graft—Trace graft messages.
- neighbor—Trace neighbor probe messages.
- normal—Trace normal events.
- packets—Trace all DVMRP packets.
- poison—Trace poison-route-reverse packets.
- policy—Trace policy processing.
- probe—Trace probe packets.
- prune—Trace prune messages.
- report—Trace DVMRP route report packets.
- route—Trace routing information.
- state—Trace state transitions.
- task—Trace routing protocol task processing.
- timer—Trace routing protocol timer processing.

For general information about tracing and global tracing options, see the *JUNOS Routing Protocols Configuration Guide*.

Configuration Examples

This section contains the following DVMRP configuration examples:

Example: Tracing DVMRP Protocol Traffic

Trace only unusual or abnormal operations to the file routing-log, and trace detailed information about all DVMRP messages to the file dvmrp-log:

```
[edit]
routing-options {
    traceoptions {
        file routing-log;
    }
}
protocols {
    dvmrp {
        traceoptions {
            file dvmrp-log;
            flag packets;
        }
        interface so-0/0/0;
    }
}
```

Example: Configuring DVMRP

Configure DVMRP on the router:

```
[edit]
routing-options {
    interface-routes {
        rib-group ifrg;
    }
    rib-groups {
        ifrg {
            import-rib [ inet.0 inet.2 ];
        }
        dvmrp-rib {
            import-rib inet.2;
            export-rib inet.2;
        }
    }
}
protocols {
    sap;
    dvmrp {
        rib-group dvmrp-rib;
        traceoptions {
            flag normal;
            flag state;
        }
        interface ip-f/p/0.0 {
            hold-time 130;
        }
    }
}
```

Example: Configuring DVMRP to Announce Unicast Routes

In this example, DVMRP is used to announce unicast routes used solely for multicast reverse-path forwarding (RPF). Include the mode unicast-routing statement at the [edit protocols dvmrp interface] hierarchy level. Redistribute static routes by including the static statement at the [edit routing-options] hierarchy level to export the routes to all DVMRP neighbors.

```
[edit]
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;
        }
    }
}
protocols {
    dvmrp {
        rib-group inet dvmrp-rg;
        export dvmrp-export;
        interface all {
            mode unicast-routing;
        }
    }
    pim {
        rib-group inet pim-rg;
        interface all;
    }
}
policy-options {
    policy-statement dvmrp-export {
        term 10 {
            from {
                protocol static;
                route-filter 0.0.0.0/0 exact;
            }
            then accept;
        }
    }
}
```

Chapter 28

Summary of DVMRP Configuration Statements

The following sections explain each of the Distance Vector Multicast Routing Protocol (DVMRP) configuration statements. The statements are organized alphabetically.

disable

Syntax disable;

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*],
[edit protocols dvmrp interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Explicitly disable DVMRP on an interface.

Usage Guidelines See “Disabling DVMRP on an Interface” on page 179.

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

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 name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols],
[edit protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable DVMRP on the router.

Default DVMRP is disabled on the router.

Options The statements are explained separately.

Usage Guidelines See “Enabling DVMRP” on page 177.

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

export

Syntax export [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp rib-group *group-name*],
[edit protocols dvmrp rib-group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Apply one or more policies to routes being exported from the routing table into DVMRP.

Options *policy-names*—Name of one or more policies.

Usage Guidelines See “Configuring DVMRP Routing Policy” on page 179.

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

See Also import on page 187

hold-time

Syntax hold-time *seconds*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*],
[edit protocols dvmrp interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description How long a neighbor should consider the sending router (this router) to be operative (up).

Options *seconds*—Hold time.
Range: 1 through 255
Default: 35 seconds

Usage Guidelines See "Modifying the DVMRP Hold-Time Period" on page 178.

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

import

Syntax import [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp rib-group *group-name*],
[edit protocols dvmrp rib-group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Apply one or more policies to routes being imported into the routing table from DVMRP.

Options *policy-names*—Name of one or more policies.

Usage Guidelines See "Configuring DVMRP Routing Policy" on page 179.

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

See Also export on page 186

interface

Syntax
interface *interface-name* {
    disable;
    hold-time *seconds*;
    metric *metric*;
}

Hierarchy Level
[edit logical-routers *logical-router-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information
Statement introduced before JUNOS 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. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

The remaining statements are explained separately.

Usage Guidelines
See "Enabling DVMRP" on page 177.

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

metric

Syntax
metric *metric*;

Hierarchy Level
[edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*],
[edit protocols dvmrp interface *interface-name*]

Release Information
Statement introduced before JUNOS Release 7.4.

Description
Define the DVMRP metric value.

Options
metric—Metric value.
    Range: 1 through 31
    Default: 1

Usage Guidelines
See "Modifying the Metric Value" on page 178.

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

mode

Syntax mode (forwarding | unicast-routing)

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp interface *interface-name*],
[edit protocols dvmrp interface *interface-name*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure DVMRP multicast traffic forwarding or unicast routing.

Options forwarding—DVMRP does unicast routing as well as multicast data forwarding.

unicast-routing—DVMRP does the routing only. To forward multicast data, you must configure Protocol Independent Multicast (PIM) on the interface.

Usage Guidelines See “Configuring DVMRP Routing Modes” on page 180.

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

rib-group

Syntax rib-group *group-name*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information Statement introduced before JUNOS 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.

Usage Guidelines See “Enabling DVMRP” on page 177.

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

traceoptions

Syntax

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols dvmrp],
[edit protocols dvmrp]

Release Information Statement introduced before JUNOS 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 *name*—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 also must specify a maximum file size with the size option.

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 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.

Usage Guidelines See “Tracing DVMRP Protocol Traffic” on page 180.

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.

Part 10
PIM

Chapter 29
PIM Overview

Protocol Independent Multicast (PIM) is used for efficiently 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. The JUNOS software supports sparse mode, dense mode, and sparse-dense mode.

For information about standards supported for PIM, see “IP Multicast Standards” on page 20.

PIM Modes

Because the mode you choose determines the PIM configuration properties, you first must decide whether PIM operates in 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 router unless it 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 joins and prunes are common.

Unlike sparse 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.

For more information about how the PIM modes operate, see:

- PIM Sparse Mode on page 196
- PIM Dense Mode on page 213
- PIM Sparse-Dense Mode on page 214
- RP Mapping with Anycast RP on page 215
- Multicast over Layer 3 VPNs on page 215
- Tunnel Services PICs and Multicast on page 216
- Filtering Multicast Messages on page 217
- Embedded RP for IPv6 Multicast on page 219

For more information about mode-dependent configurations, see:

- Configuring PIM Dense Mode Properties on page 231
- Configuring PIM Sparse Mode Properties on page 231
- Configuring Sparse-Dense Mode Properties on page 250

PIM Sparse Mode

A 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 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's 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's DR encapsulates the packets in a PIM register message and forwards it 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 simply 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 gets the first packet from the RPT, the DR sends a PIM join message toward the source's DR to start building an SPT back to the source. When the source's 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 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 knows of the active source's existence and no longer needs to receive advertisement of the source (which utilizes resources).

This section contains more information about the routers and PIM sparse-mode functions briefly described above:

Designated Router

In a PIM sparse-mode domain, there are two types of designated routers to consider:

- The receiver's DR sends PIM join and PIM prune messages from the receiver network toward the RP.
- The source's DR sends PIM register messages from the source network to the RP.

Regardless of whether it is the receiver's DR or the source's DR, a DR is selected from other routers in a network by the exchange of IP addresses. Neighboring PIM sparse-mode routers multicast periodic PIM hello messages to each other every 30 seconds (the default). 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.

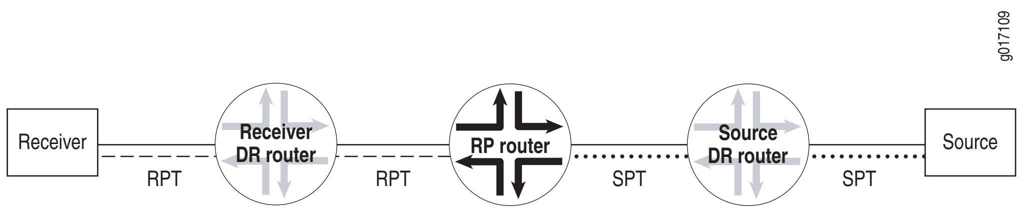
If a DR fails, a new one is selected using the same process of comparing IP addresses.

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 get to 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 SPT. As shown in Figure 14, the RP router is upstream from the receiver and thus forms one end of the RPT.

Figure 14: The RP 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 on page 199
- Anycast RP on page 200
- Auto-RP on page 200
- Bootstrap Router on page 201

Static Configuration

You can configure a static RP configuration that is very 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.

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.

Anycast RP

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.

Having a single active 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.

When anycast RP is configured, the shared address is used in the RP-to-group mapping. This allows multicast groups to have multiple active RPs in a PIM domain. However, the RPs must use some protocol to synchronize the active source information so that the active RP for each group is known to all RPs.

There are two methods for RP active source synchronization in anycast RP, one using the Multicast Source Discovery Protocol (MSDP) and the other using PIM itself.

When MSDP is used with PIM sparse mode, anycast RP provides a faster failover rate than auto-RP or a bootstrap router. However, MSDP only works for IPv4. When PIM alone is used for anycast RP, the solution works for both IPv4 and IPv6.

For more information about configuring static RPs, see “Configuring Static RPs” on page 238. For more information about configuring anycast RP, see “Configuring Auto-RP” on page 241 and “Example: Configuring Anycast RP” on page 258.

Auto-RP

You can configure a more dynamic way of assigning 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 stops operating, one of the other preconfigured routers takes over the RP functions. This capability is controlled by the auto-RP mapping agent.

For more information, see “Configuring Auto-RP” on page 241.

Bootstrap Router

To determine which router is the 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's bootstrap router originates 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.

For more information, see "Configuring Bootstrap Properties" on page 238.

Building an RPT Between RP and Receivers

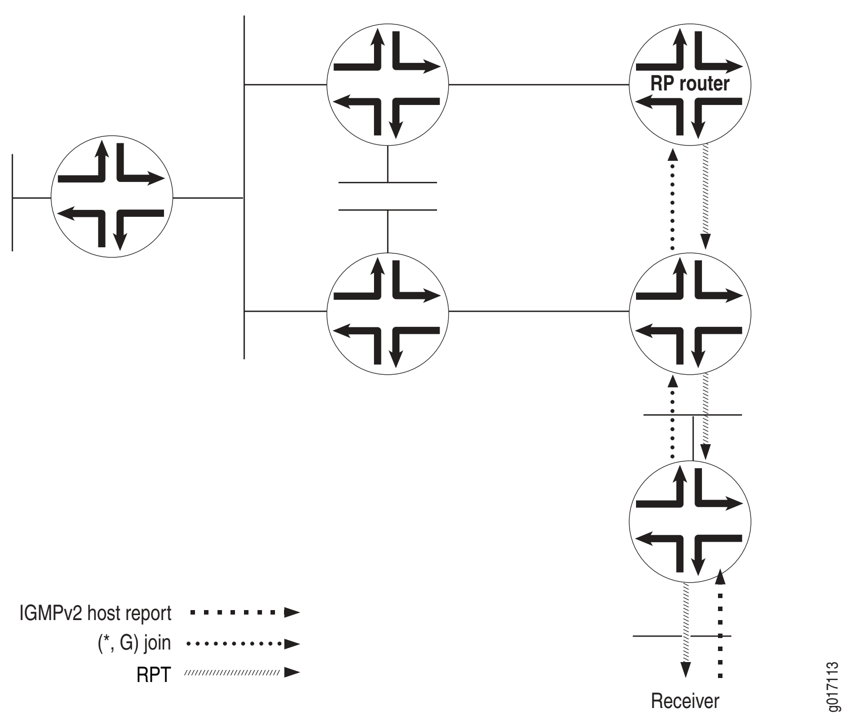
The RPT is the path between the RP and receivers (hosts) in a multicast group (see Figure 15). The RPT is built by means of a PIM join message from a receiver's DR:

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 15: Building an RPT Between RP and Receiver



PIM Sparse-Mode Source Registration

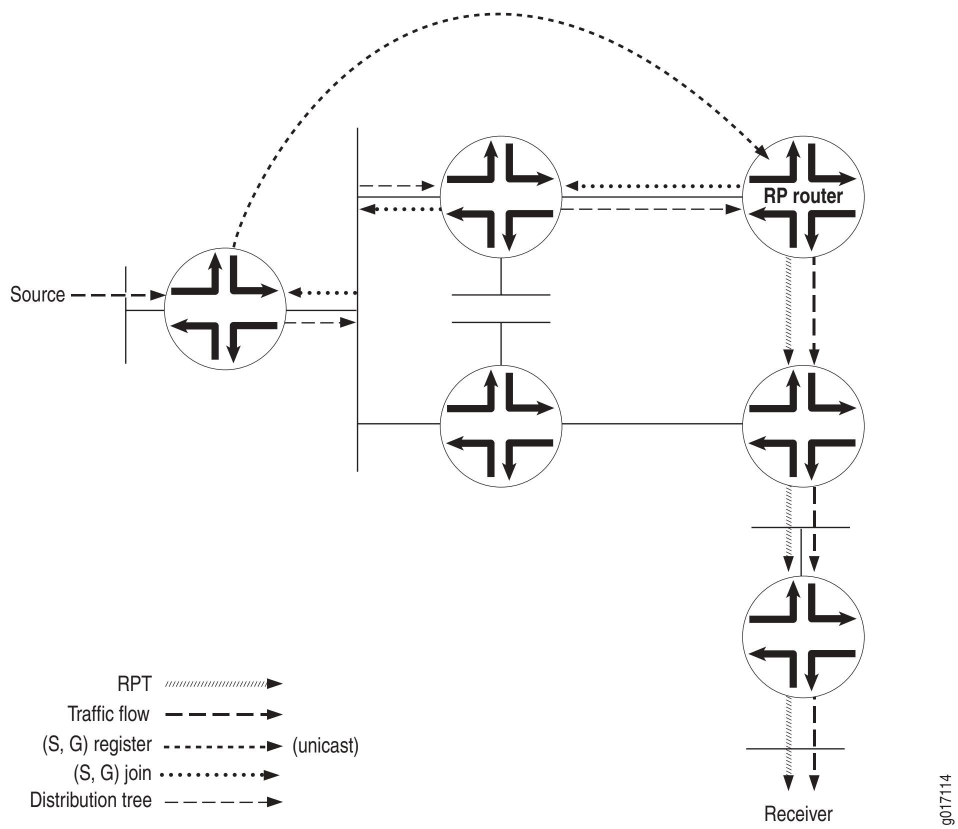
The RPT is a unidirectional tree, permitting traffic to flow down from the RP to the receiver in one direction. For multicast traffic to reach the receiver from the source, another branch of the distribution tree called the SPT needs to be built from the source's DR to the RP.

The SPT 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 out to the RP router (see Figure 16).

2. When the RP router receives the PIM register message from the source, it sends a PIM join message back to the source.

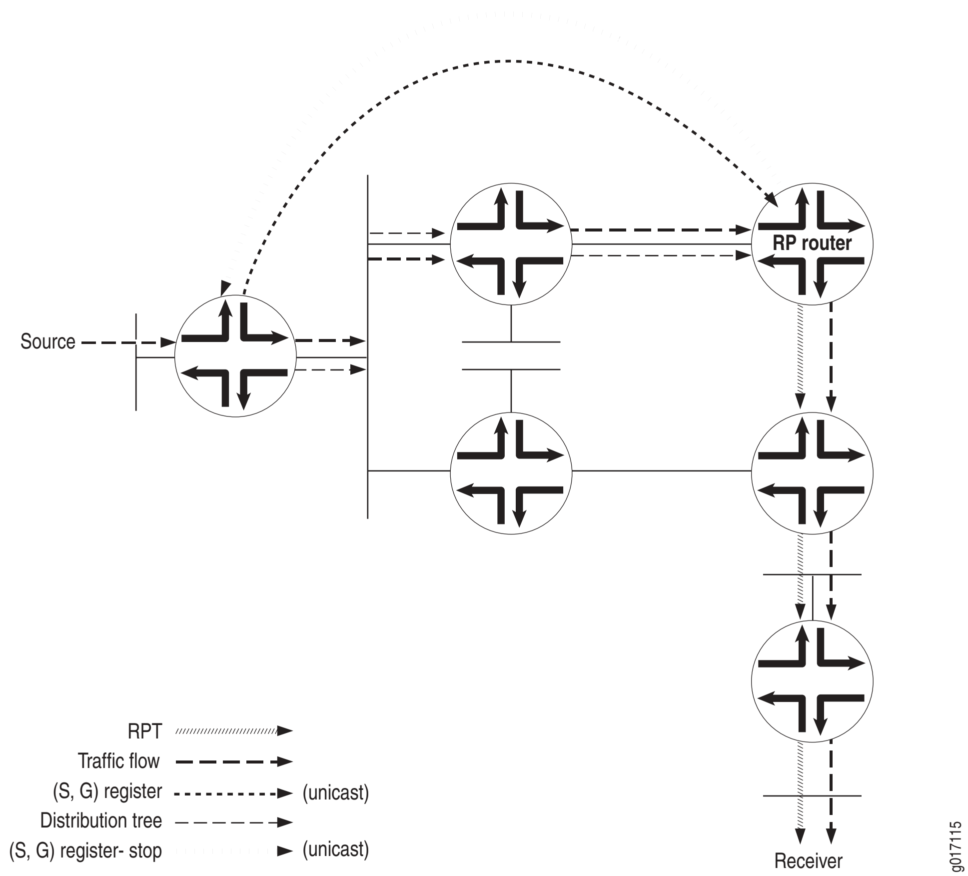
Figure 16: PIM Register Message and PIM Join Message Exchanged



3. The source's DR receives the PIM join message and begins sending traffic down the SPT toward the RP router (see Figure 17).

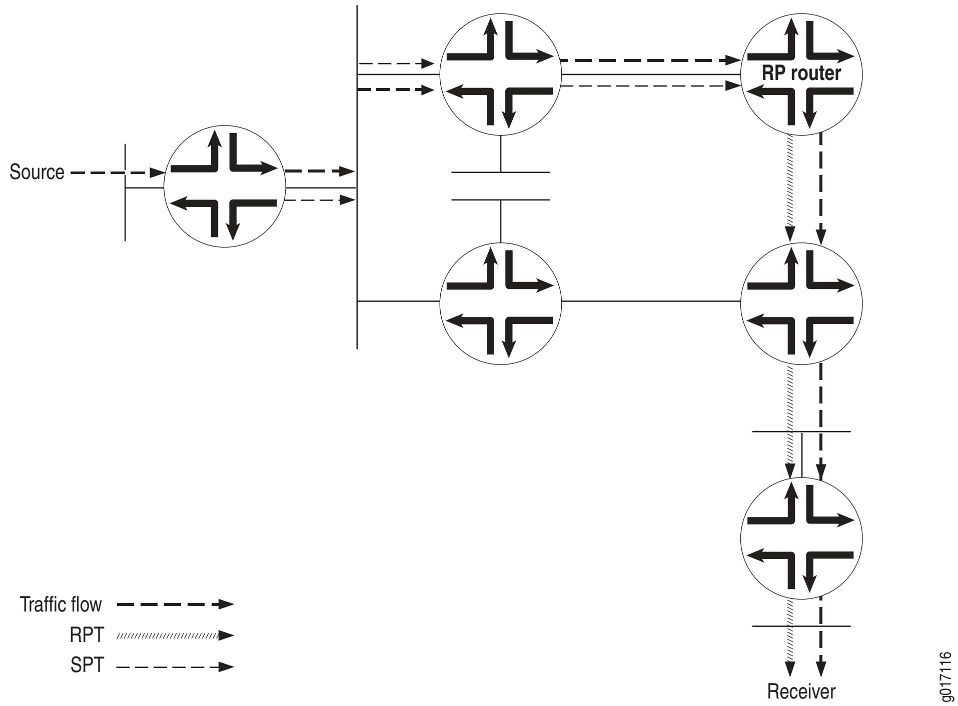
4. Once traffic is received by the RP router, it sends a register stop message to the source's DR to stop the register process.

Figure 17: 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 18).

Figure 18: Traffic Sent from the RP Router Toward the Receiver



PIM Sparse-Mode SPT Cutover

The RPT is not always the most direct path for delivering multicast traffic to a receiver. In many cases, a direct SPT from the last-hop router to the source is a better way to receive a multicast stream.

SPT Cutover

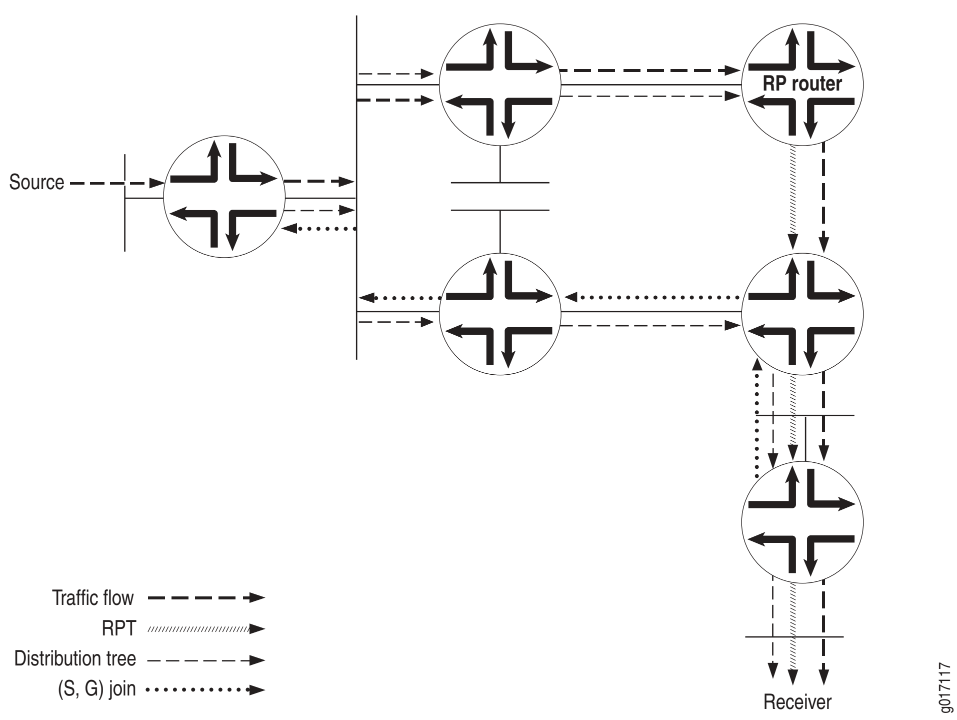
Instead of continuing to use the SPT to the RP and the RPT toward the receiver, a direct SPT is created between the source and the receiver in the following way:

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 19).

2. The source's DR receives the PIM join message, and an additional (S,G) state is created to form the SPT.

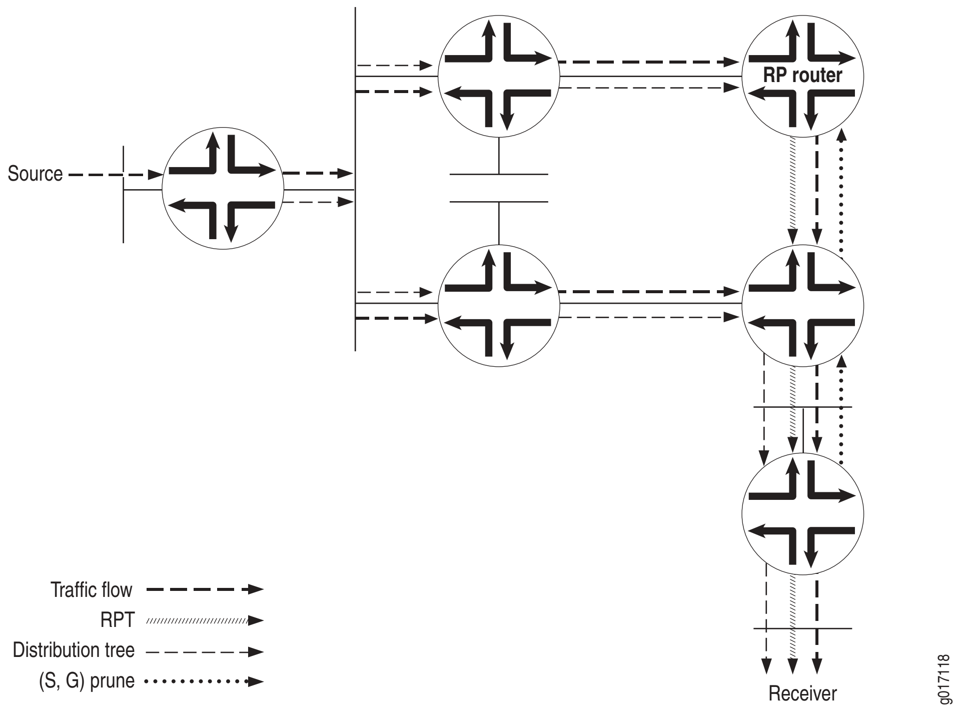
3. Multicast packets from that particular source begin coming from the source's DR and flowing down the new SPT to the receiver's DR. The receiver's DR is now receiving two copies of each multicast packet sent by the source—one from the RPT and one from the new SPT.

Figure 19: Receiver DR Sends a PIM Join Message to the Source



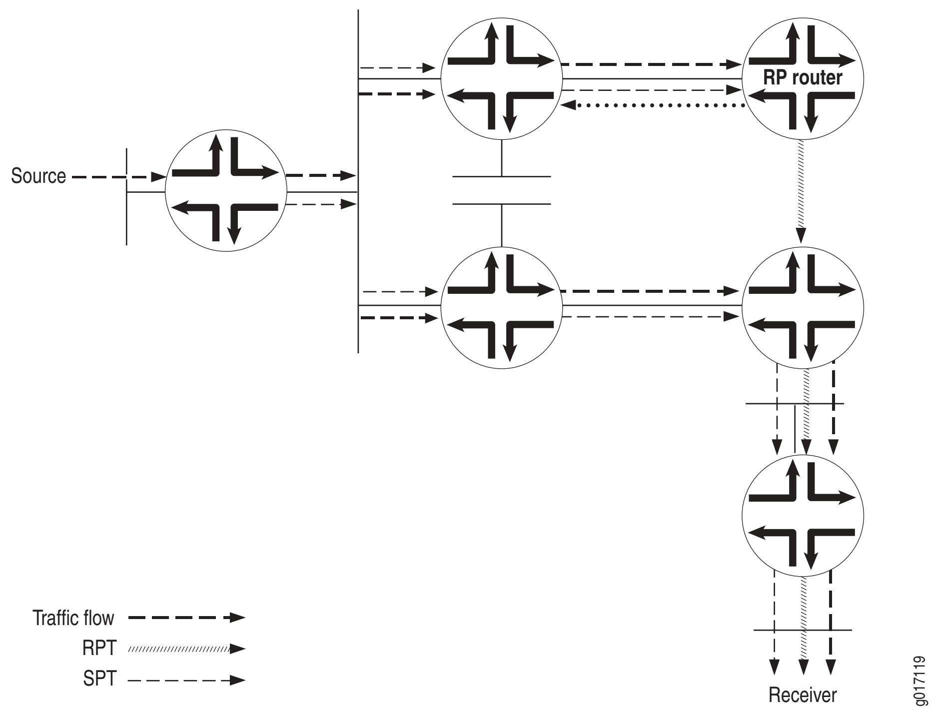
4. To stop duplicate multicast packets, the receiver's DR sends a PIM prune message toward the RP router, letting it know that the multicast packets from this particular source coming in from the RPT are no longer needed (see Figure 20).

Figure 20: PIM Prune Message is Sent from the Receiver's DR Toward the RP Router



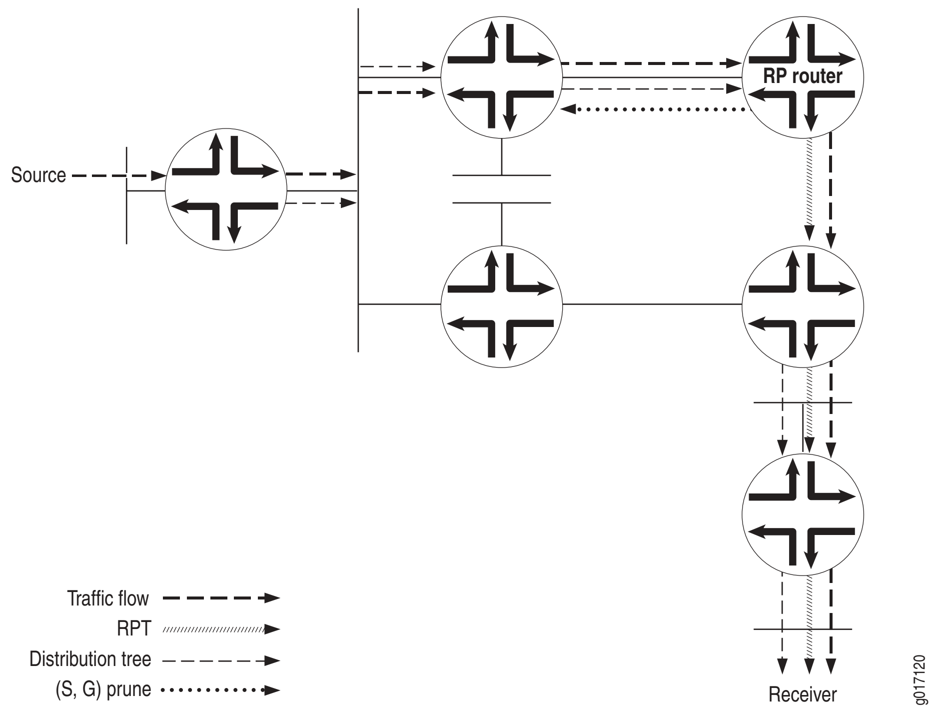
5. The PIM prune message is received by the RP router, and it stops sending multicast packets down to the receiver's DR. The receiver's DR is getting multicast packets only for this particular source over the new SPT. However, multicast packets from the source are still arriving from the source's DR toward the RP router (see Figure 21).

Figure 21: RP Router Receives PIM Prune Message



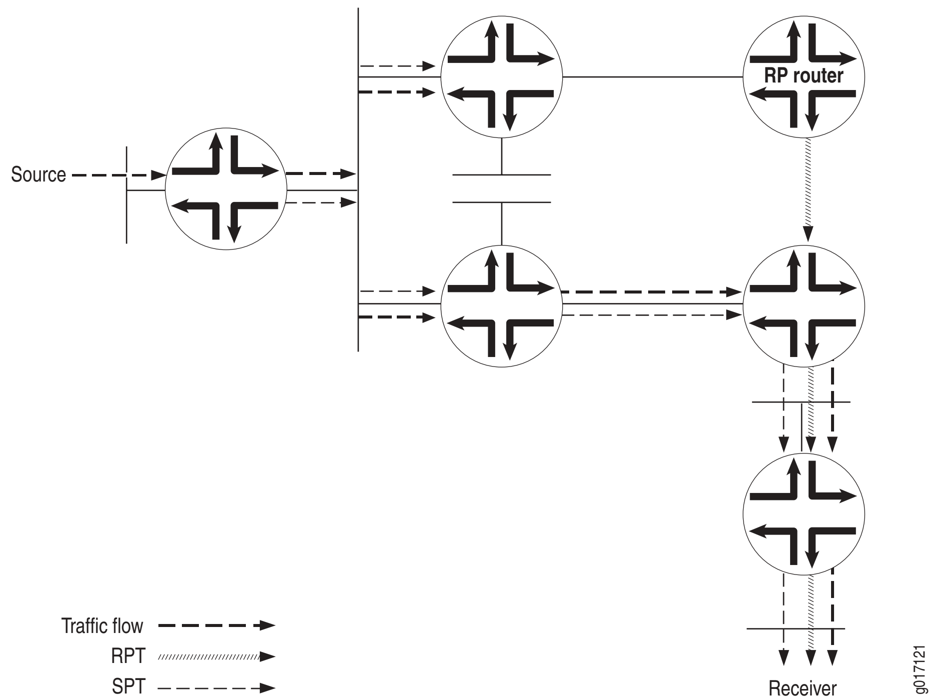
6. To stop the unneeded multicast packets from this particular source, the RP router sends a PIM prune message to the source's DR (see Figure 22).

Figure 22: 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 23).

Figure 23: Source's DR Stops Sending Duplicate Multicast Packets Toward the RP Router



SPT Cutover Control

In some cases, the last-hop router should 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.

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.

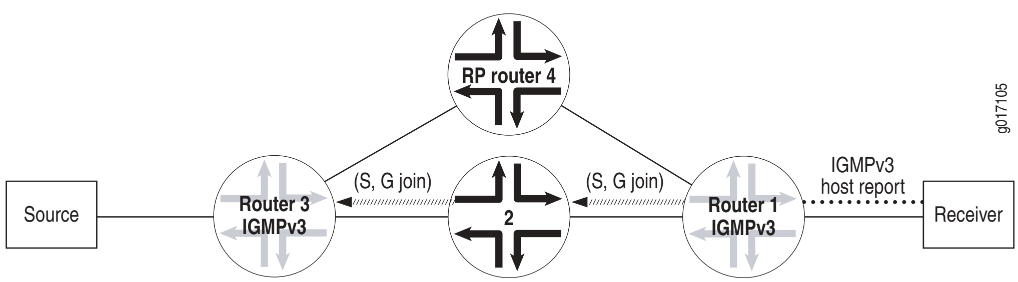
By default, the SSM group multicast address is limited to the IP address range 232.0.0.0 to 232.255.255.255. However, you can extend SSM operations into another Class D range by including the address statement at the [edit routing-options multicast ssm-groups] hierarchy level.

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 only 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.

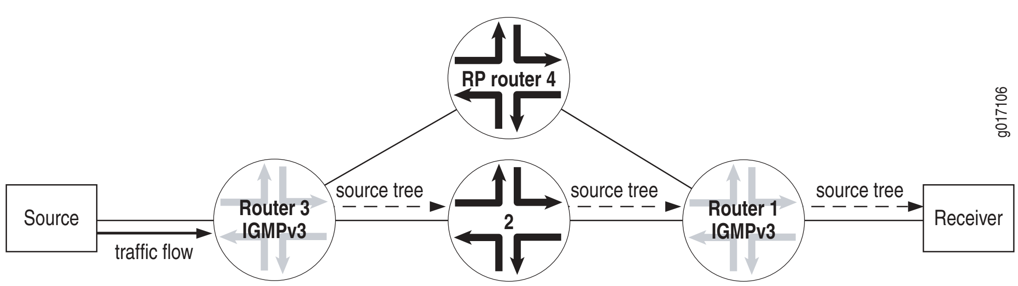
In a PIM SSM-configured network, a host subscribes to an SSM channel (by means of IGMPv3), announcing a desire to join group G and source S (see Figure 24). The directly connected PIM sparse-mode router, the receiver's DR, sends an (S,G) join message to its RPF neighbor for the source. Notice in Figure 24 that the RP is not contacted in this process by the receiver, as would be the case in normal PIM sparse-mode operations.

Figure 24: Receiver Announces Desire to Join Group G and Source S



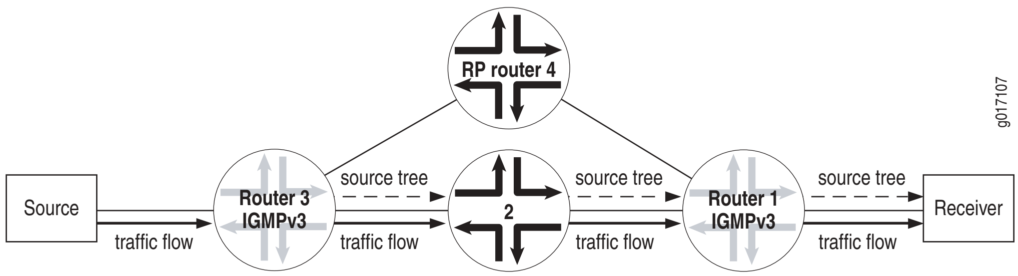
The (S,G) join message initiates the source tree, then builds it out hop by hop until it reaches the source. In Figure 25, the source tree is built across the network to Router 3, the last-hop router connected to the source.

Figure 25: Router 3 (Last-Hop Router) Joins the Source Tree



Using the source tree, multicast traffic is delivered to the subscribing host (see Figure 26).

Figure 26: The (S,G) State Is Built Between the Source and the Receiver



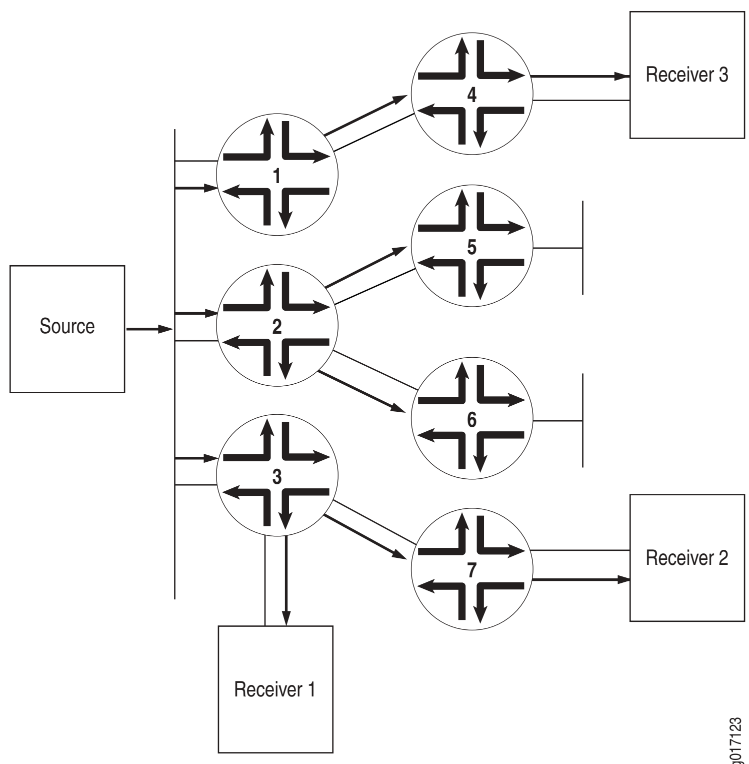
To configure additional SSM groups, include the ssm-groups statement at the [edit routing-options multicast] hierarchy level.

For more information about PIM SSM, see "Example: Configuring PIM SSM on a Network" on page 146.

PIM Dense Mode

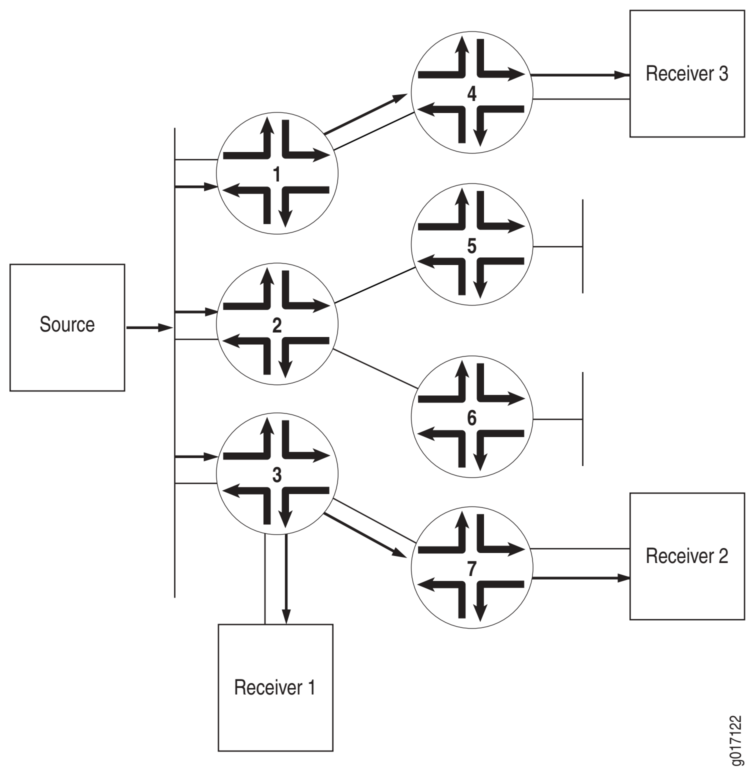
Unlike sparse mode, in which data is forwarded only to routers sending an explicit request, dense mode implements a *flood-and-prune* mechanism, similar to DVMRP. In PIM dense mode, there is no RP. A router receives the multicast data on the interface closest to the source, then forwards the traffic to all other interfaces (see Figure 27).

Figure 27: Multicast Traffic Flooded from the Source Using PIM Dense Mode



Flooding occurs periodically. It is used to refresh state information, such as the source IP address and multicast group pair. If the 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 28).

Figure 28: Prune Messages Sent Back to the Source to Stop Unwanted Multicast Traffic



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 "PIM Sparse Mode" on page 196 and "PIM Dense Mode" on page 213.

RP Mapping with Anycast RP

For the purposes of load balancing and redundancy, you can configure anycast RP. 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 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 goes down, sources and receivers are taken to a new RP by means of unicast routing.

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 Web site at http://www.ietf.org.

We recommend a static RP mapping with anycast RP and a bootstrap router (BSR) with auto-RP configuration because static mapping provides all the benefits of a bootstrap router and auto-RP without the complexity of the full BSR and auto-RP mechanisms.

See also “Example: Configuring Anycast RP” on page 258.

Multicast over Layer 3 VPNs

In the unicast environment for Layer 3 virtual private networks (VPNs), all VPN state information is contained within the provider edge (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 new 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.

For more information about configuring multicast for Layer 3 VPNs, see “Configuring Multicast for Layer 3 VPNs” on page 251. For multicast Layer 3 VPN examples, see “Example: Configuring PIM Sparse Mode over Layer 3 VPNs” on page 267.

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’s processor. The hardware used to create tunnel interfaces is a Tunnel Services Physical Interface Card (PIC). All RP routers and IP version 4 (IPv4) PIM sparse-mode DRs connected to a source require a Tunnel Services PIC.

In PIM sparse mode, the source DR takes the initial multicast packets and encapsulates them in PIM register messages. It then unicasts them 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 on any router (potentially a PIM sparse-mode source DR) and a Tunnel Services PIC is present on the router, a PIM register encapsulation interface, or pe interface, is automatically created for each RP address that is used to encapsulate source data packets and send them to respective RP addresses on the PIM DR as well as the PIM RP. The pe interface receives PIM register messages and encapsulates them by means of the hardware.

NOTE: 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 only appear 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 to be present.

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 should not be confused with IPv4 PIM sparse-mode requirements.

Table 7 shows the complete matrix of IPv4 and IPv6 PIM Tunnel PIC requirements.

Table 7: Tunnel PIC Requirements for IPv4 and IPv6 Multicast

IP Version:	Tunnel PIC on RP	Tunnel PIC on DR
IPv4	Yes	Yes
IPv6	Yes	No

Filtering Multicast Messages

Multicast sources and routers generate a considerable number of control messages, especially when using PIM sparse mode. These messages form distribution trees, locate RPs and 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 two types of multicast filtering to control the sending and receiving of multicast control messages.

This section discusses three types of multicast filtering. The last two require configuration:

- Filtering MAC Addresses on page 217
- Filtering RP/DR Register Messages on page 218
- Filtering MSDP SA Messages on page 219

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 Open Shortest Path First (OSPF), Routing Information Protocol version 2 (RIPv2), or Network Time Protocol (NTP) are configured on an interface, each of these protocols individually requests the interface to program the MAC filter to pick up their respective multicast group alone. 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/DR Register Messages

You can filter PIM register messages sent from the DR or to the 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 knows about, or which sources a DR tells other RPs about, is desired. A high degree of control over PIM register messages is provided by RP/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 in combination.

If the action of the register filter policy is to discard the register message, the router should send a register-stop message to the DR. These 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 are used to configure the filtering on these two fields. The router-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 will be 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 should have the same RP register message filtering policies configured; otherwise, it might be possible to circumvent the filtering policy. For more information on anycast RP, see “RP Mapping with Anycast RP” on page 215.

For more information on filtering RP/DR register messages, see “Configuring RP/DR Register Message Filtering” on page 246 and “Example: Configuring RP/DR Register Message Filters” on page 263.

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 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 should know the address of only one RP router, having more than one in the network can create problems. See “Example: Configuring PIM BSR Filters” on page 262 for a sample filter configuration.

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.

For more information, see “Multicast Scoping Overview” on page 133 and “Example: Configuring PIM Join Filters” on page 262.

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.

Embedded RP for IPv6 Multicast

Global IPv6 multicast between routing domains has been possible only with 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.

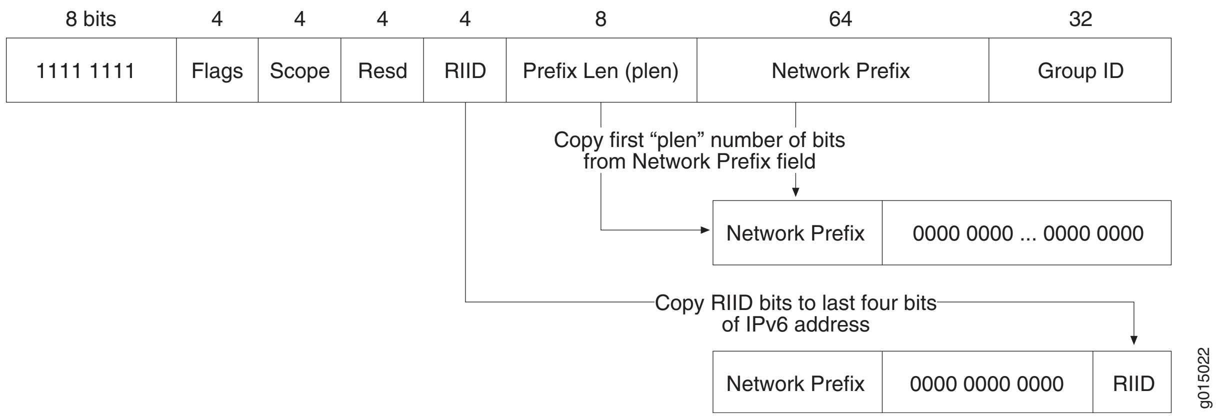
All IPv6 multicast addresses begin with 8 1-bits (`1111 1111`) followed by a 4-bit flag field normally set to `0011`. The flag field is set to `0111` when embedded RP is used. Then the low-order bits of the normally reserved field in the IPv6 multicast address carry the 4-bit RP interface identifier (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 29 for an illustration of this procedure.

Figure 29: 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 BSR 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.

Chapter 30

PIM Configuration Guidelines

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

```
pim {
    assert-timeout seconds;
    dense-groups {
        addresses;
    }
    disable;
    graceful-restart {
        disable;
        restart-duration seconds;
    }
    import [ policy-names ];
    interface interface-name {
        disable;
        hello-interval seconds;
        mode (dense | sparse | sparse-dense);
        neighbor-policy policy-name;
        priority number;
        version version;
    }
    rib-group group-name;
    rp {
        auto-rp {
            (announce | discovery | mapping);
            (mapping-agent-election | no-mapping-agent-election);
        }
        bootstrap {
            family (inet | inet6) {
                priority number;
                import [ policy-names ];
                export [ policy-names ];
            }
        }
        bootstrap-export [ policy-names ];
        bootstrap-import [ policy-names ];
        bootstrap-priority number;
```

```
            dr-register-policy [ policy-names ];
            embedded-rp {
                maximum-rps limit;
                group-ranges {
                    destination-mask;
                }
            rp-register-policy [ policy-names ];
            }
            local {
                family (inet | inet6) {
                    address address;
                    anycast-pim {
                        rp-set {
                            address address [forward-msdp-sa];
                        }
                        local-address address;
                    }
                    disable;
                    group-ranges {
                        destination-mask;
                    }
                    hold-time seconds;
                    priority number;
                }
            }
            static {
                address address {
                    version version;
                    group-ranges {
                        destination-mask;
        }
        spt-threshold {
            infinity [ spt-threshold-infinity-policies ];
        }
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit routing-instance *instance-name* protocols]
- [edit logical-routers *logical-router-name* protocols]
- [edit routing-instance *instance-name* logical-routers *logical-router-name* protocols]

By default, PIM is disabled.

This chapter includes the following PIM tasks:

Configuring PIM Mode-Independent Interface Properties

You can configure the following properties regardless of whether the PIM interface is configured in sparse, dense, or sparse-dense mode:

If you configure PIM on an aggregated interface (ae- or as-), each of the interfaces in the aggregate will be included in the multicast output interface list and will carry the single stream of replicated packets in a load-sharing fashion. The multicast aggregate interface will be “expanded” into its constituent interfaces in the next-hop database.

For information about aggregate interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

For information about configuring the PIM mode on an interface, see “Configuring PIM Dense Mode Properties” on page 231, “Configuring PIM Sparse Mode Properties” on page 231, and “Configuring Sparse-Dense Mode Properties” on page 250.

Changing the PIM Version

All systems on a subnet must run the same version of PIM.

By default, the JUNOS software uses PIM version 2. To configure PIM version 1, include the version statement:

```
version 1;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

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 rendezvous point (RP) mode (at the [pim rp static address *address*] hierarchy level). However, PIM version 2 is the default for interface mode (at the [pim interface *interface-name*] hierarchy level). Explicitly configured versions override the defaults.

Configuring the Designated Router Priority

By default, a PIM interface has the lowest probability of being selected as the designated router (DR). To change this, include the priority statement:

```
priority number;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The default priority is 1. Use a larger number to increase the probability of the interface's being elected as the DR.

Modifying the Hello Interval

Routers send hello messages at a fixed interval on all PIM-enabled interfaces. Using hello messages, routers advertise their existence as a PIM router on the subnet. With all PIM-enabled routers advertised, a single DR for the subnet is established.

When a router is configured for PIM, it sends out a hello message at a 30-second default interval. The interval range is from 0 through 255. When the interval counts down to 0, it sends out another hello message, and the timer is reset. A router that gets 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 router would wait for a response is 105 seconds.

To modify how often the router sends hello messages out of an interface, include the hello-interval statement:

```
hello-interval seconds;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring Interface-Level Neighbor Policies

To configure a policy to filter unwanted PIM neighbors include the neighbor-policy statement at the pim interface *interface-name* level:

```
neighbor-policy policy-name
```

For a list of hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

The neighbor policy must be a properly structured policy statement that uses a prefix list containing the neighbor primary address (or any secondary IP addresses) in a prefix list and the reject option to reject the unwanted address. For example:

```
prefix-list nbrGroup1 {
    20.20.20.1/32
}
policy-statement nbr-policy {
    from {
      prefix-list nbrGroup1
    }
    then reject;
}
```

For more information about configuring prefix lists or policy statements, see the *JUNOS Policy Framework Configuration Guide*.

The PIM interface compares neighbor IP addresses with the IP addresses 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 with 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.

Disabling the PIM Interface

To disable PIM on an interface, include the disable statement:

```
disable;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Configuring Other PIM Mode-Independent Properties

You can configure the following properties regardless of whether PIM is configured in sparse, dense, or sparse-dense mode:

- Configuring a PIM RPF Routing Table on page 228
- Filtering PIM Join Messages on page 229

Configuring a PIM RPF Routing Table

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 to resolve the RPF neighbor for the RP address. PIM can optionally use inet.2 as its RPF routing table group. To do this, add the rib-group statement to the [routing-options] hierarchy level, and then name the routing table group in the pim statement:

```
protocols {
    pim {
        rib-group group-name;
    }
}
```

For more information on configuring RIB groups, see the *JUNOS Routing Protocols Configuration Guide*.

This example uses the routing table group pim-rg to populate inet.2 for RPF checks:

```
routing-options {
    rib-groups {
        pim-rg {
            import-rib inet.2;
        }
    }
}
protocols {
    pim {
        rib-group pim-rg;
    }
}
```

For a list of the hierarchy levels at which you can include these statements, see the statement summary section for this statement.

Specifying additional import routing table groups or an export routing table group in the routing table group has no effect on PIM operation. PIM uses the first routing table group specified as an import routing table group.

PIM uses a single routing table group as its RPF routing table group. This ensures that the route with the longest matching prefix is chosen as the RPF route.

You can configure OSPF to populate inet.2 with OSPF routes that have regular IP next hops. This allows RPF to work properly even when MPLS is configured for traffic engineering, or when OSPF is configured to use “shortcuts” for local traffic.

You can also configure IS-IS to populate inet.2 with IS-IS 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 is configured to use “shortcuts” for local traffic.

For more information on RPF tables and the OSPF and IS-IS routing protocols, see the *JUNOS Routing Protocols Configuration Guide*.

Filtering PIM Join Messages

While multicast scopes prevent 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 8 for a list of match conditions.

Table 8: 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

To create a routing policy to reject a PIM join request for a source, include a policy name at the [edit policy-options policy-statement] or [edit logical-routers *logical-router-name* policy-options policy-statement] hierarchy level.

To apply one or more policies to routes being imported into the routing table from PIM, include the import statement:

```
import [ policy-names ];
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For a PIM join filter example, see “Example: Configuring PIM Join Filters” on page 262.

NOTE: Configuring multicast scoping on all routers filters the actual data and might be preferable to a PIM join filter solution. For more information about multicast scoping, see “Multicast Scoping Overview” on page 133.

Multicast Performance and 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 configure, include the no-multicast-echo statement at the [edit system] hierarchy level:

```
system {
    no-multicast-echo;
}
```

For more information about this statement, see the *JUNOS System Basics Configuration Guide*.

Configuring PIM Trace Options

To trace PIM protocol traffic, you can specify options in the global traceoptions statement at the [edit routing-options] or [edit logical-routers *logical-router-name* routing-options] hierarchy level. Options applied at the routing options level trace all packets, and options applied at the protocol level trace only IGMP traffic.

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can configure PIM-specific options by including the traceoptions statement at the PIM hierarchy level. For a list of the hierarchy levels at which you can configure this statement, see the statement summary section for this statement.

You can specify the following PIM-specific options in the traceoptions statement:

- 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.
- 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.
- cache—Trace the packets in the PIM routing cache.
- graft—Trace graft and graft acknowledgment messages.
- hello—Trace hello packets, which are sent so that neighboring routers can discover each other.
- join—Trace join messages, which are sent to join a branch onto the multicast distribution tree.
- packets—Trace all PIM 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.
- rp—Trace candidate RP advertisements.

For general information about tracing, see the *JUNOS System Basics Configuration Guide*. For a PIM tracing example, see “Example: Tracing PIM Protocol Traffic” on page 265.

Configuring PIM Dense Mode Properties

To configure the router properties for PIM dense mode, enable the minimum PIM dense mode configuration. For information about operating interfaces in PIM dense mode, see “PIM Modes” on page 195.

By default, PIM is disabled. When you enable PIM, it operates in sparse mode by default. To enable PIM dense mode on the router, include the pim statement:

```
pim {
    rib-group group-name;
    interface interface-name;
    mode dense;
}
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

To specify that PIM dense mode use inet.2 as its RPF routing table instead of inet.0, include the rib-group statement. For more information about configuring RPF routing tables, see “Configuring a PIM RPF Routing Table” on page 228.

You can specify the interfaces on which to enable PIM. Specify the full name, including the physical and logical address components. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*. 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.

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.

Configuring PIM Sparse Mode Properties

To configure PIM sparse mode properties, see the following sections:

- Configuring the Router's Local RP Properties on page 235
- Configuring Static RPs on page 238
- Configuring Bootstrap Properties on page 238
- Configuring Auto-RP on page 241
- Configuring RP/DR Register Message Filtering on page 246
- Configuring Embedded RP for IPv6 on page 247
- Configuring the Assert Timeout on page 248
- Configuring the SPT Threshold Policy on page 248

For information about operating interfaces in PIM sparse mode, see "PIM Modes" on page 195.

Minimum PIM Sparse Mode Configuration

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 RP is the root of this shared tree.

To configure this router's properties as the candidate RP, include the rp statement:

```
rp {
    local {
      family (inet | inet6) {
        disable;
        address address;
        group-ranges {
          destination-mask;
        }
        hold-time seconds;
        priority number;
      }
    }
    auto-rp {
      (announce | discovery | mapping);
      (mapping-agent-election | no-mapping-agent-election);
    }
    bootstrap {
      family (inet | inet6) {
        priority number;
        import [ policy-names ];
        export [ policy-names ];
      }
    }
    bootstrap-export [ policy-names ];
```

```
            bootstrap-import [ policy-names ];
            bootstrap-priority number;
            dr-register-policy [ policy-name ];
            embedded-rp {
                maximum-rps limit;
                group-ranges {
                    destination-mask;
                }
            }
            rp-register-policy [ policy-name ];
            static {
                address address {
                    version version;
                    group-ranges {
                        destination-mask;
                    }
                }
            }
        }
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Logical Routers and PIM Sparse Mode

Logical routers partition a single physical router into multiple logical devices that perform independent routing tasks. Because logical routers perform a subset of the tasks once handled by the physical router, logical routers offer an effective way to maximize the use of a single router platform.

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

Enabling PIM Sparse Mode

You can configure PIM interfaces to operate in sparse, dense, or sparse-dense mode. Sparse mode is the default. There is no need to explicitly configure sparse mode on a PIM interface, but this is often done for clarity or when you configure a change from dense to sparse mode.

To explicitly configure PIM to operate in sparse mode on an interface, include the mode sparse statement:

```
mode sparse;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

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.

NOTE: If you configure PIM sparse-dense mode, only sparse multicast groups benefit from 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 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 software 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 interval timer expires. On platforms that support PIM sparse mode graceful restart, the restart can be completed within 30 to 300 seconds. The default restart duration is 60 seconds.

NOTE: 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.

To configure graceful restart for PIM sparse mode, include the graceful-restart statement:

```
graceful-restart {
    disable;
    restart-duration seconds;
}
```

For a list of the hierarchy levels at which you can configure the graceful-restart statement, see the statement summary section for this statement.

To disable graceful restart for PIM, include the disable statement:

```
disable;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

By default, the router allows 60 seconds for the restart duration. The range is from 30 to 300 seconds. After this restart time, the Routing Engine resumes normal multicast operation. To configure the restart duration, include the restart-duration statement:

```
restart-duration seconds;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For more information about graceful restart for PIM, see “Multicast Redundancy” on page 22. For more information about graceful restart and other routing protocols, see the *JUNOS Routing Protocols Configuration Guide* and the *JUNOS Feature Guide*.

Configuring the Router’s Local RP Properties

Local RP configuration makes the router a statically defined RP. To configure the router’s RP properties, include the local statement:

```
local {
    family (inet | inet6) {
        disable;
        address address;
        group-ranges {
            destination-mask;
        }
        hold-time seconds;
        priority number;
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols pim rp]
- [edit routing-instances *routing-instance-name* protocols pim rp]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

For information about the RP configuration statements, see the following sections:

- Configuring the IP Protocol Family on page 236
- Configuring the Local RP Address on page 236
- Configuring the Router’s RP Priority on page 237
- Configuring the Groups for Which the Router Is the RP on page 237
- Modifying the Local RP Hold Time on page 237

Configuring the IP Protocol Family

PIM supports both IP version 4 (IPv4) and IP version 6 (IPv6) addressing.

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.

For correct operation of PIM sparse mode, the RP address should be known to a router. The JUNOS IPv6 PIM implementation supports only static RP configuration. Automatic RP announcement and bootstrap routers are not available with IPv6. You configure the static IPv6 RP address in the same way as IPv4 addresses, by including the address statement. However, on a router that is itself the RP, include the address statement at the [edit protocols pim rp local family inet6] or [edit routing-instances *routing-instance-name* protocols pim rp local family inet6] hierarchy level.

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

The Multicast Listener Discovery (MLD) protocol is automatically enabled on any broadcast interfaces on which you configure PIM and family inet6. For an overview of MLD, see "MLD Overview" on page 71.

To specify whether IPv4 or IPv6 local RP properties apply to the configuration values, include the family statement:

```
family (inet | inet6);
```

You can include this statement at the following hierarchy levels:

- [edit protocols pim rp local]
- [edit routing-instances *routing-instance-name* protocols pim rp local]

Configuring the Local RP Address

To specify the local RP address, include the address statement:

```
address address;
```

You can include this statement at the following hierarchy levels:

- [edit protocols pim rp local family]
- [edit routing-instances *routing-instance-name* protocols pim rp local family]

Configuring the Router's RP Priority

The router's priority value for becoming the RP is included in the bootstrap messages that the router sends. Use a smaller number to increase the likelihood that the router becomes the RP for local multicast groups. Each PIM router 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 router determines algorithmically the RP from the candidate RP set using a well-known 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. To modify the router's priority, include the priority statement:

```
priority number;
```

You can include this statement at the following hierarchy levels:

- [edit protocols pim rp local family]
- [edit routing-instances *routing-instance-name* protocols pim rp local family]

The priority can be a number from 0 through 255.

Configuring the Groups for Which the Router Is the RP

By default, a router running PIM is eligible to be the RP for all groups (224.0.0.0/4). To limit the groups for which this router can be the RP, include the group-ranges statement:

```
group-ranges number {
    destination-mask;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols pim rp local family]
- [edit routing-instances *routing-instance-name* protocols pim rp local family]

Modifying the Local RP Hold Time

If the local router 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 router 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 router from its list of candidate RPs. The default hold time is 150 seconds.

To modify the hold-time value for the local RP, include the hold-time statement:

```
hold-time seconds;
```

You can include this statement at the following hierarchy levels:

- [edit protocols pim rp local family]

- [edit routing-instances *routing-instance-name* protocols pim rp local family]

Configuring Static RPs

Static RP configuration directs the router to another statically defined RP. To configure static RPs, include the static statement:

```
static {
    address address {
        version version;
        group-ranges {
            destination-mask;
        }
    }
}
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

NOTE: You can configure a static RP in a logical router only if the logical router is not directly connected to a source.

To configure other static RPs, include one or more address statements. The default multicast address group range is 224.0.0.0/4.

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 RP that you select for a particular group must be consistent across all routers in a multicast domain.

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 (at the [pim rp static address *address*] hierarchy level). However, PIM version 2 is the default for interface mode (at the [pim interface *interface-name*] hierarchy level). Explicitly configured versions override the defaults.

Configuring Bootstrap Properties

Bootstrap routers are supported in IPv4 and IPv6. For legacy configuration purposes, configuration details are retained for IPv4. However, a different configuration hierarchy can be used for both IPv4 and IPv6. This section describes both configuration methods, but only the IPv4/IPv6 configuration method is recommended.

To configure the router's bootstrap properties *for IPv4 only*, see the following sections:

- Configuring the Router's IPv4 Bootstrap Router Priority on page 239
- Filtering PIM IPv4 Bootstrap Messages on page 239

To configure IPv4 or IPv6 bootstrap properties, see the following sections:

Configuring the Router's IPv4 Bootstrap Router Priority

To determine which router is the RP, all routers within a PIM domain collect bootstrap messages. A PIM domain is a contiguous set of routers that implement PIM; all are configured to operate within a common boundary. The domain's bootstrap router originates 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.

By default, the router has a bootstrap priority of 0, which means the router can never be the bootstrap router. To modify this priority, include the bootstrap-priority statement. The router with the highest priority value is elected to be the bootstrap router. In the case of a tie, the router with the highest IP address is elected to be the bootstrap router. A simple bootstrap configuration simply assigns a bootstrap priority value to a router.

```
bootstrap-priority number;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

Filtering PIM IPv4 Bootstrap Messages

You can create import and export policies to control the flow of IPv4 bootstrap messages to and from the RP, and apply them to PIM. To apply one or more import policies to IPv4 bootstrap messages imported into the RP, include the bootstrap-import statement:

```
bootstrap-import [ policy-names ];
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

To apply one or more export policies to IPv4 bootstrap messages exported from the RP, include the bootstrap-export statement:

```
bootstrap-export [ policy-names ];
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

NOTE: The bootstrap-priority, bootstrap-import, and bootstrap-export statements support IPv4 only. A bootstrap-priority of 0 disables the function for IPv4 and does *not* cause the router to send BSR packets with a 0 in the priority field.

Configuring the Router's Bootstrap Router Priority

To determine which router is the RP, all routers within a PIM domain collect bootstrap messages. A PIM domain is a contiguous set of routers that implement PIM; all are configured to operate within a common boundary. The domain's bootstrap router originates 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.

The bootstrap configuration hierarchy supports both IPv4 and IPv6 multicasting. It can be combined with the bootstrap statements supported in IPv4 only, as long as the added statements are used for IPv6 only, but this is not recommended. There is a change in the meaning of the bootstrap priority when the value is set to 0.

In the IPv4 configuration hierarchy, a bootstrap-priority of 0 disables the function for IPv4 and does *not* cause the router to send BSR packets with a 0 in the priority field. In the IPv4/IPv6 configuration hierarchy, a priority of 0 does *not* disable the function, but causes the router to send BSR packets with a 0 in the priority field. To disable the bootstrap function in the IPv4/IPv6 hierarchy, delete the configuration statements.

A commit error will occur if the same IPv4 bootstrap statements are configured under the IPv4-only and IPv4/IPv6 sections of the hierarchy. The error message is "duplicate IPv4 bootstrap configuration."

We recommend that legacy IPv4-only configurations be transitioned to the IPv4/IPv6 configuration hierarchy.

To modify the bootstrap priority for IPv4 or IPv6, include the priority statement for the appropriate address family: inet for IPv4 and inet6 for IPv6. The router with the highest priority value is elected to be the bootstrap router. In the case of a tie, the router with the highest IP address is elected to be the bootstrap router.

```
priority number;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

This example snippet sets a bootstrap priority of 1 for both IPv4 and IPv6 multicasts:

```
pim {
    rp {
      bootstrap {
        family inet {
          priority 1;
        }
        family inet6 {
          priority 1;
        }
```

An error results when this configuration is combined with the use of the bootstrap-priority statement.

Filtering PIM Bootstrap Messages

You can create import and export policies to control the flow of bootstrap messages to and from the RP, and apply them to PIM. To apply one or more import policies to bootstrap messages imported into the RP, include the import statement:

```
import [ policy-names ];
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

To apply one or more export policies to bootstrap messages exported from the RP, include the export statement:

```
export [ policy-names ];
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

For an example, see "Example: Rejecting PIM Bootstrap Messages at the Boundary of a PIM Domain" on page 266.

NOTE: The bootstrap statement stanza supports both IPv4 and IPv6. A priority of 0 does *not* disable the function, but causes the router to send BSR packets with a 0 in the priority field. To disable the bootstrap function, delete the configuration statements.

Configuring Auto-RP

You can configure a mode-dynamic way of assigning 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. Auto-RP operates in PIM version 1 and version 2.

NOTE: If the router receives auto-RP announcements split across multiple messages, the router will lose the information in the previous part of the message as soon as the next part of the message message is received.

To configure auto-RP properties, see the following sections:

Configuring Auto-RP Announcement, Mapping, and Discovery

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: you can configure multiple routers as RP candidates. If the elected RP stops operating, one of the other preconfigured routers takes over the RP functions. This capability is controlled by the auto-RP mapping agent.

If PIM is operating in sparse or sparse-dense mode, configure how the router operates in auto-RP by specifying the following auto-RP options:

- Use the discovery option to let the router receive and process discovery messages from the mapping agent. This is the most basic auto-RP option.
- Add the announce option on the router to allow the router to send announce messages in the network, advertising itself as a candidate RP. Routers configured with this option must also be configured as RPs, or announce messages are not sent.
- Add the mapping option on the router to allow the router to perform the mapping function. If the router is also an RP, the mapping option also allows the router to send auto-RP announcements (mapping on an RP allows the router to perform both the announcement and mapping functions).

The router 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.

In most cases, how the router handles auto-RP discovery, announce, or mapping messages depends on whether the router is an RP (configured as local RP) or not. Table 9 shows how the router behaves depending on the local RP configuration.

Table 9: Local RP and Auto-RP Message Types

Auto-RP Message Type	Local RP?	Router Behavior
discovery	No	Listen for auto-RP mapping messages.
discovery	Yes	Listen for auto-RP mapping messages.
announce	No	Listen for auto-RP mapping messages.
announce	Yes	Listen for auto-RP mapping messages. Send auto-RP announce messages.
mapping	No	Listen for auto-RP mapping messages. Listen for auto-RP announce messages. If elected mapping agent, send auto-RP mapping messages.
mapping	Yes	Listen for auto-RP mapping messages. Send auto-RP announce messages. Listen for auto-RP announce messages. If elected mapping agent, send auto-RP mapping messages.

To configure auto-RP at the main hierarchy level, follow these steps:

1. Include the mode statement, and specify the option sparse-dense on all interfaces at the [edit protocols pim] hierarchy level:

```
[edit protocols pim]
interface all {
    mode sparse-dense;
}
```

This configuration allows the router to operate in sparse mode for most groups and dense mode for others. The default is to operate in sparse mode unless the router is specifically informed of a dense mode group.

2. Configure two multicast dense mode groups (224.0.1.39 and 224.0.1.40) using the dense-groups statement at the [edit protocols pim] hierarchy level:

```
[edit protocols pim]
dense-groups {
    224.0.1.39/32;
    224.0.1.40/32;
}
```

   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.

3. Include the auto-RP statement at the [edit protocols pim rp] hierarchy level to configure auto-RP on each router in the group. There are four possible categories for each router.

   - If the router is not a local RP and listens only for auto-RP mapping messages, include the auto-rp discovery statement to the router RP configuration at the [edit protocols pim rp] hierarchy level:

```
[edit protocols pim rp]
auto-rp discovery;
```

   - If the router is a local RP, sends auto-RP announcements, and listens for auto-RP mapping messages, configure the router as a local RP and include the auto-rp announce statement to the router RP configuration at the [edit protocols pim rp] hierarchy level:

```
[edit protocols pim rp]
local {
    address 10.0.1.1;
}
auto-rp announce;
```

   - If the router performs only the mapping function to listen for auto-RP announcements, performs the auto-RP-to-group mapping, and sends auto-RP mapping messages, include the auto-rp mapping statement at the [edit protocols pim rp] hierarchy level. When multiple candidate RP routers announce their capabilities to support multicast groups, there must be a single router in the network to act as mapping agent. The mapping agent sends out discovery messages to the network, informing all routers in a multicast group of the RP to use:

```
[edit protocols pim rp]
auto-rp mapping;
```

- If the router combines the local RP function to send announcements and also perform the mapping function, configure the router as a local RP and include the auto-rp mapping statement to the router RP configuration at the [edit protocols pim rp] hierarchy level:

```
[edit protocols pim rp]
local {
    address 10.0.1.1;
}
auto-rp mapping;
```

All routers must also have a routable IP address on the lo0 interface:

```
interface lo0 {
    unit 0 {
      family inet {
        address 127.0.0.1;   /* this address cannot be used by auto-rp */
        address 192.168.27.1 { /* this example uses a private IP address */
          preferred;
        }
      }
    }
}
```

You can include these statements at the following hierarchy levels (auto-RP announce is not supported in logical routers):

- [edit protocols]
- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-routers *logical-router-name* protocols] (all statements except auto-rp announce)
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols] (all statements except auto-rp announce)

Use the show pim rps command to verify the auto-RP information:

```
user@host> show pim rps
RP address     Type    Holdtime    Timeout  Active groups   Group prefixes
192.168.5.1    auto-rp     150       123                1   224.0.0.0/4
```

Use 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.

Configuring Auto-RP Mapping Agent Election

Auto-RP specifications state that mapping agents should not send mapping messages if they receive messages from a mapping agent with a higher IP address. This process is called *mapping agent election*. 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.

For compatibility, you can suppress mapping messages with 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.

The default auto-RP operation is to perform mapping agent election. To explicitly enable mapping agent election, configure the mapping-agent-election statement at the [edit protocols pim rp auto-rp] hierarchy level of an auto-RP mapping agent:

```
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 explicitly disable mapping agent election for compatibility with other vendor's equipment, configure the no-mapping-agent-election statement at the [edit protocols pim rp auto-rp] hierarchy level of an auto-RP mapping agent:

```
auto-rp {
    mapping;
    no-mapping-agent-election;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-routers *logical-router-name* protocols]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols]

Configuring RP/DR Register Message Filtering

You configure RP/DR register message filtering to control the number and location of multicast sources that an RP knows. 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 should have the same register message filtering policy configured.

To filter incoming register messages at the RP, configure the rp-register-policy statement at the [edit protocols pim rp] hierarchy level of an RP:

```
pim {
    rp {
        rp-register-policy [ example-rp-register-policy ];
        local {
            address 10.10.10.5;
        }
    }
}
```

To filter outgoing register messages at the DR, configure the dr-register-policy statement at the [edit protocols pim rp] hierarchy level of a DR:

```
pim {
    rp {
        dr-register-policy [ example-dr-register-policy ];
        static {
            address 10.10.10.5;
        }
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit routing-instances *routing-instance-name* protocols]
- [edit logical-routers *logical-router-name* protocols]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols]

If you delete a group and source address from a filter policy on an RP router, the RP will register the group and source only when the DR sends a null register message.

You can configure more than one policy for each statement. If a referenced policy does not exist, the configuration commit checkout will fail. For examples of both types of register filters, see “Example: Configuring RP/DR Register Message Filters” on page 263.

For more information on RP/DR register message filtering, see “Filtering RP/DR Register Messages” on page 218.

Configuring 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.

Embedded RP is disabled by default. To configure embedded RP for IPv6 PIM sparse mode, include the embedded-rp statement:

```
embedded-rp {
    maximum-rps limit;
    group-ranges {
        destination-mask;
    }
}
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

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.

The group-range statement determines 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.

If the derived RP address is not a valid IPv6 unicast address, it is treated as any other multicast group address and not used for RP information. Verification fails if the extracted RP address is a local interface, unless the routing platform is configured as an RP and the extracted RP address matches the configured RP address. Then the local RP decides whether it is configured to act as an RP for the embedded RP multicast address.

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.

For more information about the use of embedded RP, see “Embedded RP for IPv6 Multicast” on page 219.

Configuring the Assert Timeout

You configure the assert timeout to determine how often multicast routers running PIM sparse mode enter a PIM assert message cycle. Multicast routers 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 routers determine which router forwards the traffic and prunes the RPT for this group.

Assert messages are useful for LANs that connect multiple routers and no hosts. For more information about network configurations using assert timeouts, see “PIM Sparse-Mode SPT Cutover” on page 206.

To configure the assert timeout for PIM sparse mode, include the assert-timeout statement:

```
assert-timeout seconds;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

The range is from 5 through 210 seconds. The default is 210 seconds.

Configuring the SPT Threshold Policy

Multicast routers running PIM sparse mode can forward the same stream of multicast packets onto the same LAN through a rendezvous-point tree (RPT) rooted at the RP or a shortest-path tree (SPT) rooted at the source. In some cases, the last-hop router should stay on the shared RPT to the RP and *not* transition to a direct SPT to the source. For more information about these SPT cutover cases, see “SPT Cutover Control” on page 210.

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.

To configure the SPT threshold and policy for PIM sparse mode, include the spt-threshold statement:

```
spt-threshold {
    infinity [ spt-threshold-infinity-policies ];
}
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

NOTE: If you want the PE router to stay on the RPT for control traffic, include the spt-threshold statement under the main PIM instance.

The infinity 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.

Several points are important when configuring the SPT threshold policy:

- SPT Threshold Policy Configuration Changes on page 249
- Examples of SPT Threshold Policy Configuration on page 249

SPT Threshold Policy Configuration Changes

Configuration changes to the SPT threshold policy affect how the router handles the SPT transition:

- When the policy is configured for the first time, the router 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.

NOTE: 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 router 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 router does *not* use the new policy until the PIM-join state is cleared with the clear pim join command.

Examples of SPT Threshold Policy Configuration

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.

```
protocols {
    pim {
...
        spt-threshold {
            infinity spt-threshold-infinity-policy;
        }
...
    }
...
policy-options {
    policy-statement spt-threshold-infinity-policy {
        term one {
            from {
                router-filter 224.1.1.1/32 exact;
                source-address-filter 10.10.10.1/32 exact;
            then accept;
```

```
            term two {
                then reject;
            }
        }
    }
```

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 will never transition 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 will transition to a direct SPT to the source.

Configuring Sparse-Dense Mode Properties

To configure PIM to operate in sparse-dense mode on an interface, include the mode sparse-dense statement. Include the dense-groups statement to specify which groups are operating in dense mode:

```
dense-groups {
    addresses;
}
interface interface-name {
    mode sparse-dense;
}
```

For a list of the hierarchy levels at which you can include these statements, see the statement summary section for this statement.

You can configure graceful restart with PIM sparse-dense mode, but only sparse multicast groups benefit from graceful restart. For more information about graceful restart for PIM sparse mode, see “Configuring PIM Sparse Mode Graceful Restart” on page 234.

For an example of a sparse-dense mode configuration, see “Example: Configuring Sparse-Dense Mode” on page 258.

Configuring the BFD Protocol

The bidirectional forwarding detection (BFD) protocol uses control packets and shorter detection time limits to detect failures more rapidly in a network. Working with a wide variety of network environments and topologies, BFD failure detection timers provide faster detection by using shorter time limits than the PIM hello hold time. These timers are also adaptive and you can adjust them to be more or less aggressive.

To enable failure detection, include the bfd-liveness-detection statement:

```
bfd-liveness-detection {
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    minimum-transmit-interval milliseconds;
```

```
        multiplier number;
        version (0 | 1 | automatic);
    }
```

NOTE: You must specify the minimum transmit and minimum receive intervals to enable BFD on PIM.

To specify the minimum transmit and receive interval for failure detection, include the minimum-interval statement:

```
minimum-interval milliseconds;
```

NOTE: Specifying an interval smaller than 300 ms can cause undesired BFD flapping.

To specify only the minimum receive interval for failure detection, include the minimum-receive-interval statement:

```
minimum-receive-interval milliseconds;
```

To specify only the minimum transmit interval for failure detection, include the minimum-transmit-interval statement:

```
minimum-transmit-interval milliseconds;
```

To specify the detection time multiplier for failure detection, include the multiplier statement:

```
multiplier number;
```

To specify the BFD version used for detection, include the version statement:

```
version (0 | 1 | automatic);
```

For a list of hierarchy levels at which you can configure these statements, see the statement summary sections for these statements.

Configuring Multicast for Layer 3 VPNs

If the service provider supports PIM, you can configure multicast for a Layer 3 virtual private network (VPN) using PIM version 2 as the multicast protocol. The JUNOS software complies with RFC 2547 and Internet draft draft-rosen-vpn-mcast-07.txt, *BGP/MPLS VPNs* and *Multicast in MPLS/BGP VPNs,* Section 2 (Multicast Domains).

For multicast to work on Layer 3 VPNs, each of the following routers must have a Tunnel Services PIC, which is hardware used to encapsulate and de-encapsulate data packets into tunnels:

- 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 PIC.

When you complete the configuration, two multicast tunnel interfaces are configured automatically. You do not need to configure the tunnel interfaces. The interface mt-[*xxxxx*], used for encapsulation, 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. 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.

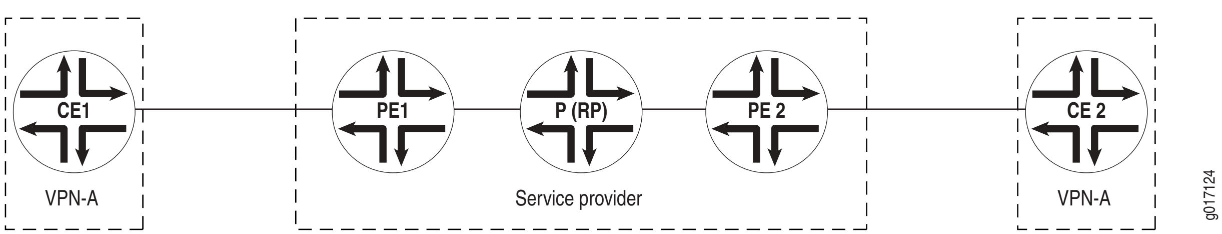
NOTE: It is possible for the PE router to be configured as the VPN customer RP (C-RP) router. The PE router can also act as the DR. This type of PE configuration can simplify configuration of customer DRs and VPN C-RPs for multicast VPNs. However, the BSR and auto-RP features are not supported. This section does not discuss the use of the PE as the VPN C-RP.

This section describes how to configure multicast for Layer 3 VPNs:

Configuring the VPN

You must first configure the VPN. Figure 30 shows a configuration for VPN-A, used as an example later in this section. For more information about configuring VPNs, see the *JUNOS VPNs Configuration Guide*.

Figure 30: Configuring the VPN



Configuring PIM Connectivity Between the Provider and PE Routers

To configure PIM on the main routing instance for all provider and PE routers, include statements at the [edit protocols pim] hierarchy level:

1. Configure the interfaces between each provider router and the PE routers by including the interface statement at the [edit protocols pim] hierarchy level. On all PE routers, enable PIM version 2 and sparse mode on interface lo0 of the PE routers, either by configuring that specific interface or by including the statement set version 2 mode sparse for interface all at the [edit protocols pim] hierarchy level on a PE router.

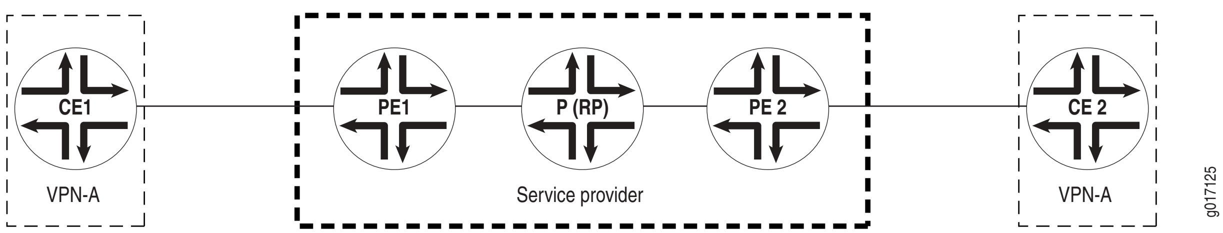
2. Configure PIM version 2 by including the version statement at the [edit protocols pim interface *interface-name*] hierarchy level.

3. Configure sparse mode (the mode in which the PIM interfaces operate) by including the mode statement at the [edit protocols pim interface *interface-name*] hierarchy level.

4. Configure the RP address by including the static statement at the [edit protocols pim rp] hierarchy level. In Figure 31, the provider router is the RP.

Figure 31 shows a multicast configuration on the provider network.

Figure 31: Multicast Configuration on the Provider Network



Configuring Multicast Connectivity on the CE Routers

To configure PIM for the master routing instance on all CE routers, include statements at the [edit protocols pim] hierarchy level:

1. Configure the interfaces going toward the provider router acting as the RP by including the interface statement at the [edit protocols pim] hierarchy level. In Figure 32, the interfaces are labeled A and B.

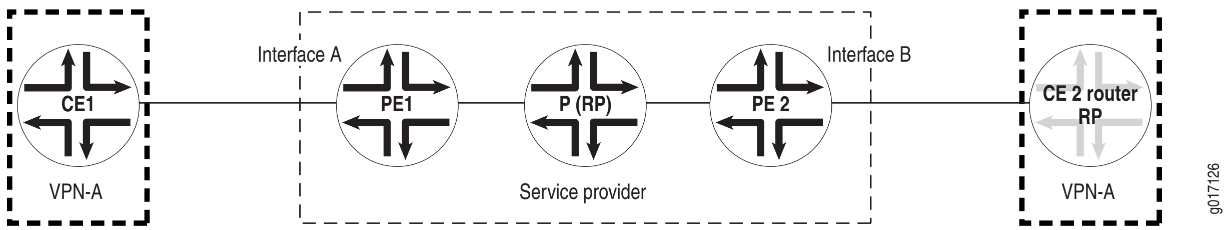
2. Configure PIM version 2 by including the version statement at the [edit protocols pim interface *interface-name*] hierarchy level.

3. Configure sparse mode or sparse-dense mode (the mode in which the PIM interfaces operate) by including the mode statement at the [edit protocols pim interface *interface-name*] hierarchy level.

4. Configure the RP address by including the static statement at the [edit protocols pim rp] hierarchy level. In Figure 32, CE2 is the RP router; however, the RP router can be anywhere in the customer network.

Figure 32 shows multicast connectivity on the customer edge.

Figure 32: Multicast Connectivity on the CE Routers



Configuring Multicast Connectivity for the VPN on the PE Router

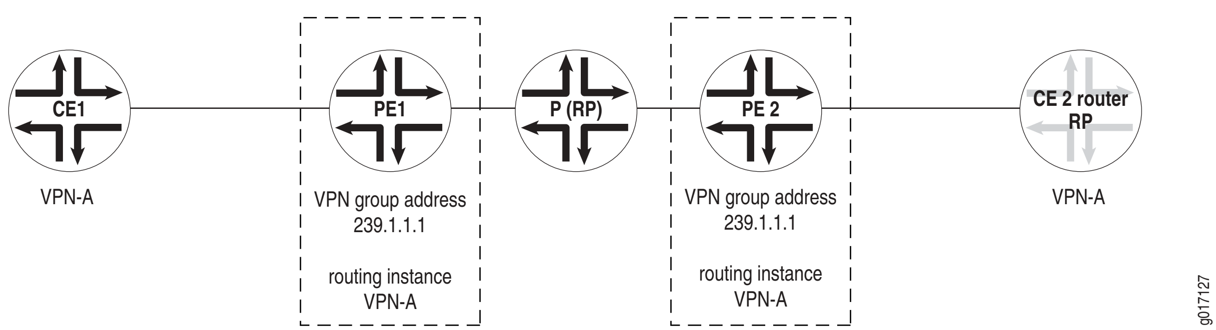
To configure multicast connectivity for the VPN on the PE router, you must configure a VPN group address and configure the interfaces toward the router acting as RP. To configure the VPN group address, include the vpn-group-address statement at the [edit routing-instances *instance-name* protocols pim] hierarchy level:

```
[edit routing-instances instance-name protocols pim]
vpn-group-address address;
```

The PIM configuration in the VPN routing and forwarding (VRF) instance on the PE routers should 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). See the *JUNOS VPNs Configuration Guide* for information about configuring VPNs on PE routers.

NOTE: 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 33 on page 254, the shared VPN group address configuration is 239.1.1.1.

Figure 33: Multicast Connectivity for the VPN



Configuring the Routing Group

Routing groups are usually configured at the [edit routing-instances routing-options] hierarchy level. However, with multicast in VRF instances, you must configure routing groups differently. Configure the multicast routing group by adding the rib-groups statement at the [edit routing-options] hierarchy level.

After you configure the multicast routing group in the main routing instance, add the routing group to the VPN's VRF instance. To do this, include the rib-group statement at the [edit routing-instances *instance-name* protocols pim] hierarchy level.

For a multicast for Layer 3 VPN example, see "Example: Configuring PIM Sparse Mode over Layer 3 VPNs" on page 267.

Configuring Multicast for Virtual Routers

You can configure PIM for the virtual-router routing 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, but is used for non-VPN-related applications.

The virtual-router instance type has no VRF import, VRF export, VRF target, or route distinguisher requirements. The virtual-router instance type is used for non-Layer 3 VPN situations, for example, to allow the use of IP Security (IPSec) tunnels within VPNs.

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*. For more information about multicast draft support, see "IP Multicast Standards" on page 20. 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.

To configure PIM for a virtual-router instance type, include the pim statement and specify the virtual-router instance type:

```
instance-type virtual-router;
protocols {
    pim {
        ...pim-configuration...
    }
}
```

You can include these statements at the following hierarchy levels:

- [edit routing-instances *routing-instance-name*]
- [edit logical-routers *logical-router-name* routing-instances *routing-instance-name*]

Do not include the vpn-group-address statement for the virtual-router instance type.

Configuration Examples

This section contains the following PIM configuration examples:

Example: Configuring PIM Dense Mode

The following example shows a configuration for PIM dense mode:

```
protocols {
    pim {
      interface so-5/0/1 {
        mode dense;
      }
      interface so-5/0/2 {
        mode dense;
      }
      traceoptions {
        file log-pim;
        flag normal;
        flag state;
      }
    }
}
```

Example: Configuring PIM Sparse Mode

The following example shows a configuration for the RP router and for non-RP routers.

Configuring the RP Router

This example shows a static RP configuration. Add the address statement at the [edit protocols pim rp local] hierarchy level.

For all interfaces, use the mode statement to set the mode to sparse, and use the version statement to set the PIM version to 2 at the [edit protocols PIM rp interface all] hierarchy level. When configuring all interfaces, exclude the fxp0.0 management interface by adding the disable statement for that interface.

NOTE: You do not need to configure Internet Group Management Protocol (IGMP) version 2 for a sparse mode configuration. When PIM is enabled, by default, IGMP version 2 is also enabled.

```
protocols {
    pim {
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
      rp {
        local {
          family inet {
            address 192.168.3.253;
          }
        }
      }
    }
}
```

Configuring All Non-RP Routers

In this example, configure a non-RP router for PIM sparse mode. To specify a static RP address, add the address statement at the [edit protocols pim rp static] hierarchy level. Use the version statement at the [edit protocols pim rp static address] hierarchy level to specify PIM version 2.

Add the mode statement at the [edit protocols pim interface all] hierarchy level to configure the interfaces for sparse mode operation. Then add the version statement at the [edit protocols pim interface all mode] to specify PIM version 2 for all interfaces. When configuring all interfaces, exclude the fxp0.0 management interface by adding the disable statement for that interface.

```
protocols {
    pim {
        rp {
            static {
                address 198.58.3.253 {
                    version 2;
                }
            }
        }
        interface all {
            mode sparse;
                version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Example: Configuring Sparse-Dense Mode

Configure PIM sparse-dense mode on all interfaces, specifying that the groups 224.0.1.39 and 224.0.1.40 are using dense mode:

```
protocols {
    pim {
        dense-groups {
            224.0.1.39;
            224.0.1.40;
        }
        interface all {
            version 1;
            mode sparse-dense;
        }
    }
}
```

Example: Configuring Anycast RP

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 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 goes down, 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 covered in this section.

For information about standards supported for anycast RP, see “IP Multicast Standards” on page 20.

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.

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.

Configuring the RP Router with MSDP

In this example, configure an RP using the lo0 or loopback interface, which is always up. Use the address statement to specify the unique and routable router ID 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).

For all interfaces, use the mode statement to set the mode to sparse and the version statement to specify PIM version 2 at the [edit protocols pim rp local interface all] hierarchy level. When configuring all interfaces, exclude the fxp0.0 management interface by adding the disable statement for that interface.

```
protocols {
    pim {
        rp {
            local {
                family inet;
                address 198.58.3.253;
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

To configure MSDP peering, add the peer statement to configure the address of the MSDP peer at the [edit protocols msdp] hierarchy level. For MSDP peering, use the unique, primary addresses instead of the anycast address. To specify the local address for MSDP peering, add the local-address statement at the [edit protocols msdp peer] hierarchy level.

```
protocols {
    msdp {
      peer 198.58.3.250 {
        local-address address 198.58.3.254;
      }
    }
}
```

Configuring the RP Router Using Only PIM

In this example, configure an RP using the lo0 or 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 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 should be 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, lo0 primary is
                                                      used
                    }
                }
            }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

MSDP configuration is not necessary for this type of IPv4 anycast RP configuration.

Configuring All Non-RP Routers

Whether MSDP is used or not, the anycast RP configuration for a non-RP router is the same as a static RP configuration for a non-RP router. Specify a static RP by adding the address at the [edit protocols pim rp static] hierarchy level. Use the version statement at the [edit protocols pim rp static address] hierarchy level to set PIM version 2.

```
protocols {
    pim {
        rp {
            static {
                address 198.58.3.253 {
                    version 2;
                }
            }
        }
    }
}
```

Use the mode statement at the [edit protocols pim rp interface all] hierarchy level to specify sparse mode on all interfaces. Then add the version statement at the [edit protocols pim rp interface all mode] to configure all interfaces for PIM version 2. When configuring all interfaces, exclude the fxp0.0 management interface by adding the disable statement for that interface.

```
protocols {
    pim {
        interface all {
            mode sparse;
                version 2;
        }
        interface fxp0.0 {
```

```
                disable;
            }
        }
    }
```

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;
    }
}
```

Example: Configuring PIM Join Filters

In this example, you create the PIM join filter by including the import pim-join-filter statement at the [edit protocols pim] hierarchy level. Define pim-join-filter by adding the policy-statement pim-join filter statement at the [edit policy-options] hierarchy level. The filter is composed of a route filter and a source address filter—bad-groups and bad-sources, respectively. Policy bad-groups prevents (*,G) or (S,G) join messages from being received for all groups listed. Policy bad-sources 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.

```
protocols {
    pim {
        import pim-join-filter;
    }
}
policy-statement pim-join-filter {
    term bad-groups {
        from {
            route-filter 224.0.1.2/32 exact;
            route-filter 224.0.1.3/32 exact;
            route-filter 224.0.1.8/32 exact;
            route-filter 224.0.1.22/32 exact;
            route-filter 224.0.1.24/32 exact;
            route-filter 224.0.1.25/32 exact;
            route-filter 224.0.1.35/32 exact;
            route-filter 224.0.1.39/32 exact;
            route-filter 224.0.1.40/32 exact;
            route-filter 224.0.1.60/32 exact;
            route-filter 224.0.2.1/32 exact;
            route-filter 224.0.2.2/32 exact;
```

```
                route-filter 225.1.2.3/32 exact;
                route-filter 229.55.150.208/32 exact;
                route-filter 234.42.42.42/30 orlonger;
                route-filter 239.0.0.0/8 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;
                source-address-filter 172.16.0.0/12 orlonger;
                source-address-filter 192.168.0.0/16 orlonger;
            }
            then reject;
        }
        term last {
            then accept;
        }
}
```

Example: Configuring RP/DR Register Message Filters

Configure an RP filter to drop the register packets for multicast group range 224.1.1.0/24 from source address 10.10.94.1:

```
protocols {
        pim {
            rp {
                rp-register-policy incoming-policy-for-rp;
                local {
                    address 10.10.10.5;
                }
            }
        }
}
policy-options {
        policy-statement incoming-policy-for-rp {
            from {
                router-filter 224.1.1.0/24 orlonger;
                source-address-filter 10.10.94.2/32 exact;
            then reject;
        }
}
```

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:

```
protocols {
    pim {
        rp {
            dr-register-policy outgoing-policy-for-dr;
            static {
                address 10.10.10.3;
            }
        }
    }
}
policy-options {
    policy-statement outgoing-policy-for-rp {
        from {
            router-filter 224.1.1.0/24 orlonger;
            source-address-filter 10.10.10.1/32 exact;
        then reject;
    }
}
```

More complex register message filtering is possible. This example configures a policy expression to accept register messages for multicast group 224.1.1.5 but reject those for 224.1.1.1:

```
protocols {
    pim {
        rp {
            rp-register-policy [ reject_224_1_1_1 | accept_224_1_1_5 ];
            local {
                address 10.10.10.5;
            }
        }
    }
}
policy-options {
    policy-statement reject_224_1_1_1 {
        from {
            router-filter 224.1.1.0/24 orlonger;
            source-address-filter 10.10.94.2/32 exact;
        then reject;
    }
    policy-statement accept_224_1_1_5 {
        term one {
            from {
                router-filter 224.1.1.5/32 exact;
                source-address-filter 10.10.94.2/32 exact;
            }
            then accept;
        }
```

```
            term two {
               then reject;
            }
         }
      }
```

Example: Configuring Externally-Facing 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;
        }
    }
}
```

Example: Tracing PIM Protocol Traffic

Trace only unusual or abnormal operations to a routing log file, and trace detailed information about all PIM messages to a PIM log file:

```
routing-options {
    traceoptions {
        file routing-log;
        flag errors;
    }
}
protocols {
    pim {
        interface so-0/0/0;
        traceoptions {
            file pim-log;
            flag packets;
        }
    }
}
```

Example: Rejecting PIM Bootstrap Messages at the Boundary of a PIM Domain

In this example, the policy statement from interface so-0-1/0 then reject 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;
    }
}
```

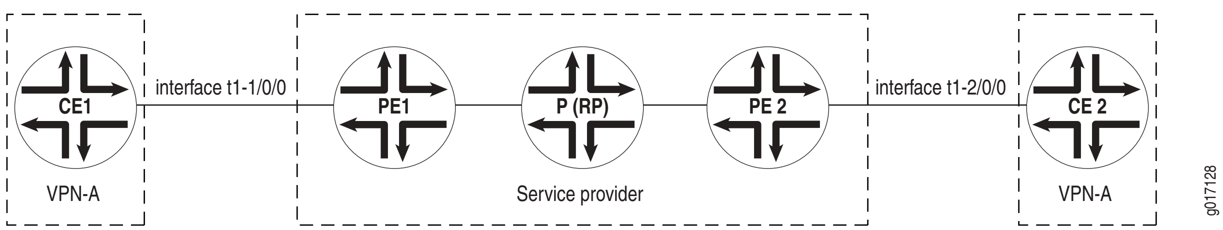
Example: Configuring PIM Sparse Mode over Layer 3 VPNs

This section illustrates how multicast is configured in PIM sparse mode for a multicast range for VPN-A (see Figure 34), and shows how to configure the following:

- Configuring PIM on the P Router on page 267
- Configuring PIM on the PE1 Router on page 268
- Configuring PIM on the PE2 Router on page 268
- Configuring PIM on the CE1 Router on page 269
- Configuring PIM on the CE2 Router on page 269
- Configuring the Routing Instance on the PE1 Router on page 269
- Configuring the Routing Instance on the PE2 Router on page 271
- Configuring the PE Router for Interoperability on page 272
- Configuring the Routing Table Group on page 273

For information about configuring VPNs, see the *JUNOS VPNs Configuration Guide*.

Figure 34: Customer Edge and Service Provider Networks



Configuring PIM on the P Router

Configure PIM on the P router. The P router acts as the P (RP) router in this example. Specify the P router's address (10.255.71.47) at the [edit protocols pim rp local] hierarchy level.

```
protocols {
    pim {
        dense-groups {
            224.0.1.39/32;
            224.0.1.40/32;
        }
        rp {
            local {
                address 10.255.71.47;
            }
        }
        interface all {
            mode sparse;
            version 2;
```

```
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring PIM on the PE1 Router

Configure PIM on the provider edge 1 (PE1) router. Specify a static route to the service provider RP router—the P router (10.255.71.47).

```
protocols {
    pim {
        rp {
            static {
                address 10.255.71.47;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring PIM on the PE2 Router

Configure PIM on the provider edge 2 (PE2) router. Specify a static route to the service provider RP—the P router (10.255.71.47).

```
protocols {
    pim {
        rp {
            static {
                address 10.255.71.47;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring PIM on the CE1 Router

Configure PIM on the customer edge (CE1) router. Specify the RP address for the VPN RP—router CE2 (10.255.245.91).

```
protocols {
    pim {
        rp {
            static {
                address 10.255.245.91;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring PIM on the CE2 Router

Configure PIM on the customer edge 2 (CE2) router, which acts as the VPN RP. Specify router CE2's address (10.255.245.91) at the [edit protocols pim rp local] hierarchy level:

```
protocols {
    pim {
        rp {
            local {
                address 10.255.245.91;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring the Routing Instance on the PE1 Router

Configure the routing instance (VPN-A) for the Layer 3 VPN on router PE1. As part of the configuration, you need to establish the PIM instance for the VPN. Use the vpn-group-address statement at the [edit routing-instances *routing-instance-name* protocols pim] hierarchy level to specify the VPN group address, which is needed for multicast over a Layer 3 VPN configuration.

Set the RP configuration for the VRF instance at the [edit routing-instances *routing-instance-name* protocols pim] hierarchy level. The RP configuration within the VRF instance provides explicit knowledge of the RP address, so that the (*,G) state can be forwarded.

For Release 5.5 or later, configure an additional unit on the loopback interface of the PE router at the [edit interfaces] hierarchy level, and assign an address from the VPN address space. Then add the newly created loopback interface in two places:

- Routing instance (VPN-A) at the [edit routing-instances *routing-instance-name*] hierarchy level.
- Routing instance (VPN-A) at the [edit routing-instances *routing-instance-name* protocols pim] hierarchy level.

Also, add the loopback interface to the IGP and Border Gateway Protocol (BGP) policies to advertise the interface in the VPN address space. For more information about how to configure a logical unit on a loopback interface, see the *JUNOS VPNs Configuration Guide*.

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 with next-hop self configured, 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 the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

```
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;
        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;
                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;
                }
            }
        }
    }
```

```
}
interfaces {
    lo0 {
        description “unit 1 has the important PIM address”
        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;
            }
        }
    }
}
```

NOTE: Multicast Layer 3 VPNs require the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

Configuring the Routing Instance on the PE2 Router

Configure the routing instance (VPN-A) for the Layer 3 VPN on the PE2 router. You need to set the PIM instance for the VPN. Use the vpn-group-address statement at the [edit routing-instances *routing-instance-name* protocols pim] hierarchy level to specify the VPN group address, which is used for multicast over a Layer 3 VPN configuration. As you did for the PE1 router, configure an additional unit on the loopback interface of the PE2 router at the [edit interfaces] hierarchy level and assign an address from the VPN address space.

```
routing-instances {
    VPN-A {
        instance-type vrf;
        interface t1-2/0/0:0.0;
        interface lo0.1;
        route-distinguisher 10.255.71.51:100;
        vrf-import VPNA-import;
        vrf-export VPNA-export;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface t1-2/0/0:0.0;
                    interface lo0.1;
                }
            }
            pim {
                vpn-group-address 239.1.1.1;
                rp {
                    static {
                        address 10.255.245.91;
                    }
                }
```

```
                interface t1-2/0/0:0.0 {
                    mode sparse;
                    version 2;
                interface lo0.1 {
                    mode sparse;
                    version 2;
                }
            }
        }
    }
}
interfaces {
    lo0 {
        description “unit 1 has the important PIM address”
        unit 0 {
            family inet {
                address 192.168.27.14/32;
                    primary;
                address 127.0.0.1/32;
            }
        }
        unit 1 {
            family inet {
                address 10.10.47.102/32;
            }
        }
    }
}
```

NOTE: Multicast Layer 3 VPNs require the BGP next-hop address of the VPN route to match the BGP next-hop address of the loopback VRF instance address.

Configuring the PE Router for Interoperability

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.

```
lo0 {
    description “unit 0 and unit 1 configured for Cisco IOS interoperability”;
    unit 0 {
        family inet {
            address 192.168.27.14/32;
                primary;
            address 127.0.0.1/32;
        }
    }
```

```
        unit 1 {
            family inet {
                address 192.168.27.14/32;
            }
        }
    }
```

Configuring the Routing Table Group

Configure the multicast routing table group by adding the VPNA-mcast-rib statement at the [edit routing-options] hierarchy level. 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.

You must also include the interface routes in inet.2. For more information about creating routing table groups, see the *JUNOS Routing Protocols Configuration Guide*.

```
routing-options {
    interface-routes {
        rib-group VPNA-mcast-rib;
    }
    rib-groups {
        VPNA-mcast-rib {
            export-rib VPN-A.inet.2;
            import-rib VPN-A.inet.2;
        }
    }
}
```

After you configure the multicast routing table group, activate it by including the statement rib-group inet VPNA-mcast-rib at the [edit routing-instances *instance-name* protocols pim] hierarchy level of the VPN's VRF instance.

```
routing-instances {
    VPN-A {
        protocols {
            pim {
                rib-group inet VPNA-mcast-rib;
            }
        }
    }
}
```

Use the following commands to verify the configuration:

- To display all PE tunnel interfaces, issue the command show pim join from the provider router acting as the RP.
- To display multicast tunnel information and the number of neighbors, issue the command show pim interfaces instance *instance-name* 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
```

- To display multicast tunnel interface information, DR information, and the PIM neighbor status between VRF instances on PE1 and PE2, issue the command show pim neighbors instance *instance-name* from either PE router. When issued from the PE1 router, the output display is:

```
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
```

Example: Configuring PIM Dense Mode over Layer 3 VPNs

Multicast over Layer 3 VPNs for dense mode works much the same way as in sparse mode. In the following example, the VPN network uses dense mode for the entire multicast group range. Compare this with the configuration used in "Example: Configuring PIM Sparse Mode over Layer 3 VPNs" on page 267. In that configuration, sparse mode is used for the entire multicast group range.

To support PIM dense mode over Layer 3 VPNs, follow the same steps used in "Example: Configuring PIM Sparse Mode over Layer 3 VPNs" on page 267, with the following differences:

- Configure dense mode for the CE router using the mode statement at the [edit protocols pim interface] hierarchy level. In the example below, the CE-facing interface is t1-1/0/0:0.
- Configure dense mode in the routing instance of the PE router facing the CE router (configured for dense mode) using the mode statement at the [edit routing-instances *instance-name* protocols pim] hierarchy level.
- Remove the RP configurations from the CE router and from the routing instance on the PE router.

This section shows how to do the following tasks:

- Configuring PIM on the P Router on page 275
- Configuring PIM on the PE Router on page 275
- Configuring PIM on the CE Router on page 275
- Configuring the Routing Instance on the PE Router on page 276

For information about configuring VPNs, see the *JUNOS VPNs Configuration Guide*.

Configuring PIM on the P Router

Configure PIM on the P router as in the PIM sparse mode example:

```
protocols {
    pim {
      dense-groups {
        224.0.1.39/32;
        224.0.1.40/32;
      }
      rp {
        local {
          address 10.255.71.47;
        }
      }
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }
}
```

Configuring PIM on the PE Router

Configure PIM on the PE router. Use the mode statement at the [edit protocols pim interface] hierarchy level to specify sparse mode.

```
protocols {
    pim {
      rp {
        static {
          address 10.255.71.47;
        }
      }
      interface all {
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }
}
```

Configuring PIM on the CE Router

Configure PIM on the CE router. Use the mode statement at the [edit protocols pim interface] hierarchy level to specify dense mode. An RP is not used with dense mode, so no RP statements are required on the CE router.

```
protocols {
    pim {
      interface all {
        mode dense;
        version 2;
```

```
            }
            interface fxp0.0 {
                disable;
            }
        }
    }
```

Configuring the Routing Instance on the PE Router

Use the mode statement at the [edit routing-instances instance pim interface] hierarchy level to specify dense mode for interface t1-1/0/0:0.0. An RP is not used with dense mode, so no RP statements are required for the routing instance on the PE router.

```
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;
        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;
                interface t1-1/0/0:0.0 {
                    mode dense;
                    version 2;
                }
                interface lo0.1 {
                    mode dense;
                    version 2;
                }
            }
        }
    }
}
interfaces {
    lo0 {
        description "unit 1 has the important PIM address";
        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;
```

```
            }
        }
    }
}
```

Example: Configuring PIM Sparse-Dense Mode over Layer 3 VPNs

Multicast over Layer 3 VPNs for sparse-dense mode works much the same way as in sparse mode. In the following example, the VPN network uses dense mode for group range 229.0.0.0/8 and sparse mode for the remaining multicast group range outside 229.0.0.0/8. Compare this with the configuration used in "Example: Configuring PIM Sparse Mode over Layer 3 VPNs" on page 267. In that configuration, sparse mode is used for the entire multicast group range.

To support PIM dense mode over Layer 3 VPNs, follow the same steps used in "Example: Configuring PIM Sparse Mode over Layer 3 VPNs" on page 267, with the following differences:

- Configure sparse-dense mode for the CE router and PE router interfaces using the mode statement at the [edit protocols pim interface] hierarchy level. In the example below, the CE-facing interface is t1-1/0/0:0.
- Configure the dense-groups statement to define the desired group range on the CE router at the [edit protocols pim] hierarchy level and in the routing instance at the [edit routing-instances *instance-name* protocols pim] hierarchy level on the PE router.

This section shows how to do the following tasks:

For information about configuring VPNs, see the *JUNOS VPNs Configuration Guide*.

Configuring PIM on the P Router

Configure PIM on the P router as in the PIM sparse mode example:

```
protocols {
    pim {
        dense-groups {
            224.0.1.39/32;
            224.0.1.40/32;
        }
        rp {
            local {
                address 10.255.71.47;
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring PIM on the PE Router

Configure PIM on the PE router. Use the mode statement at the [edit protocols pim interface] hierarchy level to specify sparse-dense mode.

```
protocols {
    pim {
        rp {
            static {
                address 10.255.71.47;
            }
        }
        interface all {
            mode sparse-dense;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring PIM on the CE Router

Configure PIM on the CE router. Use the dense-groups statement at the [edit protocols pim] hierarchy level to define the desired group range on the CE router. Use the mode statement at the [edit protocols pim interface] hierarchy level to specify sparse-dense mode.

```
protocols {
    pim {
        dense-groups {
            229.0.0.0/8;
        }
        rp {
            static {
                address 10.255.245.91;
            }
        }
        interface all {
            mode sparse-dense;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring the Routing Instance on the PE Router

Use the dense-groups statement at the [edit routing-instances *instance-name* protocols pim] hierarchy level to define the desired group range for the routing instance on the PE router. Use the mode statement at the [edit routing-instances instance pim interface] hierarchy level to specify sparse-dense mode for interface t1-1/0/0:0.0.

```
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;
        protocols {
            ospf {
                export bgp-to-ospf;
                area 0.0.0.0 {
                    interface t1-1/0/0:0.0;
                    interface lo0.1;
                }
            }
            pim {
                dense-groups {
                    229.0.0.0/8;
                }
                vpn-group-address 239.1.1.1;
                rp {
                    static {
```

```
                address 10.255.245.91;
            }
        }
        interface t1-1/0/0:0.0 {
            mode sparse-dense;
            version 2;
        }
        interface lo0.1 {
            mode sparse-dense;
            version 2;
        }
      }
    }
  }
}
interfaces {
    lo0 {
      description "unit 1 has the important PIM address";
      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;
        }
      }
    }
}
```

Chapter 31

Summary of PIM Configuration Statements

The following sections explain each of the Protocol Independent Multicast (PIM) configuration statements. The statements are organized alphabetically.

address

See the following sections:

- address (Anycast RPs) on page 281
- address (Local RPs) on page 282
- address (Static RPs) on page 282

address (Anycast RPs)

Syntax
address *address* [forward-msdp-sa];

Hierarchy Level
[edit logical-routers *logical-router-name* protocols pim rp local family anycast-pim rp-set],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp local family anycast-pim rp-set],
[edit protocols pim rp local family anycast-pim rp-set],
[edit routing-instances *routing-instance-name* protocols pim rp local family anycast-pim rp-set]

Release Information
Statement introduced in JUNOS Release 7.4.

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 MSDP connections, then the RP must forward MSP source active (SA) messages.

Options
address—RP address in an RP set.
[forward-msdp-sa]—Forward MSDP SAs to this address.

Usage Guidelines
See "Example: Configuring Anycast RP" on page 258.

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

address (Local RPs)

Syntax address *address*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp local family],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp local family]
[edit protocols pim rp local family],
[edit routing-instances *routing-instance-name* protocols pim rp local family]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the local rendezvous point (RP) address.

Options *address*—Local RP address.

Usage Guidelines See "Configuring the Local RP Address" on page 236.

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

address (Static RPs)

Syntax

```
address address {
    version version;
    group-ranges {
        destination-mask;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim static],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim static],
[edit protocols pim static],
[edit routing-instances *routing-instance-name* protocols pim static]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure static rendezvous point (RP) addresses. You can configure a static RP in a logical router only if the logical router 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.

Usage Guidelines See "Configuring Static RPs" on page 238.

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

anycast-pim

Statement

```
anycast-pim {
    rp-set {
        address address [forward-msdp-sa];
    }
    local-address address;
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim rp local family],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp local family],
[edit protocols pim rp local family],
[edit routing-instances *routing-instance-name* protocols pim rp local family]

Release Information

Statement introduced in JUNOS Release 7.4.

Description

Configure properties for anycast RP using PIM.

Options

The statements are explained separately.

Usage Guidelines

See “Example: Configuring Anycast RP” on page 258.

Required Privilege Level

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

assert-timeout

Syntax

assert-timeout *seconds*;

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

Description

Multicast routers 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 routers determine which router forwards the traffic and prunes the RPT for this group. By default, routers enter an assert cycle every 210 seconds. You can configure this assert timeout between 5 and 210 seconds.

Options

seconds—Time for router to wait before another assert message cycle.
Range: 5 through 210 seconds
Default: 210 seconds

Usage Guidelines

See “Configuring the Assert Timeout” on page 248.

Required Privilege Level

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

auto-rp

Syntax

```
auto-rp {
    (announce | discovery | mapping);
    (mapping-agent-election | no-mapping-agent-election);
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.5.

Description Configure automatic RP announcement and discovery.

Options announce—Configures the router to listen only for mapping packets and also to advertise itself if it is an RP.

discovery—Configures the router to listen only for mapping packets.

mapping—Configures the router to announce, listens for and generates mapping packets, and announces that the router is eligible to be an RP.

The remaining statement is explained separately.

Usage Guidelines See "Configuring Auto-RP" on page 241.

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

bfd-liveness-detection

Syntax

```
bfd-liveness-detection {
    minimum-interval milliseconds;
    minimum-receive-interval milliseconds;
    minimum-transmit-interval milliseconds;
    multiplier number;
    version (0 | 1 | automatic);
}
```

Hierarchy Level [edit protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*]

Release Information Statement introduced in JUNOS Release 8.1.

Description Configure bidirectional forwarding detection (BFD) timers.

Options The statements are explained separately.

Usage Guidelines See "Configuring the BFD Protocol" on page 250.

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

bootstrap

Syntax

```
bootstrap {
    family (inet | inet6) {
        priority number;
        import [ policy-names ];
        export [ policy-names ];
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.6.

Description Configure parameters to control bootstrap routers and messages.

Options The statements are explained separately.

Usage Guidelines See "Configuring the Router's Bootstrap Router Priority" on page 240 and "Filtering PIM Bootstrap Messages" on page 241.

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

bootstrap-export

Syntax bootstrap-export [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.4.

Description Apply one or more export policies to control outgoing PIM bootstrap messages.

Options *policy-names*—Name of one or more import policies.

Usage Guidelines See “Filtering PIM IPv4 Bootstrap Messages” on page 239.

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

See Also bootstrap-import on page 287.

bootstrap-import

Syntax bootstrap-import [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.4.

Description Apply one or more import policies to control incoming PIM bootstrap messages.

Options *policy-names*—Name of one or more import policies.

Usage Guidelines See "Filtering PIM IPv4 Bootstrap Messages" on page 239.

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

See Also bootstrap-export on page 286.

bootstrap-priority

Syntax bootstrap-priority *number*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure whether this router is eligible to be a bootstrap router. In the case of a tie, the router 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 router is not eligible to be the bootstrap router.
Range: 0 through 255
Default: 0

Usage Guidelines See "Configuring PIM Sparse Mode Properties" on page 231.

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

dense-groups

Syntax

```
dense-groups {
    addresses;
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure which groups are operating in dense mode.

Options *addresses*—Operate in dense mode.

Usage Guidelines See "Configuring Sparse-Dense Mode Properties" on page 250.

Required Privilege Level

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

disable

See the following sections:

- disable (PIM Interfaces) on page 288
- disable (PIM Graceful Restart) on page 289

disable (PIM Interfaces)

Syntax disable;

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-name* protocols pim rp local family],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit protocols pim],
[edit protocols pim interface *interface-name*],
[edit protocols pim rp local family],
[edit routing-instances *routing-instance-name* logical-routers *logical-router-name* protocols pim],
[edit routing-instances *routing-instance-name* logical-routers *logical-router-name* protocols pim rp local family],
[edit routing-instances *routing-instance-name* protocols pim],
[edit routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit routing-instances *routing-instance-name* protocols pim rp local family]

Release Information Statement introduced before JUNOS Release 7.4.

Description Explicitly disable PIM.

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

disable (PIM Graceful Restart)

Syntax disable;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim graceful-restart],
[edit logical-routers *logical-router-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 Release 7.4.

Description Explicitly disable PIM sparse mode graceful restart.

Usage Guidelines See “Configuring PIM Sparse Mode Graceful Restart” on page 234.

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

dr-register-policy

Syntax dr-register-policy [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.6.

Description Apply one or more policies to control outgoing PIM register messages.

Options *policy-names*—Name of one or more import policies.

Usage Guidelines See “Configuring RP/DR Register Message Filtering” on page 246.

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

See Also rp-register-policy on page 308.

embedded-rp

Statement

```
embedded-rp {
    maximum-rps limit;
    group-ranges {
      destination-mask;
    }
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure properties for embedded IP version 6 (IPv6) RPs.

The statements are explained separately.

Usage Guidelines See "Configuring Embedded RP for IPv6" on page 247.

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

export

Syntax export [*policy-names*];

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim rp bootstrap family],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp bootstrap family],
[edit protocols pim rp bootstrap family],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap family]

Release Information Statement introduced in JUNOS Release 7.6.

Description Apply one or more export policies to control outgoing PIM bootstrap messages.

Options *policy-names*—Name of one or more import policies.

Usage Guidelines See "Filtering PIM Bootstrap Messages" on page 241.

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

See Also import on page 294.

family

See the following sections:

- family (Bootstrap) on page 291
- family (Local RP) on page 291

family (Bootstrap)

Syntax

```
family (inet | inet6) {
    priority number;
    import [ policy-names ];
    export [ policy-names ];
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim rp bootstrap],
[edit logical-routers *logical-router-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 Release 7.6.

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.

Usage Guidelines

See "Configuring the Router's Bootstrap Router Priority" on page 240 and "Filtering PIM Bootstrap Messages" on page 241.

Required Privilege Level

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

family (Local RP)

Syntax

```
family (inet | inet6) {
    address address;
    anycast-pim {
      rp-set {
        address address [forward-msdp-sa];
      }
      local-address address;
    }
    disable;
    group-ranges {
      destination-mask;
    }
    hold-time seconds;
    priority number;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp local],
[edit logical-routers *logical-router-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 Release 7.4.

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.

Usage Guidelines See “Configuring PIM Sparse Mode Properties” on page 231.

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

graceful-restart

Syntax

```
graceful-restart {
    disable;
    restart-duration seconds;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim graceful-restart],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure PIM sparse mode graceful restart.

Options The statements are explained separately.

Usage Guidelines See “Configuring PIM Sparse Mode Graceful Restart” on page 234.

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

group-ranges

Syntax
group-ranges {
    destination-mask;
}

Hierarchy Level
[edit logical-routers *logical-router-name* protocols pim rp embedded-rp],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name*
    protocols pim rp embedded-rp],
[edit protocols pim rp embedded-rp],
[edit protocols pim rp local family],
[edit protocols pim rp static address *address*],
[edit routing-instances *routing-instance-name* protocols pim rp embedded-rp],
[edit routing-instances *routing-instance-name* protocols pim rp local family],
[edit routing-instances *routing-instance-name* protocols pim rp static address *address*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the address ranges of the multicast groups for which this router can be an RP.

Default The router 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-mask*—Addresses or address ranges for which this router can be an RP.

Usage Guidelines See "Configuring the Groups for Which the Router Is the RP" on page 237 and "Configuring Embedded RP for IPv6" on page 247.

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

hello-interval

Syntax hello-interval *seconds*;

Hierarchy Level
[edit logical-routers *logical-router-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description How often the router sends PIM hello packets out of an interface.

Options *seconds*—Length of time between PIM hello packets.
Range: 0 through 255
Default: 30 seconds

Usage Guidelines See "Modifying the Hello Interval" on page 226.

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

See Also hold-time on page 294

hold-time

Syntax hold-time *seconds*;

Hierarchy Level [edit protocols pim rp local family],
[edit routing-instances *routing-instance-name* protocols pim rp local family]

Release Information Statement introduced before JUNOS Release 7.4.

Description How long a neighbor should consider the sending router (this router) to be operative (up).

Options *seconds*—Hold time.
Range: 0 through 255
Default: 0 seconds

Usage Guidelines See “Modifying the Local RP Hold Time” on page 237.

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

import

See the following sections:

- import (Bootstrap) on page 294
- import (PIM) on page 295

import (Bootstrap)

Syntax import [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp bootstrap family],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp bootstrap family],
[edit protocols pim rp bootstrap family],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap family]

Release Information Statement introduced in JUNOS Release 7.6.

Description Apply one or more import policies to control incoming PIM bootstrap messages.

Options *policy-names*—Name of one or more import policies.

Usage Guidelines See “Filtering PIM Bootstrap Messages” on page 241.

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

See Also export on page 290.

import (PIM)

Syntax import [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

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 from entering the network.

Options *policy-names*—Name of one or more policies.

Usage Guidelines See "Filtering PIM Join Messages" on page 229.

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

infinity

Syntax infinity [*spt-threshold-infinity-policies*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim spt-threshold],
[edit logical-routers *logical-router-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 Release 8.0.

Description Apply one or more policies to set the SPT threshold to infinity for a source-group address pair. Use to prevent the last-hop router from transitioning from the RPT rooted at the RP to an SPT rooted at the source for that source-group address pair.

Options *spt-threshold-infinity-policies*—Name of one or more policies.

Usage Guidelines See "Configuring the SPT Threshold Policy" on page 248.

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

interface

Syntax

```
interface [all | interface-name] {
    disable;
    bfd-liveness-detection {
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        minimum-transmit-interval milliseconds;
```

```
            multiplier number;
            version (0 | 1 | automatic);
        }
        hello-interval seconds;
        mode (dense | sparse | sparse-dense);
        neighbor-policy policy-name;
        priority number;
        version version;
    }
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

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. For details about specifying interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

The remaining statements are explained separately.

Usage Guidelines See "Configuring PIM Mode-Independent Interface Properties" on page 225.

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

local

Syntax

```
local {
    family (inet | inet6) {
        address address;
        anycast-pim {
            rp-set {
                address address [forward-msdp-sa];
            }
            local-address address;
        }
        disable;
        group-ranges {
            destination-mask;
        }
        hold-time seconds;
        priority number;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure the router's RP properties.

Options The statements are explained separately.

Usage Guidelines See "Configuring the Router's Local RP Properties" on page 235.

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

local-address

Syntax local-address *address*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp local family anycast-pim],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp local family anycast-pim],
[edit protocols pim rp local family anycast-pim],
[edit routing-instances *routing-instance-name* protocols pim rp local family anycast-pim]

Release Information Statement introduced in JUNOS Release 7.4.

Description Configure the router's local address for 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.

Usage Guidelines See "Example: Configuring Anycast RP" on page 258.

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

mapping-agent-election

Syntax (mapping-agent-election | no-mapping-agent-election);

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp auto-rp],
[edit logical-routers *logical-router-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 Release 7.5.

Description Configure the router's mapping announcements as a mapping agent.

- 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

Usage Guidelines See “Configuring Auto-RP Mapping Agent Election” on page 245.

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

maximum-rps

Statement maximum-rps *limit*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp embedded-rp],
[edit logical-routers *logical-router-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 Release 7.4.

Description Limit the number of RPs that the routing platform acknowledges.

Options *limit*—Number of RPs.
Range: 1 through 500
Default: 100

Usage Guidelines See “Configuring Embedded RP for IPv6” on page 247.

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

minimum-interval

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 Release 8.1.

Description Configure the bidirectional forwarding detection (BFD) minimum interval timer. This timer specifies the same value for both the minimum transmit interval and minimum receive interval for the bfd-liveness-detection statement.

Options *milliseconds*—Minimum transmit and receive interval.
Range: 1 through 255,000 milliseconds

Usage Guidelines See "Configuring the BFD Protocol" on page 250.

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

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 Release 8.1.

Description Configure the bidirectional forwarding detection (BFD) minimum receive interval timer. This timer specifies only the minimum receive interval for the bfd-liveness-detection statement.

Options *milliseconds*—Minimum receive interval.
Range: 1 through 255,000 milliseconds

Usage Guidelines See "Configuring the BFD Protocol" on page 250.

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

minimum-transmit-interval

Syntax minimum-transmit-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 Release 8.1.

Description Configure the bidirectional forwarding detection (BFD) minimum transmit interval timer. Configuring this timer specifies only the minimum transmit interval for the bfd-liveness-detection statement.

Options *milliseconds*—Minimum transmit interval.
Range: 1 through 255,000 milliseconds

Usage Guidelines See "Configuring the BFD Protocol" on page 250.

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

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 Release 8.1.

Description Configure the multiplier for bidirectional forwarding detection (BFD) timers.

Options *number*—Detection time multiplier.
Range: 1 through 255
Default: 3

Usage Guidelines See "Configuring the BFD Protocol" on page 250.

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

mode

Syntax mode (dense | sparse | sparse-dense);

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure PIM to operate in sparse, dense, or sparse-dense mode.

Options dense—Operate in dense mode.

sparse—Operate in sparse mode.

sparse-dense—Operate in sparse-dense mode.

Default: sparse

Usage Guidelines See “Configuring PIM Dense Mode Properties” on page 231, “Configuring PIM Sparse Mode Properties” on page 231, and “Configuring Sparse-Dense Mode Properties” on page 250.

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

neighbor-policy

Syntax neighbor-policy *policy-name*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-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 Release 8.2.

Description Apply a PIM interface-level policy to filter neighbor IP addresses.

Options *policy-name*—Name of the policy that filters neighbor IP addresses. For details about configuring policy statements, see the *JUNOS Policy Framework Configuration Guide*.

Usage Guidelines See “Configuring Interface-Level Neighbor Policies” on page 227.

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

pim

Syntax

```
pim {
    assert-timeout seconds;
    dense-groups {
      addresses;
    }
    disable;
    import [ policy-names ];
    interface interface-name {
      disable;
      bfd-liveness-detection {
        minimum-interval milliseconds;
        minimum-receive-interval milliseconds;
        minimum-transmit-interval milliseconds;
        multiplier number;
        version (0 | 1 | automatic);
      }
      hello-interval seconds;
      mode (dense | sparse | sparse-dense);
      neighbor-policy policy-name;
      priority number;
      version version;
    }
```

```
rib-group group-name;
rp {
    auto-rp {
        (announce | discovery | mapping);
        (mapping-agent-election | no-mapping-agent-election);
    }
    bootstrap {
        family (inet | inet6) {
            priority number;
            import [ policy-names ];
            export [ policy-names ];
        }
    }
    bootstrap-import [ policy-names ];
    bootstrap-export [ policy-names ];
    bootstrap-priority number;
    dr-register-policy [ policy-names ];
    embedded-rp {
        maximum-rps limit;
        group-ranges {
            destination-mask;
        }
    }
    local {
        family (inet | inet6) {
            disable;
            address address;
            anycast-pim {
                rp-set {
                    address address [forward-msdp-sa];
                }
                local-address address;
            }
            group-ranges {
                destination-mask;
            }
            hold-time seconds;
            priority number;
        }
    }
    rp-register-policy [ policy-names ];
    spt-threshold {
        infinity [ spt-threshold-infinity-policies ];
    }
    static {
        address address {
            version version;
            group-ranges {
                destination-mask;
            }
        }
    }
}
```

```
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
    }
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols],
[edit protocols],
[edit routing-instances *routing-instance-name* protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable PIM on the router.

Default PIM is disabled on the router.

Options The statements are explained separately.

Usage Guidelines See "PIM Configuration Guidelines" on page 223.

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

priority

See the following sections:

- priority (Bootstrap) on page 304
- priority (PIM Interfaces) on page 304
- priority (PIM RPs) on page 305

priority (Bootstrap)

Syntax priority *number*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp bootstrap family],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim rp bootstrap family],
[edit protocols pim rp bootstrap family],
[edit routing-instances *routing-instance-name* protocols pim rp bootstrap family]

Release Information Statement introduced in JUNOS Release 7.6.

Description Configure the router's likelihood to be elected as the bootstrap router.

Options *number*—Router'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 router has the least likelihood of becoming the bootstrap router and sends packets with a priority of 0.)

Usage Guidelines See "Configuring the Router's Bootstrap Router Priority" on page 240.

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

See Also bootstrap-priority on page 287.

priority (PIM Interfaces)

Syntax priority *number*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure the router's likelihood to be elected as the designated router.

Options *number*—Router's priority for becoming the designated router. A higher value corresponds to a higher priority.
Range: 1 through a 32-bit number
Default: 1 (The router has the least likelihood of becoming the designated router.)

Usage Guidelines See "Configuring the Designated Router Priority" on page 226.

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

priority (PIM RPs)

Syntax priority *number*;

Hierarchy Level [edit protocols pim rp local family],
[edit routing-instances *routing-instance-name* protocols pim rp local family]

Release Information Statement introduced before JUNOS Release 7.4.

Description This router's priority for becoming an RP. The bootstrap router uses this field when selecting the list of candidate RPs to send in the bootstrap message. A smaller number increases the likelihood that the router becomes the RP for local multicast groups. A priority value of 0 means that bootstrap router can override the group range being advertised by the candidate RP.

Options *number*—Router's priority for becoming an RP. A lower value corresponds to a higher priority.
Range: 0 through 255
Default: 1

Usage Guidelines See "Configuring the Router's Local RP Properties" on page 235.

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

restart-duration

Syntax restart-duration *seconds*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim graceful-restart],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure the duration of the graceful restart interval.

Options *seconds*—Time the routing platform waits to complete PIM sparse mode graceful restart.
Range: 30 through 300
Default: 60

Usage Guidelines See "Configuring PIM Sparse Mode Graceful Restart" on page 234.

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

rib-group

Syntax rib-group *group-name*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

Description Associate a routing table group with PIM.

Options *group-name*—Name of the routing table group. The name must be one that you defined with the rib-group statement at the [edit routing-options] hierarchy level.

Usage Guidelines See "Configuring a PIM RPF Routing Table" on page 228.

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

rp

Syntax

```
rp {
    auto-rp {
        (announce | discovery | mapping);
        (mapping-agent-election | no-mapping-agent-election);
    }
    bootstrap {
        family (inet | inet6) {
            priority number;
            import [ policy-names ];
            export [ policy-names ];
        }
    }
    bootstrap-export [ policy-names ];
    bootstrap-import [ policy-names ];
    bootstrap-priority number;
    dr-register-policy [ policy-names ];
    embedded-rp {
        maximum-rps limit;
        group-ranges {
            destination-mask;
        }
    }
    local {
        family (inet | inet6) {
            disable;
            address address;
            anycast-pim {
                rp-set {
                    address address [forward-msdp-sa];
                }
                local-address address;
```

```
            }
            group-ranges {
               destination-mask;
            }
            hold-time seconds;
            priority number;
         }
      }
      rp-register-policy [ policy-names ];
      static {
         address address {
            version version;
            group-ranges {
               destination-mask;
            }
         }
      }
   }
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

Description Configure the router as an actual or potential RP. A router can be an RP for more than one group.

Default If you do not include the rp statement, the router can never become the RP.

Options The statements are explained separately.

Usage Guidelines See “Configuring PIM Sparse Mode Properties” on page 231.

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

rp-register-policy

Syntax rp-register-policy [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.6.

Description Apply one or more policies to control incoming PIM register messages.

Options *policy-names*—Name of one or more import policies.

Usage Guidelines See "Configuring RP/DR Register Message Filtering" on page 246.

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

See Also dr-register-policy on page 289.

rp-set

Syntax

```
rp-set {
    address address [forward-msdp-sa];
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim local family anycast-pim],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim local family anycast-pim],
[edit protocols pim local family anycast-pim],
[edit routing-instances *routing-instance-name* protocols pim local family anycast-pim]

Release Information Statement introduced in JUNOS Release 7.4.

Description Configure a set of rendezvous point (RP) addresses for anycast RP. You can configure up to 15 RPs.

Options The statement is explained separately.

Usage Guidelines See "Example: Configuring Anycast RP" on page 258.

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

spt-threshold

Syntax

```
spt-threshold {
    infinity [ spt-threshold-infinity-policies ];
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 8.0.

Description Last-hop multicast routers running PIM sparse mode can forward the same stream of multicast packets onto the same LAN through a RPT rooted at the RP or a SPT rooted at the source. By default, last-hop routers transition to a direct SPT to the source. You can configure this router to set the SPT transition value to infinity to prevent this transition for any source-group address pair.

Options The statement is explained separately.

Usage Guidelines See "Configuring the SPT Threshold Policy" on page 248.

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

static

Syntax

```
static {
    address address {
        version version;
        group-ranges {
            destination-mask;
        }
    }
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols pim rp],
[edit logical-routers *logical-router-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 Release 7.4.

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 router only if the logical router 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 The statements are explained separately.

Usage Guidelines See “Configuring Static RPs” on page 238.

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

traceoptions

Syntax

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim],
[edit logical-routers *logical-router-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 Release 7.4.

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 *name*—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 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.

PIM Tracing Flags

- assert—Assert messages
- 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
- packets—All PIM packets
- prune—Prune messages
- register—Register and register stop messages
- rp—Candidate RP advertisements

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 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.

Usage Guidelines See “Tracing IGMP Protocol Traffic” on page 56, “Tracing DVMRP Protocol Traffic” on page 180, “Configuring PIM Trace Options” on page 230, and “Tracing MSDP Protocol Traffic” on page 335.

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.

version

See the following sections:

- version (PIM) on page 312
- version (BFD) on page 313

version (PIM)

Statement version *version*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-name* protocols pim rp static address *address*],
[edit logical-routers *logical-router-name* routing-instances *routing-instance-name* protocols pim interface *interface-name*],
[edit logical-routers *logical-router-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 Release 7.4.

Description Specify the version of PIM.

Options *version*—PIM version number.

Range: 1 or 2

Default: PIM version 2

Usage Guidelines See “Changing the PIM Version” on page 226.

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

version (BFD)

Syntax version (0 | 1 | automatic);

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 Release 8.1.

Description Specify the bidirectional forwarding detection (BFD) protocol version that you want to detect.

Options 0—BFD version 0; this version is deprecated.

1—BFD version 1.

automatic—Auto-detect the BFD version (default).

Usage Guidelines See “Configuring the BFD Protocol” on page 250.

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

vpn-group-address

Syntax vpn-group-address *address*;

Hierarchy Level [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols pim],
[edit routing-instances *instance-name* protocols pim]

Release Information Statement introduced before JUNOS Release 7.4.

Description Specify a group address on which to encapsulate multicast traffic from a virtual private network (VPN) instance.

Options *address*—IP address 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. For more information about addresses, see “Multicast Addresses” on page 22.

Usage Guidelines See “Configuring Multicast for Layer 3 VPNs” on page 251.

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

Part 11
MSDP

Chapter 32

MSDP Overview

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 Border Gateway Protocol (BGP). 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 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 this peer is toward 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 “Configuring MSDP Mesh Groups” on page 324.

Router R determines 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 Border Gateway Protocol (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 (IBPG) 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.

For more information about PIM sparse mode, see "Configuring PIM Sparse Mode Properties" on page 231.

The MSDP peers that receive source-active TLVs can be constrained by BGP reachability information. If the AS path of the network layer reachability information (NLRI) contains the receiving peer's AS number prepended second to last, the sending peer is using the receiving peer as a next hop for this source. If the split horizon information is not being received, the peer can be pruned from the source-active TLV distribution list.

For information about standards supported for MSDP, see "IP Multicast Standards" on page 20.

Chapter 33
MSDP Configuration Guidelines

To configure the Multicast Source Discovery protocol (MSDP), include the msdp statement:

```
msdp {
    active-source-limit {
        maximum number;
        threshold number;
    }
    data-encapsulation (disable | enable);
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    rib-group group-name;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    peer address {
        active-source-limit {
            maximum number;
            threshold number;
        }
        authentication-key peer-key;
        default-peer;
        disable;
        export [ policy-names ];
        import [ policy-names ];
        local-address address;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <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);
        traceoptions {
```

```
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
        peer address; {
            active-source-limit {
                maximum number;
                threshold number;
            }
            authentication-key peer-key;
            default-peer;
            disable;
            export [ policy-names ];
            import [ policy-names ];
            local-address address;
            traceoptions {
                file name <replace> <size size> <files number> <no-stamp>
                    <world-readable | no-world-readable>;
                flag flag <flag-modifier> <disable>;
            }
        }
        source prefix/mask {
            active-source-limit {
                maximum number;
                threshold number;
            }
        }
    }
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit routing-instances *instance-name* protocols]
- [edit logical-routers *logical-router-name* protocols]
- [edit routing-instances *instance-name* logical-routers *logical-router-name* protocols]

For an overview of logical routers and a detailed example of logical router configuration, see the logical routers chapter of the *JUNOS Feature Guide*.

By default, MSDP is disabled.

This chapter describes the following tasks for configuring MSDP:

- Minimum MSDP Configuration on page 321
- Enabling MSDP on page 322
- Configuring MSDP Peers on page 322
- Configuring MSDP Groups on page 323
- Configuring MSDP Mesh Groups on page 324

For a configuration example, see "Example: Configuring MSDP" on page 336.

Minimum MSDP Configuration

To enable MSDP on the router, include at least the following statements:

```
msdp {
    local-address address;
    peer address;
}
```

You can include these statements at the following hierarchy levels:

- [edit protocols]
- [edit routing-instances *instance-name* protocols]
- [edit logical-routers *logical-router-name* protocols]
- [edit routing-instances *instance-name* logical-routers *logical-router-name* protocols]

You must configure at least one peer. The peer and the local-address statements are required. You should also configure the router to be a Protocol Independent Multicast (PIM) sparse-mode rendezvous point (RP). For more information about configuring PIM, see "PIM Configuration Guidelines" on page 223.

Enabling MSDP

To enable MSDP peering on the router, include the msdp statement:

```
msdp {
    local-address address;
    peer address;
    rib-group group-name;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols]
- [edit logical-routers *logical-router-name* protocols]
- [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols]
- [edit routing-instances *instance-name* protocols]

To associate with MSDP a routing table group that imports and exports routes into the specified routing table group, include the rib-group statement. The routing table group is a group that you defined with the rib-groups statement at the [edit routing-options] hierarchy level. For more information, see the *JUNOS Routing Protocols Configuration Guide*.

Configuring MSDP Peers

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.

To configure MSDP peers, include the peer statement:

```
peer address {
    active-source-limit {
      maximum number;
      threshold number;
    }
    authentication-key peer-key;
    default-peer;
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
      file name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
}
```

The peer and the local-address statements are required.

You can configure MSDP peers globally or in a group.

- Globally for all MSDP peers at the following hierarchy levels:
  - [edit protocols msdp]
  - [edit logical-routers *logical-router-name* protocols msdp]
  - [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp]
  - [edit routing-instances *instance-name* protocols msdp]
- In a group at the following hierarchy levels:
  - [edit protocols group *group-name*]
  - [edit logical-routers *logical-router-name* protocols group *group-name*]
  - [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols group *group-name*]
  - [edit routing-instances *instance-name* protocols group *group-name*]

If you configure MSDP peers in a group, each individual peer in a group inherits all group-level options.

Configuring MSDP Groups

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.

To configure MSDP groups, include one or more of the following statements:

```
group group-name {
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    mode <mesh-group | standard>;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    peer address; {
        active-source-limit {
            maximum number;
            threshold number;
        }
        authentication-key peer-key;
        default-peer;
        disable;
```

```
            export [ policy-names ];
            import [ policy-names ];
            local-address address;
            traceoptions {
                file name <replace> <size size> <files number> <no-stamp>
                    <world-readable | no-world-readable>;
                flag flag <flag-modifier> <disable>;
            }
        }
    }
```

You can include these statements at the following hierarchy levels:

- [edit protocols msdp]
- [edit logical-routers *logical-router-name* protocols msdp]
- [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp]
- [edit routing-instances *instance-name* protocols msdp]

The local-address and peer statements are mandatory. The individual statements are discussed in separate sections.

Configuring MSDP Mesh Groups

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.

To configure an MSDP mesh group, define a peer group, and include the mode mesh-group statement:

```
group group-name {
    local-address address;
    mode mesh-group;
    peer address;
}
```

You can include this statement at the following hierarchy levels:

- [edit protocols msdp]
- [edit logical-routers *logical-router-name* protocols msdp]
- [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp]
- [edit routing-instances *instance-name* protocols msdp]

CAUTION: When configuring MSDP mesh groups, you must configure all members the same. If you do not configure a full mesh, excessive flooding of source-active messages can occur.

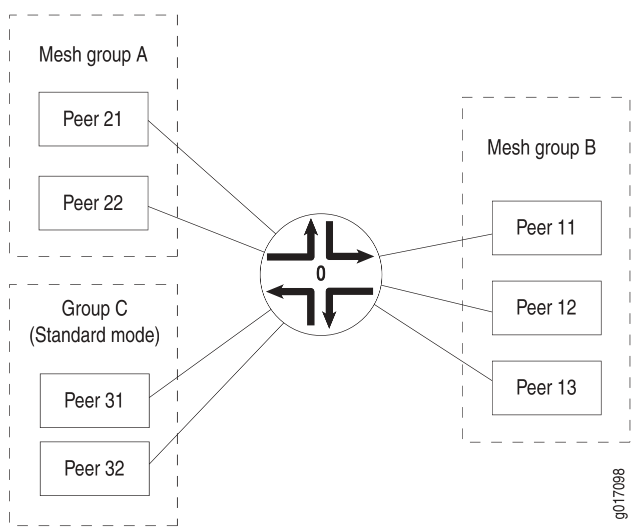
A common application for MSDP mesh groups is peer-reverse-path-forwarding (peer-RPF) check bypass. For example, if there are two MSDP peers inside an autonomous system (AS), and only one of them has an external MSDP session to another AS, the internal MSDP peer often rejects incoming source-active messages relayed by the peer with the external link. Rejection occurs because the external MSDP peer must be reachable by the internal MSDP peer via 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.

Table 10 explains how flooding is handled by peers in this configuration. Figure 35 illustrates source-active message flooding between different mesh groups and peers within the same mesh group.

Table 10: 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 35: Source-Active Message Flooding



Configuring the MSDP 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 at the [edit protocols msdp] and [edit protocols msdp group] hierarchy levels, the authentication key configured at the [edit protocols msdp] hierarchy level is used.

To configure MSDP authentication keys on the router, include the authentication-key statement:

```
authentication-key peer-key;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

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 (" ").

The following example configures the MSDP authentication key grandmother for MSDP peer 10.0.0.1, and the MSDP authentication keys New York and phoenix5 for peers 172.16.0.1 and 192.168.0.1 in MSDP group msdp-one:

```
msdp {
    group msdp-one {
      peer 171.16.0.1 {
        authentication-key “New York”;
        local-address 10.100.0.2;
      }
      peer 192.168.0.1 {
        authentication-key phoenix5;
        local-address 10.100.0.2;
      }
    peer 10.0.0.1 {
      authentication-key grandmother;
      local-address 10.100.0.2;
    }
}
```

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.

Configuring MSDP Routing Policy

All routing protocols use the routing table to store the routes that they learn and to determine which routes they should advertise in their protocol packets. Routing policy allows you to control which routes the routing protocols store in, and retrieve from, the routing table. For information about routing policy, see the *JUNOS Routing Protocols Configuration Guide*.

You can configure routing policy globally, for a group, or for an individual peer:

- Globally for all MSDP peers at the [edit protocols msdp] or [edit logical-routers *logical-router-name* protocols msdp] hierarchy level.
- For all peers in a group at the [edit protocols msdp group *group-name*] or [edit logical-routers *logical-router-name* protocols msdp group *group-name*] level.
- For an individual peer at the [edit protocols msdp peer *address*] or [edit logical-routers *logical-router-name* protocols msdp peer *address*] level, or the [edit protocols msdp group *group-name* peer *address*] or [edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*] level.

If you configure routing policy at the group level, each individual peer in a group inherits the group's routing policy.

To apply policies to source-active messages being imported into the source-active cache from MSDP, include the import statement, listing the names of one or more policy filters to be evaluated. See Table 11 for a list of match conditions.

Table 11: MSDP Source-Active Message Filter Match Conditions

Match Condition	Matches On
interface	Router interface or interfaces specified by name or IP address
neighbor	Neighbor address (the source address in the IP header of the source-active message)
route-filter	Multicast group address embedded in the source-active message
source-address-filter	Multicast source address embedded in the source-active message

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, MSDP shares with the routing table only those routes that were learned from MSDP routers.

```
import [ policy-names ];
```

To apply policies to source-active messages being exported from the source-active cache into MSDP, include the export statement, listing the names of one or more policies to be evaluated. 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 source-active cache entry. If no match is found, the default MSDP export policy is applied to entries in the source-active cache.

```
export [ policy-names ];
```

Configuring Multiple Rendezvous Points in a Domain

You can configure multiple RPs in a shared-tree PIM sparse-mode domain. You need to configure an MSDP local address to enable the RPs in the domain to maintain a consistent view of the active sources.

To configure a router to act as an RP in a domain with other RPs, do the following for each router in the domain that acts as an RP:

- Create the router ID by configuring a unique and routable IP address on the loopback interface and setting the primary address flag.
- Configure a non-unique, but routable, unicast address on the loopback interface.
- Use the non-unique, routable unicast address to configure the PIM router to be the local RP.
- Configure MSDP with the unique and routable address (router ID) as the local address of the peer.

For a sample configuration of multiple RPs, see "Example: Configuring a Router to Use Anycast RP" on page 328. For more information about configuring interfaces, see the *JUNOS Network Interfaces Configuration Guide*.

Example: Configuring a Router to Use Anycast RP

The following example configures a router to use anycast RP:

```
[edit]
interfaces {
    ...

    lo0 {
        unit 0 {
            family inet {
                unique routable address [and] router-id;
                address 10.1.1.1/32 {
                    primary;
                }
                non-unique routable anycast RP address;
                address 10.10.10.10/32;
                address 127.0.0.1/32;
            }
        }
    }
```

```
}

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;
        }
    }
    autonomous-system 1234;
}

protocols {
    bgp {
        group red {
            type internal;
            family inet any;
            neighbor 10.1.1.2 {
                local-address 10.1.1.1;
            }
        }
    }
    msdp {
        rib-group mcrg;
        group red {
            peer 10.1.1.2 {
                local-address 10.1.1.1;
            }
        }
    }
    pim {
        dense-groups {
            224.0.1.39/32;
            224.0.1.40/32;
        }
        rib-group mcrg;
        rp {
            local {
                address 10.10.10.10;
            }
        }
        interface all {
            mode sparse-dense;
            version 2;
        }
        interface fxp0.0 {
            disable;
        }
    }
}
```

Configuring MSDP Data Encapsulation

MSDP data encapsulation mainly concerns bursty sources of multicast traffic. Sources that send only one packet every few minutes have problems 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. So 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 denial-of-service (DoS) attack, the router's forwarding table may 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.

To configure MSDP data encapsulation on the router, include the data-encapsulation statement:

```
data-encapsulation (enable | disable);
```

You can include this statement at the following hierarchy levels:

- [edit protocols msdp]
- [edit logical-routers *logical-router-name* protocols msdp]
- [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp]
- [edit routing-instances *instance-name* protocols msdp]

You should also configure the router to be a PIM sparse-mode RP. For more information about configuring PIM, see "PIM Configuration Guidelines" on page 223.

Configuring the MSDP Active Source Limit

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 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.

By default, the router accepts 25,000 source active messages before ignoring the rest to prevent a possible DoS attack. The limit can be from 1 to 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 1,000 messages are screened by the RED profile and the accepted messages processed.

To configure the MSDP active source limit on the router, include the active-source-limit statement:

```
active-source-limit {
    maximum number;
    threshold number;
}
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

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 should be less than the number configured for the maximum number of active MSDP sources.

You can configure an active source limit at several levels of the MSDP hierarchy:

- Configuring Global, Group, and Peer Active Source Limit on page 332
- Configuring Per-Source Active Source Limit on page 332

Configuring Global, Group, and Peer Active Source Limit

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, all are applied.

The following example applies a limit of 5,000 active sources to MSDP peer 10.0.0.1, a limit of 7,500 active sources to MSDP peer 10.10.10.10 in group MSDP-group, and a limit of 10,000 active sources to all others.

```
[edit protocols msdp]
active-source-limit {
    maximum 10000;
}
group MSDP-group {
    peer 10.10.10.10;
      active-source-limit {
        maximum 7500;
      }
    peer 10.10.10.11;
}
peer 10.0.0.1 {
    active-source-limit {
      maximum 5000;
    }
}
```

Configuring Per-Source Active Source Limit

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.

```
[edit protocols msdp]
source 10.1.1.1/32 {
    active-source-limit {
      maximum 10000;
    }
}
source 10.1.0.0/16 {
    active-source-limit {
      maximum 500;
    }
}
source 0.0.0.0/0 {
    active-source-limit {
      maximum 5;
    }
}
```

In this example, the source 10.1.1.1 is allowed active sources for 10,000 groups. Any other source on the 10.1.0.0/16 network is allowed 500 groups. All other sources are allowed to source 5 active streams.

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 5,000 (and there are no other sources or limits configured), only 5,000 active source messages are accepted from this source.

Configuring a Default MSDP Peer

When a source-active message is received, a peer-RPF check is performed to make sure the peer is leading toward the originating RP and to decide whether the source-active message should be accepted. However, in networks with only one MSDP peer, especially stub networks, there is no question that the source-active message should be accepted. An MSDP default peer is an MSDP peer from which all source-active messages are accepted without performing the peer-RPF check.

To establish an MSDP peer as the default peer, include the default-peer statement:

```
default-peer;
```

For a list of the hierarchy levels at which you can include this statement, see the statement summary section for this statement.

You can establish a default peer at the peer or group level. For more information about MSDP peers and peer-RPF checks, see "MSDP Overview" on page 317.

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-routers *logical-router-name* protocols msdp]
  - [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp]
  - [edit routing-instances *instance-name* protocols msdp]
- For all peers in a group at the following hierarchy levels:
  - [edit protocols msdp group *group-name*]
  - [edit logical-routers *logical-router-name* protocols msdp group *group-name*]
  - [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp group *group-name*]
  - [edit routing-instances *instance-name* protocols msdp group *group-name*]
- For an individual peer at the following hierarchy levels:
  - [edit protocols msdp peer *address*]
  - [edit logical-routers *logical-router-name* protocols msdp peer *address*]
  - [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp peer *address*]
  - [edit routing-instances *instance-name* protocols msdp peer *address*]
  - [edit protocols msdp group *group-name* peer *address*]
  - [edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*]
  - [edit logical-routers *logical-router-name* routing-instances *instance-name* protocols msdp group *group-name* peer *address*]
  - [edit routing-instances *instance-name* protocols msdp group *group-name* peer *address*]

If you disable MSDP at the group level, each peer in the group is disabled.

Tracing MSDP Protocol Traffic

To trace MSDP protocol traffic, you can specify options in the global traceoptions statement at the [edit routing-options] or [edit logical-routers *logical-router-name* routing-options] hierarchy level, and you can specify MSDP-specific options by including the traceoptions statement. Options applied at the routing options level trace all packets, and options applied at the protocol level trace only IGMP traffic.

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

You can configure MSDP tracing as follows:

- Globally for all MSDP peers at the [edit protocols msdp] hierarchy level
- For all peers in a group at the [edit protocols msdp group *group-name*] level
- For an individual peer at the [edit protocols msdp peer *address*] or the [edit protocols msdp group *group-name* peer *address*] level

If you configure tracing options at the group level, each peer in the group inherits the group's tracing options.

You can configure tracing options globally for all MSDP peers (at the [edit protocols msdp] hierarchy level), for all peers in a group (at the [edit protocols msdp group *group-name*] level), or for an individual peer (at the [edit protocols msdp peer *address*] or the [edit protocols msdp group *group-name* peer *address*] level). If you configure tracing options at the group level, each peer in the group inherits the group's tracing options.

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

- keepalive—Trace keepalive messages.
- packets—Trace all MSDP packets.
- route—Trace MSDP changes to the routing table.
- sa—Trace source-active packets.
- sa-request—Trace source-active request packets.
- sa-response—Trace source-active response packets.

For general information about tracing, see the *JUNOS System Basics Configuration Guide*.

Example: Tracing MSDP Protocol Traffic

Trace only unusual or abnormal operations to routing-log, and trace detailed information about all MSDP messages to msdp-log:

```
[edit]
routing-options {
    traceoptions {
        file routing-log;
        flag errors;
    }
}
protocols {
    msdp {
        peer 192.68.2.120; {
        local-address 192.68.1.200;
        }
        traceoptions {
            file msdp-log;
            flag packets;
        }
    }
}
```

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;
            }
        }
    }
}
```

Chapter 34

Summary of MSDP Configuration Statements

The following sections explain each of the Multicast Source Discovery Protocol (MSDP) configuration statements. The statements are organized alphabetically.

active-source-limit

Syntax

```
active-source-limit {
      maximum number;
      threshold number;
}
```

Hierarchy Level

[edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*
 peer *address*],
[edit logical-routers *logical-router-name* protocols msdp peer *address*],
[edit logical-routers *logical-router-name* protocols msdp source *prefix/mask*],
[edit protocols msdp],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit protocols msdp source *prefix/mask*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols
 msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols
 msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols
 msdp peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols
 msdp source *prefix/mask*],
[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*],
[edit routing-instances *instance-name* protocols msdp source *prefix/mask*]

Release Information Statement introduced before JUNOS Release 7.4.

Description Limit the number of active source messages the router 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.

Usage Guidelines See "Configuring the MSDP Active Source Limit" on page 331.

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

authentication-key

Syntax authentication-key *peer-key*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-name* protocols msdp peer *address*],
[edit protocols msdp group *group-name* peer *address*],
[edit protocols msdp peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp peer *address*],
[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 before JUNOS Release 7.4.

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 (" ").

Usage Guidelines See "Configuring the MSDP Authentication Key" on page 326.

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

data-encapsulation

Syntax data-encapsulation (disable | enable);

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp]

Release Information Statement introduced before JUNOS Release 7.4.

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

Usage Guidelines See "Configuring MSDP Data Encapsulation" on page 330.

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

default-peer

Syntax default-peer;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-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 *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*],
[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 before JUNOS Release 7.4.

Description Establishes this peer as the default MSDP peer and accepts source-active messages from the peer without the usual peer-reverse-path-forwarding (peer-RPF) check.

Usage Guidelines See "Configuring a Default MSDP Peer" on page 333.

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

disable

Syntax disable;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-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 *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*],
[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 before JUNOS Release 7.4.

Description Explicitly disable MSDP.

Usage Guidelines See "Disabling MSDP" on page 334.

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

export

Syntax export [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-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 *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols

msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*],
[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 before JUNOS Release 7.4.

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.

Usage Guidelines See "Configuring MSDP Routing Policy" on page 327.

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

See Also import on page 344

group

Syntax

```
group group-name {
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    mode (mesh-group|standard);
    traceoptions {
      file name <replace> <size size> <files number> <no-stamp>
        <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
    peer address; {
      active-source-limit {
        maximum number;
        threshold number;
      }
      authentication-key peer-key;
      default-peer;
      disable;
      export [ policy-names ];
      import [ policy-names ];
      local-address address;
      traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
          <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
      }
    }
}
```

Hierarchy Level [edit protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Define an MSDP peer group. MSDP peers within groups share common traceoptions, 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.

Usage Guidelines See "Configuring MSDP Groups" on page 323.

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

import

Syntax import [*policy-names*];

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-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 *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*],
[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 before JUNOS Release 7.4.

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.

Usage Guidelines See “Configuring MSDP Routing Policy” on page 327.

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

See Also export on page 342

local-address

Syntax local-address *address*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-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 *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*],
[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 before JUNOS Release 7.4.

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.

Usage Guidelines See “Minimum MSDP Configuration” on page 321.

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

maximum

Syntax maximum *number*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp active-source-limit],
[edit protocols msdp active-source-limit],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp active-source-limit],
[edit routing-instances *instance-name* protocols msdp active-source-limit]

Release Information Statement introduced before JUNOS Release 7.4.

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

Usage Guidelines See "Configuring the MSDP Active Source Limit" on page 331.

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

See Also threshold on page 350

mode

Syntax mode (mesh-group | standard);

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* protocols msdp group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

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—(Optional) Group of peers that are mesh group members.

standard—Use standard MSDP source-active flooding rules.
Default: standard

Usage Guidelines See "Configuring MSDP Mesh Groups" on page 324.

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

msdp

Syntax

```
msdp {
    active-source-limit {
        maximum number;
        threshold number;
    }
    data-encapsulation (enable | disable);
    disable;
    rib-group group-name;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
        flag flag <flag-modifier> <disable>;
    }
    peer address {
        active-source-limit {
            maximum number;
            threshold number;
        }
        authentication-key peer-key;
        default-peer;
        disable;
        export [ policy-names ];
        import [ policy-names ];
        local-address address;
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <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);
        traceoptions {
            file name <replace> <size size> <files number> <no-stamp>
                <world-readable | no-world-readable>;
            flag flag <flag-modifier> <disable>;
        }
        peer address {
            active-source-limit {
                maximum number;
                threshold number;
            }
            authentication-key peer-key;
```

```
            default-peer;
            disable;
            export [ policy-names ];
            import [ policy-names ];
            local-address address;
            traceoptions {
                file name <replace> <size size> <files number> <no-stamp>
                    <world-readable | no-world-readable>;
                flag flag <flag-modifier> <disable>;
            }
        }
        source prefix/mask {
            active-source-limit {
                maximum number;
                threshold number;
            }
        }
    }
}
```

Hierarchy Level [edit protocols],
[edit logical-routers *logical-router-name* protocols],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols],
[edit routing-instances *instance-name* protocols]

Release Information Statement introduced before JUNOS Release 7.4.

Description Enable MSDP on the router. You must also configure at least one peer for MSDP to function.

Default MSDP is disabled on the router.

Options The statements are explained separately.

Usage Guidelines See "Enabling MSDP" on page 322.

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

peer

Syntax

```
peer address {
    active-source-limit {
        maximum number;
        threshold number;
    }
    authentication-key peer-key;
    default-peer;
    disable;
    export [ policy-names ];
    import [ policy-names ];
    local-address address;
    traceoptions {
        file name <replace> <size size> <files number> <no-stamp>
            <world-readable | no-world-readable>;
```

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

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit protocols msdp],
[edit protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*]

Release Information Statement introduced before JUNOS Release 7.4.

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 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.

Usage Guidelines See "Configuring MSDP Peers" on page 322.

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

rib-group

Syntax rib-group *group-name*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp]

Release Information Statement introduced before JUNOS Release 7.4.

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.

Usage Guidelines See "Enabling MSDP" on page 322.

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

source

Syntax

```
source prefix/mask {
    active-source-limit {
        maximum number;
        threshold number;
    }
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp]

Release Information Statement introduced before JUNOS Release 7.4.

Description Limit the number of active source messages the router accepts from sources in this address range.

Default If you do not include this statement, the router accepts any number of MSDP active source messages.

Options The other statements are explained separately.

Usage Guidelines See "Configuring Per-Source Active Source Limit" on page 332.

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

threshold

Syntax threshold *number*;

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp active-source-limit],
[edit protocols msdp active-source-limit],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp active-source-limit],
[edit routing-instances *instance-name* protocols msdp active-source-limit]

Release Information Statement introduced before JUNOS Release 7.4.

Description Configure the random early discard (RED) threshold for MSDP active source messages. This number should 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

Usage Guidelines See "Configuring the MSDP Active Source Limit" on page 331.

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

See Also maximum on page 346

traceoptions

Syntax

```
traceoptions {
    file name <replace> <size size> <files number> <no-stamp>
      <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
}
```

Hierarchy Level [edit logical-routers *logical-router-name* protocols msdp],
[edit logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit logical-routers *logical-router-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 *instance-name* logical-routers *logical-router-name* protocols msdp],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp group *group-name* peer *address*],
[edit routing-instances *instance-name* logical-routers *logical-router-name* protocols msdp peer *address*],
[edit routing-instances *instance-name* protocols msdp],
[edit routing-instances *instance-name* protocols msdp group *group-name*],
[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 before JUNOS Release 7.4.

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 *name*—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 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.

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 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.

Usage Guidelines See “Tracing MSDP Protocol Traffic” on page 335.

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.

Part 12

Indexes

Index

N

P

Q

V

Index of Statements and Commands

V