JUNIPER NETWORKS

Junos® OS

Multicast over Layer 3 VPNs Feature Guide

Release

11.1

Published: 2011-02-07

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

Junos® OS Multicast over Layer 3 VPNs Feature Guide
Release 11.1

Revision History
January 2011—R1 Junos OS 11.1

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 OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

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 (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) 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, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the 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 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 or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations 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 Embedded Software on non-Juniper equipment; (j) use Embedded 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 payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

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)).

Table of Contents

List of Figures

PART 1

Multicast over Layer 3 VPNs

CHAPTER 1

Multicast over Layer 3 VPNs Concept and Reference Materials

This chapter covers these topics:

Multicast over Layer 3 VPNs Overview

The Junos OS provides three ways to configure IP version 4 (IPv4) multicast over Layer 3 virtual private networks (VPNs):

- Multiprotocol BGP-based multicast VPNs: next-generation, defined by a set of sender sites and a set of receiver sites and use BGP as the signaling protocol. We recommend using this method to configure multicast on Layer 3 VPNs because it is a simpler implementation than draft-rosen multicast VPNs.
- Draft-rosen multicast VPNs with service provider tunnels operating in any-source multicast (ASM) mode—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on Section Two of the IETF Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs*. This information is provided to you in case you already have dual PIM multicast VPNs configured on your network.
- Draft-rosen multicast VPNs with service provider tunnels operating in source-specific multicast (SSM) mode—Described in RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)* and based on the IETF Internet draft draft-rosen-vpn-mcast-07.txt, *Multicast in MPLS/BGP IP VPNs*. For more information about these types of draft-rosen multicast VPNs, see the *Junos Multicast Protocols Configuration Guide*.

The reader should be familiar with Layer 3 VPN operation on Juniper Networks routers, as well as standard PIM configurations. For more information on Protocol Independent

Multicast (PIM) and Multicast Source Discovery Protocol (MSDP) and their usage in a Layer 3 VPN, see the *Junos Multicast Protocols Configuration Guide*. For more information on Layer 3 VPN configuration, see the *Junos VPNs Configuration Guide*. Both manuals are located at **http://www.juniper.net/techpubs/software/index.html**.

Multiprotocol BGP-Based Multicast VPNs: Next-Generation

Multiprotocol BGP-based multicast VPNs (also referred to as next-generation Layer 3 VPN multicast) constitute the next evolution after dual multicast VPNs (draft-rosen) and provide a simpler solution for administrators who want to configure multicast over Layer 3 VPNs.

The main characteristics of multiprotocol BGP-based multicast VPNs are:

- They extend Layer 3 VPN service (RFC 2547) to support IP multicast for Layer 3 VPN service providers
- They follow the same architecture as specified by RFC 2547 for unicast VPNs. Specifically, BGP is used as the control plane.
- They eliminate the requirement for the virtual router (VR) model, which is specified in Internet draft draft-rosen-vpn-mcast, *Multicast in MPLS/BGP VPNs*, for multicast VPNs.
- They rely on RFC-based unicast with extensions for intra-AS and inter-AS communication.

Multiprotocol BGP-based VPNs are defined by two sets of sites: a sender set and a receiver set. Hosts within a receiver site set can receive multicast traffic and hosts within a sender site set can send multicast traffic. A site set can be both receiver and sender, which means that hosts within such a site can both send and receive multicast traffic. Multiprotocol BGP-based VPNS can span organizations (so the sites can be intranets or extranets), can span service providers, and can overlap.

Site administrators configure multiprotocol BGP-based VPNs based on customer requirements and the existing BGP and MPLS VPN infrastructure. For more detailed information about multiprotocol BGP-based VPN configuration statements, see the *Junos OS VPNs Configuration Guide*.

Dual PIM Multicast VPNs: Draft Rosen

Junos OS supports Layer 3 VPNs based on the Internet draft draft-rosen-rfc2547bis, *BGP/MPLS VPNs*. This Internet draft defines a mechanism by which service providers can use their IP backbones to provide Layer 3 VPN services to their customers. The sites that make up a Layer 3 VPN are connected over a provider's existing public Internet backbone.

VPNs based on draft-rosen-rfc2547bis are also known as BGP/MPLS VPNs because BGP is used to distribute VPN routing information across the provider's backbone, and MPLS is used to forward VPN traffic across the backbone to remote VPN sites.

NOTE: Draft-rosen multicast VPNs are not supported in a logical system environment even though the configuration statements can be configured under the logical-systems hierarchy.

Customer networks, because they are private, can use either public addresses or private addresses, as defined in RFC 1918, *Address Allocation for Private Internets*. When customer networks that use private addresses connect to the public Internet infrastructure, the private addresses might overlap with the private addresses used by other network users. BGP/MPLS VPNs solve this problem by prefixing a VPN identifier to each address from a particular VPN site, thereby creating an address that is unique both within the VPN and within the public Internet. In addition, each VPN has its own VPN-specific routing table that contains the routing information for that VPN only.

In a unicast environment for Layer 3 VPNs, all VPN states are contained within the provider edge (PE) routers. With multicast over Layer 3 VPNs, two PIM adjacencies are established: one between the customer edge (CE) and PE routers through a VPN routing and forwarding (VRF) routing instance, the second between the main PE routers and their service provider core neighbors.

The set of master PIM adjacencies throughout the service provider's network makes up the forwarding path, and eventually forms a rendezvous point (RP) multicast distribution tree. The tree is rooted at the RP contained within the service provider's network. Because of this, core provider transit routers within the service provider's network must maintain multicast state information for the VPNs.

For multicast in Layer 3 VPNs to work correctly, there must be two types of rendezvous points. The VPN customer rendezvous point (VPN C-RP) is an RP that resides within a VPN that connects the segments of a customer network. The service provider rendezvous point (SP-RP) resides within the service provider network itself. Because a PE router connects to both the customer network and the service provider network, a PE router can act as an SP-RP, a VPN C-RP, or both.

NOTE: If you configure auto-RP or bootstrap router (BSR) on a PE router, the PE router cannot act as a VPN C-RP in a routing instance, but can learn about another router acting as the VPN C-RP.

System Requirements for Multiprotocol BGP-Based Multicast VPNs: Next-Generation

To implement multiprotocol BGP-based multicast VPNs, your system must meet these minimum requirements:

- Junos OS Release 9.2 or later for PIM dense mode, bootstrap router (BSR), auto-RP, and configuration of a PE router as the VPN C-RP. (Configuration of a PE router as the VPN C-RP is not a requirement in Junos OS Release 10.0 and later if you configure RPT-SPT mode. See the *Junos VPNs Configuration Guide*.)
- Junos OS Release 8.5 or later for point-to-multipoint traffic engineering provider tunnels (next-generation multicast VPN only).
- Junos OS Release 8.4 or later.
- Any hardware needed in your network to enable your Juniper Networks routers to act as PE routers.
- On M Series and T Series routers, a Tunnel Services PIC for any provider core router acting as an SP-RP.
- On M Series and T Series routers, a Tunnel Services PIC for any PE router where GRE tunneling is needed.
- On M Series and T Series routers, a Tunnel Services PIC for any CE or PE router acting as a DR or VPN C-RP.
- On M Series and T Series routers, a Tunnel Services PIC is required for GRE tunneling, as specified in Section Two of the IETF Internet draft *Multicast in MPLS/BGP VPNs*.

System Requirements for Dual PIM Multicast VPNs: Draft Rosen

- Junos OS Release 8.2 or later for support on MX Series routing platforms.
- Junos OS Release 7.2 or later for MSDP in a Layer 3 VPN.
- Junos OS Release 7.1 or later for multicast distribution trees for data.
- Junos OS Release 6.4 or later for PIM sparse mode graceful restart and configuring a PE router as the VPN C-RP.
- Junos OS Release 5.5 or later for PIM dense mode and logical loopback interfaces.
- Junos OS Release 5.3 or later for PIM sparse mode.
- Any hardware needed in your network to enable your Juniper Networks routers to act as PE routers.
- On M Series and T Series routers, a Tunnel Services PIC for any provider core router acting as an SP-RP.
- On M Series and T Series routers, a Tunnel Services PIC for any PE router where GRE tunneling is needed.
- On M Series and T Series routers, a Tunnel Services PIC for any CE or PE router acting as a DR or VPN C-RP.

- On M Series and T Series routers, a Tunnel Services PIC for GRE tunneling, as specified in Section Two of the IETF Internet draft *Multicast in MPLS/BGP VPNs*.
- For point-to-multipoint traffic engineering, a Tunnel Services PIC or vrf-tabel-label configuration.

Multicast over Layer 3 VPNs Terms and Acronyms

I

inclusive tree — In multiprotocol BGP multicast VPNs, a single multicast distribution tree in the backbone that carries all the multicast traffic from a specified set of one or more multicast VPNs. An inclusive tree that carries the traffic of more than one multicast VPNs is an *aggregate inclusive tree*. An inclusive tree contains, as its members, all the PEs that attach to receiver sites of any of the multicast VPNs using the tree.

M

master PIM instance — The global instance of PIM that is configured at the **[edit protocols pim]** hierarchy level.

multicast domain — The set of VPN routing and forwarding (VRF) instances associated with interfaces that can send multicast traffic to one another.

S

selective tree — In multiprotocol BGP multicast VPNs, a single multicast distribution tree in the backbone that carries traffic that belongs to a specified set of one or more multicast groups from one or more multicast VPNs. Such a tree is referred to as an *aggregate selective tree* when the multicast groups belong to different multicast VPNs.

SP-RP — The rendezvous point (RP) for the service provider (this RP is not contained within the VPN).

V

VPN C-RP — The customer RP for the VPN (this RP is contained within the VPN).

CHAPTER 2

Configuring MBGP MVPNs

Configuring multiprotocol BGP-based (MBGP) multicast VPNs (MVPNs) (also referred to as next-generation Layer 3 VPN multicast) involves a series of major steps. Each major step can have variations; for example, you can choose to configure different types of provider tunnels. Think of the major steps as building blocks that can be used together in various ways.

To implement multiprotocol BGP-based multicast VPNs, you must perform one or more of the following major steps:

Creating a Unique Logical Loopback Interface for the Routing Instance for MBGP MVPNs

To facilitate the PIM protocol within a Layer 3 VPN, configure a unique loopback interface for the routing instance at the **[edit interfaces lo0 unit]** hierarchy level:

```
[edit interfaces]
lo0 {
   unit 1 {
      family inet {
         address ip-address;
      }
   }
}
```

Configuring Interfaces for Layer 3 VPNs

Configure the Layer 3 VPN logical interfaces and specify the family as **inet**:

```
[edit interfaces]
interface-name {
   unit logical-unit-number {
      family inet;
   }
}
```

Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for MBGP MVPNs

To send multicast traffic across a Layer 3 VPN, you must configure network protocols to handle *intradomain routing* (an interior gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 3 VPN, see the *Junos VPNs Configuration Guide*.

NOTE: 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 does not work because PIM cannot transmit upstream interface information for multicast sources behind remote PE routers 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.

Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN

By default a multiprotocol BGP-based multicast VPN routing instance attaches to both sender and receiver sites. You can also manually configure the instance to attach to only sender or only receiver sites.

To create a multicast VPN routing instance, include the **mvpn** statement at the [**edit routing-instances *routing-instance name* protocols**] hierarchy level:

```
[edit]
routing-instances {
```

```
    vpn-a {
      instance-type vrf;
      protocols {
        mvpn { # Enables BGP/MPLS multicast VPN configuration.
        }
      }
    }
}
```

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

Option: Configuring Sender and Receiver Sites

By default, multiprotocol BGP-based VPNs are attached to both sender and receiver sites. To specify that a VPN be attached only to a sender site or only to a receiver site, include the **receiver-site** or **sender-site** statement at the [**edit routing instances** *routing-instance-name* **protocols mvpn**] hierarchy level:

```
[edit]
routing-instances {
  vpn-a {
    instance-type vrf;
    protocols {
      mvpn {
        receiver-site;
        sender-site;
      }
    }
  }
}
```

Option: Specifying Route Targets

Specifying route targets for sender and receiver sites is useful when there is a mix of sender only, receiver only, and sender and receiver sites. This is because a sender site's routing table is used for exporting routes from a sender site and importing routes from a receiver site. A receiver site's routing table is used for exporting routes from a receiver site and importing routes from a sender site. A sender and receiver site does both. The route targets configured under multicast VPNs apply only to multicast VPN AD routes of type 1, 2, 3, and 5.

A PE router with sites in a specific multicast VPN must determine whether a received automatic discovery route is from a sender site or receiver site based on the following:

- If the PE router is configured to be only in a sender site, route targets are imported only from receiver sites. Imported automatic discovery routes must be from a receiver site.

- If the PE router is configured to be only in a receiver site, route targets are imported only from sender sites. Imported automatic discovery routes must be from a sender site.
- If the PE router is configured to be both in sender and receiver sites, the following guidelines apply:
  - Along with an import route target, you can optionally configure whether the route target is from a receiver or a sender site.
  - If a configuration is not provided, an imported automatic discovery route is treated as belonging to both the sender site and the receiver site.

For more information on route targets, see the *Junos OS VPNs Configuration Guide*.

In the following example, the export target is used for multicast VPN autodiscovery routes. It overrides the default VRF export target if the **unicast** statement is not included. The unicast statement applies the VRF export target and multicast VPN export route target to multicast VPN autodiscovery routes. The **apply-groups** statement specifies the groups from which to inherit configuration data. The a**pply-groups-except** statement specifies the groups from which to not inherit configuration data. The **import-target** statement specifies the import target for multicast VPN autodiscovery routes. It overrides the default VRF import target if the **unicast** statement is not included.

To specify route targets, include the **route-target** statement at the [**edit routing-instances** *routing-instance-name* **protocols mvpn**] hierarchy level:

```
[edit]
routing-instances {
  vpn-a {
    protocols {
      mvpn {
        route-target {
          export-target {
            target target-community;
            unicast; #
            apply-groups group-name;
            apply-groups-except group-name;
          }
          import-target { #
            target target-value { # Target community.
              receiver target-value; # Target community used when importing receiver #
                site routes.
              sender target-value; # Target community used when importing sender # site
                routers.
            }
            unicast {
              receiver;
              sender;
            }
            apply-groups group-name;
            apply-groups-except group-name;
          }
        }
      }
```

```
    }
  }
}
```

Existing VRF import or VRF export policies for importing and exporting VPN routes might prevent import or export of multicast VPN routes if the policies reflect routes based on the protocol type. The workaround is to change the policy to not reflect routes based on the protocol type or to use additional multicast VPN-specific configuration.

If the VRF import policy does not import BGP routes, multicast VPN routes of type 1, 2, 3, or 5 imported by BGP are rejected. There are two workarounds:

- You can add a term to allow routes from the BGP protocol with **family inet-mvpn**:

  ```
  term 2 {
    from {
      protocol bgp;
      family inet-mvpn;
      community vpn_blue;
    }
    then {
      accept;
    }
  }
  ```

  The customer can configure an import target under **mvpn**. The multicast VPN route of type 1, 2, 3, or 5 matching the configured target are imported.

  ```
  protocol mvpn {
    import-target {
      target target:2:2;
    }
  }
  ```

- If the VRF export policy uses a policy qualifier of type **protocol** to reject routes, the multicast VPN routes of type 1, 2, 3, or 5 are not exported. This is because Junos OS does not support a policy qualifier for MVPN. The workaround is to configure an export target under MVPN without including the **unicast** statement:

  ```
  protocol mvpn {
    export-target {
      target target:2:2;
      }
    }
  }
  ```

Configuring Provider Tunnels

The source address for a PIM-SM provider tunnel is the loopback address of the loopback interface in **inet.0**.

To configure a provider tunnel, include the **provider-tunnel** statement at the [**edit routing-instances** *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
```

```
    vpn-a {
      provider-tunnel {
        pim-asm {
          apply-groups group-name;
          apply-groups-except group-name;
          group-address address;
        }
      }
    }
}
```

Enabling Multicast VPN in BGP

You also must enable multicast VPN by including the **inet-mvpn** or **inet6-mvpn** statements at the [**edit protocols bgp family**] hierarchy level:

```
[edit]
protocols {
  bgp {
    family {
      inet-mvpn; # Enables IPv4 multicast VPN.
      inet6-mvpn; # Enables IPv6 multicast VPN.
    }
  }
}
```

Configuring Intra-AS Inclusive Point-to-Multipoint Traffic Engineering LSPs

Point-to-multipoint traffic engineering LSPs are supported as the data plane for intra-AS inclusive provider tunnels. A multicast VPN can be configured to use inclusive trees or selective trees or a combination of both. Aggregation is not supported for point-to-multipoint traffic engineering LSPs.

NOTE: Configure either LDP or regular MPLS LSPs between PE routers to ensure VPN unicast connectivity. Point-to-multipoint LSPs are used for multicast data forwarding only.

You must configure the following when configuring point-to-multipoint LSPs in provider tunnels:

- The BGP multicast VPN control plane, as described in “Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN” on page 10.
- Point-to-multipoint traffic engineering as the provider tunnel technology on each PE configured for multicast VPN that belongs to the sender site.
- Either a VT interface or a vrf-table-label on the multicast VPN instance. For more information about configuring VT interfaces, see the *Junos OS VPNs Configuration Guide*.
- Point-to-multipoint traffic engineering support on each P router.

On each PE router, a point-to-multipoint traffic engineering LSP must be configured for every multicast VPN instance that belongs to a sender site set. This means that if there are four multicast VPN instances configured on a PE router and three of these multicast VPN instances belong to sender site sets, three point-to-multipoint traffic engineering LSPs must be configured on this PE router. The PE would be the root of the three point-to-multipoint traffic engineering LSPs, and the leaves of the LSPs would be determined either dynamically or through a static configuration.

If the multicast VPN instance is configured for dynamic leaf discovery, the leaves are automatically discovered through intra-AS autodiscovery routes. The point-to-multipoint LSPs can be signaled using default attributes or configured attributes. If you configure the multicast VPN instance to use default attributes, the LSPs cannot be signaled with bandwidth reservation and do not support CAC. Point-to-multipoint LSPs with configured attributes support both bandwidth reservation and CAC. In addition, they can be configured to support traffic engineering attributes such as fast-reroute.

If the multicast VPN instance is configured for static leaf discovery, you configure the leafs statically. Point-to-multipoint LSPs that are configured statically support all traffic engineering attributes.

To configure dynamic leaf discovery, include the **label-switched-path-template** statement at the [**edit routing-instance** *routing-instance-name* **provider-tunnel rsvp-te**] hierarchy level. Dynamic discovery can be configured by using default attributes with the **default-template** statement at the [**edit routing-instance** *routing-instance-name* **provider-tunnel rsvp-te label-switched-path-template**] hierarchy level.

If you want to signal with bandwidth reservation, use CAC, or use other traffic engineering options such as link protection, configure a template for dynamic leaf discovery by including the **label-switched-path-template** *template-name* statement at the [**edit protocols mpls**] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path mvpn-example-p2mp-temlate {
         template;
         p2mp;
         link-protection;
         optimize-timer 50;
         traceoptions {
            file mvpn-a-p2mp-lsp.log;
            flag all;
         }
      }
   }
}
```

You can apply the configured or default template by including the template name at the [**edit routing-instance** *routing-instance-name* **provider-tunnel rsvp-te label-switched-path-template**] hierarchy level. Be sure to either configure a VT interface or include the **vrf-table-label** statement in the routing instance.

```
[edit]
```

```
routing-instance {
   routing-instance configured-dynamic-example {
   instance-type vrf;
   interface ge-1/0/0.0;
   route-distinguisher 10.255.71.1:100;
   vrf-table-label;
      provider-tunnel {
         rsvp-te label switched-path-template mvpn-example-p2mp-temlate;
      }
   }
}
```

To configure static LSPs, include the **label-switched-path** *label-switched-path* statement at the [**edit protocols mpls**] hierarchy level:

```
[edit]
protocols {
   mpls {
      label-switched-path vpls-example-p2mp-s21_lsp_a {
         to 192.168.1.1
         p2mp example-static-lsp;
      }
      label-switched-path vpls-example-p2mp-s21_lsp_b {
         to 192.168.1.2;
         p2mp example-static-lsp;
      }
   }
}
```

To apply statically configured LSPs, include the **static** statement at the [**edit routing-instance** *routing-instance-name* **provider-tunnel rsvp-te static-lsp** *static-lsp-name*] hierarchy level:

```
[edit]
routing-instance example-static {
   provider-tunnel {
      rsvp-te {
         static-lsp example-static-lsp;
      }
   }
}
```

Configuring Intra-AS Selective Provider Tunnels

Point-to-multipoint traffic engineering LSPs are supported as the data plane for selective provider tunnels. A multicast VPN can be configured to use inclusive trees or selective trees or a combination of both. Aggregation is not supported for point-to-multipoint traffic engineering LSPs.

NOTE: Configure either LDP or regular MPLS LSPs between PE routers to ensure VPN unicast connectivity. Point-to-multipoint LSPs are used for multicast data forwarding only.

You must configure the following when configuring point-to-multipoint LSPs in provider tunnels:

- The BGP multicast VPN control plane, as described in "Creating a Routing Instance for a Multiprotocol BGP-Based Multicast VPN" on page 10.
- Point-to-multipoint traffic engineering as the provider tunnel technology on each PE configured for multicast VPN that belongs to the sender site.
- Either a VT interface or a vrf-table-label on the multicast VPN instance. For more information about configuring VT interfaces, see the *Junos OS VPNs Configuration Guide*.
- Point-to-multipoint traffic engineering support on each P router.

When selective trees are used, there must be a separate point-to-multipoint traffic engineering LSP for each multicast distribution tree in the backbone that carries traffic belonging to a specified set of one or more multicast groups, from one or more multicast VPNs. Multiple groups can be bound to the same selective point-to-multipoint LSP if the selective point-to-multipoint LSP leaves are statically configured. If the leaves are dynamically discovered, only one source or group can be bound to it.

Selective point-to-multipoint LSPs can be statically configured or triggered by a bandwidth threshold. If the threshold rate is configured, a S-PMSI autodiscovery route is generated for a particular (C-S, C-G) if it falls in the range specified by (C-S prefix, C-G prefix) and its data rate exceeds the configured threshold rate. In the following example, the threshold rate is set to zero Kbps. This causes the selective tunnel to be created immediately, meaning that the multicast traffic does not use an inclusive tunnel at all. Optionally, you can leave the threshold rate unconfigured and the result is the same as setting the threshold to zero.

Below is an example configuration for point-to-multipoint LSPs on a selective tunnel with statically configured leafs:

```
[edit]
routing-instances {
   selective-tunnel-example {
      instance-type vpls;
      route-distinguisher 10.255.71.2:100;
      protocols {
         vpls {
            tunnel-services { # This enables vt interfaces for this routing instance.
         }
      }
      provider-tunnel {
         rsvp-te {
            label-switched-path-template {
               mvpn_template;
            }
         }
         }
         selective {
            group 225.10.10.1/32 {
               source 192.2.1.2/32 {
```

```
            threshold-rate 0;
            rsvp-te {
              static-lsp lsp1;
            }
          }
        }
        group 226.10.10.1/32 {
          source 192.2.1.2/32 {
            rsvp-te {
              static-lsp lsp1;
            }
          }
        }
      }
    }
  }
}
```

The following example shows an example with dynamic selective trees and the default template:

```
[edit]
routing-instances {
  dynamic-selective-tunnel-example {
    instance-type vpls;
    route-distinguisher 10.255.71.2:100;
    protocols {
      vpls {
        tunnel-services {
      }
    }
    provider-tunnel {
      rsvp-te {
        label-switched-path-template {
          default-template;
        }
      }
      selective {
        group 225.10.10.1/32 {
          source 192.2.1.2/32 {
            rsvp-te {
              label-switched-path-template default-template;
            }
          }
        }
        group 226.10.10.1/32 {
          source 192.2.1.2/32 {
            rsvp-te {
              label-switched-path-template default-template;
            }
          }
        }
      }
    }
  }
}
```

Configuring the Master PIM Instance on the PE Router for BGP-Based Multicast VPNs

To configure the master PIM instance that communicates with other PIM neighbors, include the **pim** statement at the **[edit protocols]** hierarchy level. BGP-based multicast VPNs support sparse mode, dense mode, or sparse-dense mode. The first example shown enables PIM sparse mode.

```
[edit protocols]
pim {
   interface all {
      mode sparse;
      version 2;
   }
}
```

The next example shown enables PIM dense mode.

```
[edit protocols]
pim {
   interface all {
      mode dense;
   }
}
```

Configuring the Router's IPv4 Bootstrap Router Priority

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.

```
[edit protocols]
pim {
   bootstrap-priority number;
}
```

Configuring MBGP MVPNs to Support IPv6 Multicast Traffic

Multiprotocol BGP-based (MBGP) multicast VPNs (MVPNs) (also referred to as next-generation Layer 3 VPN multicast) can transport IPv6 multicast customer traffic across an IPv4 core network using RSVP-TE tunnels. Transporting IPv6 multicast customer traffic across an IPv4 core network does not require that IPv6 support or multicast support be configured in the core network.

The following features are supported for IPv6 multicast transport:

- PIM sparse mode
  - Static rendezvous points
  - Bootstrap router protocol
  - Embedded RP
- PIM dense mode

- PIM source-specific multicast
- MLD on the provider edge (PE) router to host interface
- RSVP-TE, PIM-SSM, or PIM-ASM inclusive provider tunnels
- RSVP-TE selective provider tunnels
- BGP mesh topologies or route reflectors

The configuration required to support IPv6 multicast traffic across an MBGP MVPN is largely the same as the configuration to support IPv4 multicast traffic. For an example of configuring an MBGP MVPN to transport IPv4 multicast customer traffic, see "Example: Configuring MBGP Multicast VPNs" on page 29.

To transport IPv6 multicast customer traffic across a service provider IPv4 core network:

1. If RSVP-TE LSPs and the bootstrap router protocol are used, enable the IPv6 address family on all core-facing interfaces on all PE routers participating in the MVPN. It is not necessary to configure an IPv6 address.

```
[edit protocols bgp family inet6-mvpn]
interface so-0/1/0 {
  unit 0 {
    family inet6 {
    }
  }
}
```

2. Enable MBGP to carry multicast VPN NLRI for the IPv6 address family and enable VPN signaling on all PE routers participating in the MVPN.

```
[edit protocols]
bgp {
  group vsix_mcast {
    family inet6-mvpn {
      signaling;
    }
  }
}
```

3. Enable MBGP to carry Layer 3 VPN NLRI for the IPv6 address family for unicast routes on all PE routers participating in the MVPN.

```
[edit protocols]
bgp {
  group vsix_mcast {
    family inet6-vpn {
      unicast;
    }
  }
}
```

4. Allow IPv6 routes to be resolved over an MPLS network by converting all routes stored in the **inet.3** routing table to IPv4-compatible IPv6 addresses and then copying them into the **inet6.3** routing table on all PE routers participating in the MVPN.

```
[edit protocols]
```



```
mpls {
  ipv6-tunneling;
  interface so-0/1/0;
  interface fe-0/0/0;
}
```

The **inet6.3** routing table is used to resolve next-hop addresses for both IPv6 and IPv6 VPN routes.

5. By default, the routers support MLD version 1 (MLDv1). If you want to use MLDv2 on directly connected customer hosts, configure MLD version 2 on the PE router to host interfaces.

```
[edit protocols]
mld {
  version 2;
}
```

6. Configure either a static rendezvous point (RP) or the bootstrap router (BSR) protocol using the following:

   - If you are using a static RP, configure the local RP IPv6 address on the router that is the rendezvous point.

```
[edit protocols pim rp]
local {
  family inet6 {
    address ::10.12.53.12;
  }
}
```

   - If you are using a static RP, configure the RP IPv6 address on all PE routers in the MVPN that are not the rendezvous point.

```
[edit protocols pim rp]
static {
  address ::10.12.53.12;
}
```

   - If you are using the BSR protocol for automatic RP discovery, configure a bootstrap router to support the IPv6 address family.

```
[edit protocols pim rp]
bootstrap {
  family inet6;
}
```

7. (Optional) If you are using a route reflector and the route reflector does not have a full mesh of RSVP LSPs tunnels, it might be necessary to add a static route. The static route is used to ensure that no updates are dropped by BGP dbecause no next-hop attribute are present. Configure the static route in the **inet6.3** routing table with the **0::0/0** prefix. Set the next-hop attribute to **discard** and give the route a metric of **65000**. The high metric used in this example allows the router to use the other routes with a lower metric if they exist.

```
[edit routing-options]
rib inet6.3 {
```

```
    static {
      route 0::0/0 {
        discard;
        metric 65000;
      }
    }
  }
```

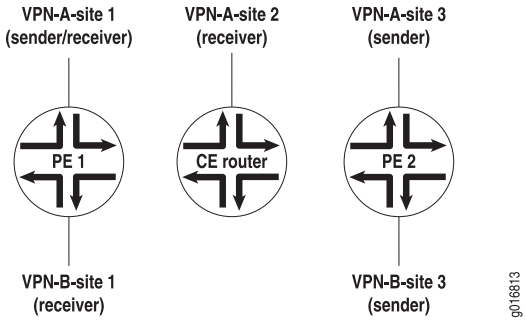
8. After you have configured the network to transport IPv6 multicast customer traffic across an IPv4 core network and allowed time for the routes to be discovered, use the **show route table *instance_name*.mvpn-inet6.0** command to see the type 1 autodiscovery routes.
9. Use the **show mvpn instance** command to verify that tthe provider tunnels have been established.
10. Use the **show route table bgp.mvpn-inet6.0** command to verify that the correct BGP routes have been learned from MVPN.
11. After multicast traffic is flowing, use the **show multicast route instance *instance_name* extensive inet6** command to verify that the multicast state, group address, source address, upstream interface list, and downstream interface list are correct.
12. Use the **show route table *instance_name*.inet6.1** command to verify that the multicast forwarding table is correct.

Example: Configuring MBGP MVPN Senders and Receivers using PIM SM and RSVP-TE Provider Tunnels

This section contains a configuration example and commands you can issue to verify multiprotocol BGP multicast VPN sender and receiver site routers.

In the example shown in Figure 1 on page 22, there are three routers: PE1, the CE router, and PE2. Each router supports a specific role as a sender, receiver, or sender and receiver. This example does not represent a complete network.

Figure 1: Multiprotocol BGP Multicast VPN Senders and Receivers



Router PE1 is configured as a multicast sender and receiver site in VPN A, and as a multicast receiver site in VPN B. The relevant configuration for Router PE1 follows.

Router PE1

```
[edit]
routing-instances {
  vpn-a {
```

```
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    provider tunnel {
      rsvp-te {
        label-switched-path-template default-template;
      }
    }
    protocols {
      mvpn {
        route-target {
          export-target unicast target:1:4;
          import-target unicast sender target target:1:4 receiver;
        }
      }
    }
    route-distinguisher 65535:0;
    vrf-target target:1:1;
    routing-options {
      auto-export;
      static {
        route 172.16.0.0/16 next-hop so-0/0/0.0;
        route 172.17.0.0/16 next-hop so-6/0/1.0;
      }
    }
  }
}

[edit]
routing-instance {
  vpn-b {
    instance-type vrf;
    interface ge-0/3/0.0;
    provider-tunnel {
      pim-sm {
        group-address 224.1.1.2;
      }
    }
    protocols {
      mvpn {
        receiver-site;
        router-target {
          export target:1:5;
          import unicast;
        }
      }
    }
    route-distinguisher 65535:1;
    vrf-target target:1:2;
    routing-options {
      auto-export;
    }
  }
}
```

Router PE2 is configured as a multicast sender-only site. The relevant configuration for Router PE2 follows.

Router PE2

```
[edit]
routing-instances {
   vpn-a {
      instance-type vrf;
      interface so-1/0/0.0;
      provider-tunnel {
         rsvp-te {
            label-switched-path-template default-template;
         }
      }
      protocols {
         mvpn {
            sender-site;
            route-target {
               export target:1:4;
               import unicast;
            }
         }
         route-distinguisher 65535:2;
         vrf-target target:1:1;
         routing-options {
            auto-export;
            static {
               route 172.16.0.0/16 next-hop so-0/0/0.1;
               route 172.17.0.0/16 next-hope so-6/0/1.0;
            }
         }
      }
   }
}

[edit]
routing-instance {
   vpn-b {
      instance-type vrf;
      interface ge-0/3/0.0;
      provider-tunnel {
         pim-sm {
            group-address 224.1.1.2;
         }
      }
      protocols {
         mvpn {
            sender-site;
            router-target {
               export target:1:5;
               import unicast;
            }
         }
      }
      route-distinguisher 65535:1;
      vrf-target target:1:2;
      routing-options {
```

```
        auto-export;
      }
    }
  }
```

Verifying Your Work

To verify correct operation of multiprotocol BGP multicast VPNs, use the following commands:

- **show mvpn c-multicast**
- **show mvpn instance**
- **show mvpn neighbor**

The following sections show the output of these commands used with the configuration example:

- show mvpn c-multicast on page 25
- show mvpn instance on page 26
- show mvpn neighbor on page 28

show mvpn c-multicast

```
Router> show mvpn c-multicast
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM


Router> show mvpn c-multicast extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel
```

```
Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM


Router> show mvpn c-multicast summary
Instance: VPN-A
  C-multicast IPv4 route count: 1
Instance: VPN-B
  C-multicast IPv4 route count: 2
```

show mvpn instance

```
Router> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.2.0.0
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM


Router> show mvpn instance extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                          St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM
MVPN instance:
```

```
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.2.0.0
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM


Router> show mvpn instance
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:RSVP-TE P2MP:10.255.71.190, 27859,10.255.71.190
  Neighbor                              I-P-tnl
  10.255.71.2                           RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2

  C-mcast IPv4 (S:G)            Ptnl                              St
  192.1.1.2/32:225.10.10.1/32   RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2
    DS
  0.0.0.0/0:225.10.10.1/32

Router> show mvpn instance summary
Instance: VPN-A
  Neighbor count: 2
  C-multicast IPv4 route count: 1
Instance: VPN-B
  Neighbor count: 4
  C-multicast IPv4 route count: 2


Router> show mvpn instance VPN-A
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                              St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1         RM


Router> show mvpn instance VPN-A extensive
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Provider tunnel: I-P-tnl:PIM-SM:10.255.14.144, 239.1.1.1
```

```
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1        RM

Router> show mvpn instance VPN-A summary
Instance: VPN-A
  Neighbor count: 2
  C-multicast IPv4 route count: 1
```

show mvpn neighbor

```
Router> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.1.1.1
  10.255.70.17                          PIM-SM:10.255.70.17, 239.1.1.1
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  Neighbor                              I-P-tnl
  10.255.14.160                         PIM-SM:10.255.14.160, 239.2.0.0
  10.255.70.17                          PIM-SM:10.255.70.17, 239.2.0.0


Router> show mvpn neighbor extensive
Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.78/32:225.5.5.5/32 PIM-SM:10.255.14.144, 239.1.1.1        RM
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-B
  C-mcast IPv4 (S:G)            Ptnl                            St
  192.168.195.94/32:226.6.6.6/32 PIM-SM:10.255.14.144, 239.2.0.0         RM
```

The following is output for a p2mp configuration.

```
Router> show mvpn neighbor
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: VPN-A
  Neighbor                              I-P-tnl
  10.255.71.2                         RSVP-TE P2MP:10.255.71.2, 39143,10.255.71.2
```

Example: Configuring MBGP Multicast VPNs

This example provides a step-by-step procedure to configure multicast services across a multiprotocol BGP (MBGP) Layer 3 virtual private network.

Before You Begin

Depending on the devices you are using, you might be required to configure static routes to:

- The multicast sender
- The Fast Ethernet interface to which the sender is connected on the multicast receiver
- The multicast receiver
- The Fast Ethernet interface to which the receiver is connected on the multicast sender

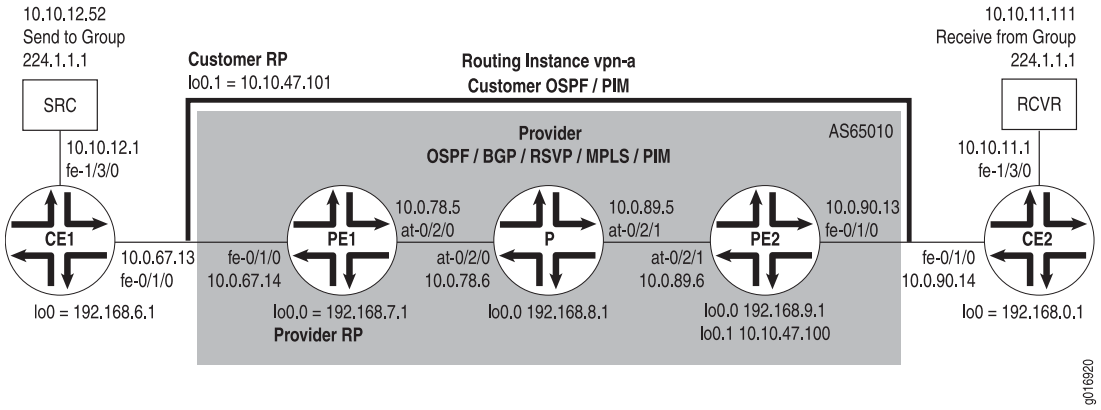
Overview and Topology

This example shows how to configure the following technologies:

- IPv4
- BGP
- OSPF
- RSVP
- MPLS
- PIM sparse mode
- Static RP

The topology of the network is shown in Figure 2 on page 30.

Figure 2: Multicast Over Layer 3 VPN Example Topology



Configuration

NOTE: In any configuration session, it is a good practice to periodically verify that the configuration can be committed using the commit check command.

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **P** identifies the provider core (P) router
- **CE2** identifies the customer edge 2 (CE2) router
- **PE2** identifies the provider edge 2 (PE2) router

To configure MBGP multicast VPNs for the network shown in Figure 2 on page 30, perform the following steps:

Configuring Interfaces

Step-by-Step Procedure

1. On each router, configure an IP address on the loopback logical interface 0 (**lo0.0**).

```
user@CE1# set interfaces lo0 unit 0 family inet address 192.168.6.1/32 primary

user@PE1# set interfaces lo0 unit 0 family inet address 192.168.7.1/32 primary

user@P# set interfaces lo0 unit 0 family inet address 192.168.8.1/32 primary

user@PE2# set interfaces lo0 unit 0 family inet address 192.168.9.1/32 primary

user@CE2# set interfaces lo0 unit 0 family inet address 192.168.0.1/32 primary
```

   Use the **show interfaces terse** command to verify that the IP address is correct on the loopback logical interface.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet interfaces. Specify the **inet** protocol family type.

```
user@CE1# set interfaces fe-1/3/0 unit 0 family inet address 10.10.12.1/24
user@CE1# set interfaces fe-0/1/0 unit 0 family inet address 10.0.67.13/30

user@PE1# set interfaces fe-0/1/0 unit 0 family inet address 10.0.67.14/30

user@PE2# set interfaces fe-0/1/0 unit 0 family inet address 10.0.90.13/30

user@CE2# set interfaces fe-0/1/0 unit 0 family inet address 10.0.90.14/30
user@CE2# set interfaces fe-1/3/0 unit 0 family inet address 10.10.11.1/24
```

   Use the **show interfaces terse** command to verify that the IP address is correct on the Fast Ethernet interfaces.

3. On the PE and P routers, configure the ATM interfaces' VPI and maximum virtual circuits. If the default PIC type is different on directly connected ATM interfaces, configure the PIC type to be the same. Configure the logical interface VCI, protocol family, local IP address, and destination IP address.

```
user@PE1# set interfaces at-0/2/0 atm-options pic-type atm1
user@PE1# set interfaces at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@PE1# set interfaces at-0/2/0 unit 0 vci 0.128
user@PE1# set interfaces at-0/2/0 unit 0 family inet address 10.0.78.5/32
  destination 10.0.78.6

user@P# set interfaces at-0/2/0 atm-options pic-type atm1
user@P# set interfaces at-0/2/0 atm-options vpi 0 maximum-vcs 256
user@P# set interfaces at-0/2/0 unit 0 vci 0.128
user@P# set interfaces at-0/2/0 unit 0 family inet address 10.0.78.6/32 destination
  10.0.78.5
user@P# set interfaces at-0/2/1 atm-options pic-type atm1
user@P# set interfaces at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@P# set interfaces at-0/2/1 unit 0 vci 0.128
user@P# set interfaces at-0/2/1 unit 0 family inet address 10.0.89.5/32 destination
  10.0.89.6
```

```
user@PE2# set interfaces at-0/2/1 atm-options pic-type atm1
user@PE2# set interfaces at-0/2/1 atm-options vpi 0 maximum-vcs 256
user@PE2# set interfaces at-0/2/1 unit 0 vci 0.128
user@PE2# set interfaces at-0/2/1 unit 0 family inet address 10.0.89.6/32
  destination 10.0.89.5
```

Use the **show configuration interfaces** command to verify that the ATM interfaces' VPI and maximum VCs are correct and that the logical interface VCI, protocol family, local IP address, and destination IP address are correct.

Configuring OSPF

Step-by-Step Procedure

1. On the P and PE routers, configure the provider instance of OSPF. Specify the **lo0.0** and ATM core-facing logical interfaces. The provider instance of OSPF on the PE router forms adjacencies with the OSPF neighbors on the other PE router and Router P.

```
user@PE1# set protocols ospf area 0.0.0.0 interface at-0/2/0.0
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.0

user@P# set protocols ospf area 0.0.0.0 interface lo0.0
user@P# set protocols ospf area 0.0.0.0 interface all
user@P# set protocols ospf area 0.0.0.0 interface fxp0 disable

user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0
user@PE2# set protocols ospf area 0.0.0.0 interface at-0/2/1.0
```

   Use the **show ospf interfaces** command to verify that the **lo0.0** and ATM core-facing logical interfaces are configured for OSPF.

2. On the CE routers, configure the customer instance of OSPF. Specify the loopback and Fast Ethernet logical interfaces. The customer instance of OSPF on the CE routers form adjacencies with the neighbors within the VPN routing instance of OSPF on the PE routers.

```
user@CE1# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@CE1# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@CE1# set protocols ospf area 0.0.0.0 interface lo0.0

user@CE2# set protocols ospf area 0.0.0.0 interface fe-0/1/0.0
user@CE2# set protocols ospf area 0.0.0.0 interface fe-1/3/0.0
user@CE2# set protocols ospf area 0.0.0.0 interface lo0.0
```

   Use the **show ospf interfaces** command to verify that the correct loopback and Fast Ethernet logical interfaces have been added to the OSPF protocol.

3. On the P and PE routers, configure OSPF traffic engineering support for the provider instance of OSPF.

   The **shortcuts** statement enables the master instance of OSPF to use a label-switched path as the next hop.

```
user@PE1# set protocols ospf traffic-engineering shortcuts

user@P# set protocols ospf traffic-engineering shortcuts
```

```
user@PE2# set protocols ospf traffic-engineering shortcuts
```

Use the **show ospf overview** or **show configuration protocols ospf** command to verify that traffic engineering support is enabled.

Configuring BGP

Step-by-Step Procedure

1. On Router P, configure BGP for the VPN. The local address is the local **lo0.0** address. The neighbor addresses are the PE routers' **lo0.0** addresses.

   The **unicast** statement enables the router to use BGP to advertise network layer reachability information (NLRI). The **signaling** statement enables the router to use BGP as the signaling protocol for the VPN.

   ```
   user@P# set protocols bgp group group-mvpn type internal
   user@P# set protocols bgp group group-mvpn local-address 192.168.8.1
   user@P# set protocols bgp group group-mvpn family inet unicast
   user@P# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@P# set protocols bgp group group-mvpn neighbor 192.168.9.1
   user@P# set protocols bgp group group-mvpn neighbor 192.168.7.1
   ```

   Use the **show configuration protocols bgp** command to verify that the router has been configured to use BGP to advertise NLRI.

2. On the PE and P routers, configure the BGP local autonomous system number.

   ```
   user@PE1# set routing-options autonomous-system 0.65010

   user@P# set routing-options autonomous-system 0.65010

   user@PE2# set routing-options autonomous-system 0.65010
   ```

   Use the **show configuration routing-options** command to verify that the BGP local autonomous system number is correct.

3. On the PE routers, configure BGP for the VPN. Configure the local address as the local **lo0.0** address. The neighbor addresses are the **lo0.0** addresses of Router P and the other PE router, PE2.

   ```
   user@PE1# set protocols bgp group group-mvpn type internal
   user@PE1# set protocols bgp group group-mvpn local-address 192.168.7.1
   user@PE1# set protocols bgp group group-mvpn family inet-vpn unicast
   user@PE1# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@PE1# set protocols bgp group group-mvpn neighbor 192.168.9.1
   user@PE1# set protocols bgp group group-mvpn neighbor 192.168.8.1

   user@PE2# set protocols bgp group group-mvpn type internal
   user@PE2# set protocols bgp group group-mvpn local-address 192.168.9.1
   user@PE2# set protocols bgp group group-mvpn family inet-vpn unicast
   user@PE2# set protocols bgp group group-mvpn family inet-mvpn signaling
   user@PE2# set protocols bgp group group-mvpn neighbor 192.168.7.1
   user@PE2# set protocols bgp group group-mvpn neighbor 192.168.8.1
   ```

   Use the **show bgp group** command to verify that the BGP configuration is correct.

4. On the PE routers, configure a policy to export the BGP routes into OSPF.

```
user@PE1# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@PE1# set policy-options policy-statement bgp-to-ospf then accept

user@PE2# set policy-options policy-statement bgp-to-ospf from protocol bgp
user@PE2# set policy-options policy-statement bgp-to-ospf then accept
```

Use the **show policy bgp-to-ospf** command to verify that the policy is correct.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, enable RSVP on the interfaces that participate in the LSP. Configure the Fast Ethernet and ATM logical interfaces.

   ```
   user@PE1# set protocols rsvp interface fe-0/1/0.0
   user@PE1# set protocols rsvp interface at-0/2/0.0

   user@PE2# set protocols rsvp interface fe-0/1/0.0
   user@PE2# set protocols rsvp interface at-0/2/1.0
   ```

2. On Router P, enable RSVP on the interfaces that participate in the LSP. Configure the ATM logical interfaces.

   ```
   user@P# set protocols rsvp interface at-0/2/0.0
   user@P# set protocols rsvp interface at-0/2/1.0
   ```

   Use the **show configuration protocols rsvp** command to verify that the RSVP configuration is correct.

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure an MPLS LSP to the PE router that is the LSP egress point. Specify the IP address of the **lo0.0** interface on the router at the other end of the LSP. Configure MPLS on the ATM, Fast Ethernet, and **lo0.0** interfaces.

   To help identify each LSP when troubleshooting, configure a different LSP name on each PE router. In this example, we use the name **to-pe2** as the name for the LSP configured on PE1 and **to-pe1** as the name for the LSP configured on PE2.

   ```
   user@PE1# set protocols mpls label-switched-path to-pe2 to 192.168.9.1
   user@PE1# set protocols mpls interface fe-0/1/0.0
   user@PE1# set protocols mpls interface at-0/2/0.0
   user@PE1# set protocols mpls interface lo0.0

   user@PE2# set protocols mpls label-switched-path to-pe1 to 192.168.7.1
   user@PE2# set protocols mpls interface fe-0/1/0.0
   user@PE2# set protocols mpls interface at-0/2/1.0
   user@PE2# set protocols mpls interface lo0.0
   ```

   Use the **show configuration protocols mpls** and **show route label-switched-path to-pe1** commands to verify that the MPLS and LSP configuration is correct.

   After the configuration is committed, use the **show mpls lsp name to-pe1** and **show mpls lsp name to-pe2** commands to verify that the LSP is operational.

2. On Router P, enable MPLS. Specify the ATM interfaces connected to the PE routers.

   ```
   user@P# set protocols mpls interface at-0/2/0.0
   ```

```
user@P# set protocols mpls interface at-0/2/1.0
```

Use the **show mpls interface** command to verify that MPLS is enabled on the ATM interfaces.

3. On the PE and P routers, configure the protocol family on the ATM interfaces associated with the LSP. Specify the **mpls** protocol family type.

```
user@PE1# set interfaces at-0/2/0 unit 0 family mpls

user@P# set interfaces at-0/2/0 unit 0 family mpls
user@P# set interfaces at-0/2/1 unit 0 family mpls

user@PE2# set interfaces at-0/2/1 unit 0 family mpls
```

Use the **show mpls interface** command to verify that the MPLS protocol family is enabled on the ATM interfaces associated with the LSP.

Configuring the VRF Routing Instance

Step-by-Step Procedure

1. On the PE routers, configure a routing instance for the VPN and specify the **vrf** instance type. Add the Fast Ethernet and **lo0.1** customer-facing interfaces. Configure the VPN instance of OSPF and include the BGP-to-OSPF export policy.

```
user@PE1# set routing-instances vpn-a instance-type vrf
user@PE1# set routing-instances vpn-a interface lo0.1
user@PE1# set routing-instances vpn-a interface fe-0/1/0.0
user@PE1# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@PE1# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface all

user@PE2# set routing-instances vpn-a instance-type vrf
user@PE2# set routing-instances vpn-a interface lo0.1
user@PE2# set routing-instances vpn-a interface fe-0/1/0.0
user@PE2# set routing-instances vpn-a protocols ospf export bgp-to-ospf
user@PE2# set routing-instances vpn-a protocols ospf area 0.0.0.0 interface all
```

Use the **show configuration routing-instances vpn-a** command to verify that the routing instance configuration is correct.

2. On the PE routers, configure a route distinguisher for the routing instance. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes. Configure a different route distinguisher on each PE router. This example uses 65010:1 on PE1 and 65010:2 on PE2.

```
user@PE1# set routing-instances vpn-a route-distinguisher 65010:1

user@PE2# set routing-instances vpn-a route-distinguisher 65010:2
```

Use the **show configuration routing-instances vpn-a** command to verify that the route distinguisher is correct.

3. On the PE routers, configure default VRF import and export policies. Based on this configuration, BGP automatically generates local routes corresponding to the route target referenced in the VRF import policies. This example uses 2:1 as the route target.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@PE1# set routing-instances vpn-a vrf-target target:2:1

user@PE2# set routing-instances vpn-a vrf-target target:2:1
```

Use the **show configuration routing-instances vpn-a** command to verify that the route target is correct.

4. On the PE routers, configure the VPN routing instance for multicast support.

```
user@PE1# set routing-instances vpn-a protocols mvpn

user@PE2# set routing-instances vpn-a protocols mvpn
```

Use the **show configuration routing-instance vpn-a** command to verify that the VPN routing instance has been configured for multicast support.

5. On the PE routers, configure an IP address on loopback logical interface 1 (**lo0.1**) used in the customer routing instance VPN.

```
user@PE1# set interfaces lo0 unit 1 family inet address 10.10.47.101/32

user@PE2# set interfaces lo0 unit 1 family inet address 10.10.47.100/32
```

Use the **show interfaces terse** command to verify that the IP address on the loopback interface is correct.

Configuring PIM

Step-by-Step Procedure

1. On the PE and P routers, enable the provider instance of PIM. Add the core-facing ATM interfaces. On the PE routers, also configure the **lo0.0** interface. Specify the mode as **sparse** and the version as **2**.

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

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

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

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

2. On the PE routers, enable the VPN customer instance of PIM. Configure the **lo0.1** and the customer-facing Fast Ethernet interface. Specify the mode as **sparse** and the version as **2**.

```
user@PE1# set routing-instances vpn-a protocols pim interface lo0.1 mode sparse
user@PE1# set routing-instances vpn-a protocols pim interface lo0.1 version 2
user@PE1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 mode
  sparse
user@PE1# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 version
  2
```

```
user@PE2# set routing-instances vpn-a protocols pim interface lo0.1 mode sparse
user@PE2# set routing-instances vpn-a protocols pim interface lo0.1 version 2
user@PE2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 mode
  sparse
user@PE2# set routing-instances vpn-a protocols pim interface fe-0/1/0.0 version
  2
```

   Use the **show pim interfaces instance vpn-a** command to verify that PIM sparse-mode is enabled on the **lo0.1** interface and the customer-facing Fast Ethernet interface.

3. On the CE routers, enable the customer instance of PIM. In this example, we configure all interfaces. Specify the mode as **sparse** and the version as **2**.

```
user@CE1# set protocols pim interface all
```

```
user@CE2# set protocols pim interface all mode sparse
user@CE2# set protocols pim interface all version 2
```

   Use the **show pim interfaces** command to verify that PIM sparse mode is enabled on all interfaces.

Configuring the Provider Tunnel

Step-by-Step Procedure

1. On Router PE1, configure the provider tunnel. Specify the multicast address to be used.

   The **provider-tunnel** statement instructs the router to send multicast traffic across a tunnel. The **pim-asm** statement instructs the router to accept the multicast stream from any source.

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

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

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

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

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

Configuring the Rendezvous Point

Step-by-Step Procedure

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

   user@PE1# set protocols pim rp local address 192.168.7.1

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

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

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

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

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

3. Configure Router PE1 to be the rendezvous point for the customer instance of PIM. Specify the **lo0.1** address of Router PE1. Specify the multicast address to be used.

   user@PE1# set routing-instances vpn-a protocols pim rp local address 10.10.47.101
   user@PE1# set routing-instances vpn-a protocols pim rp local group-ranges 224.1.1.1/32

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

4. On Router PE2, configure the static rendezvous point for the customer instance of PIM. Specify the **lo0.1** address of Router PE1.

   user@PE2# set routing-instances vpn-a protocols pim rp static address 10.10.47.101

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

5. On the CE routers, configure the static rendezvous point for the customer instance of PIM. Specify the **lo0.1** address of Router PE1.

   user@CE1# set protocols pim rp static address 10.10.47.101 version 2

   user@CE2# set protocols pim rp static address 10.10.47.101 version 2

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

6. Use the **commit check** command to verify that the configuration can be successfully committed. If the configuration passes the check, commit the configuration.

7. Start the multicast sender device connected to CE1.

8. Start the multicast receiver device connected to CE2.

9. Verify that the receiver is receiving the multicast stream.

10. Use **show** commands to verify the routing, VPN, and multicast operation.

Results The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router CE1 follows.

Router CE1

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.6.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.67.13/30;
            }
        }
    }
    fe-1/3/0 {
        unit 0 {
            family inet {
                address 10.10.12.1/24;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface lo0.0;
            interface fe-1/3/0.0;
        }
    }
    pim {
        rp {
            static {
                address 10.10.47.101 {
                    version 2;
                }
            }
        }
        interface all;
    }
}
```

The relevant sample configuration for Router PE1 follows.

Router PE1

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
```

```
            address 192.168.7.1/32 {
              primary;
            }
          }
        }
      }
      fe-0/1/0 {
        unit 0 {
          family inet {
            address 10.0.67.14/30;
          }
        }
      }
      at-0/2/0 {
        atm-options {
          pic-type atm1;
          vpi 0 {
            maximum-vcs 256;
          }
        }
        unit 0 {
          vci 0.128;
          family inet {
            address 10.0.78.5/32 {
              destination 10.0.78.6;
            }
          }
          family mpls;
        }
      }
      lo0 {
        unit 1 {
          family inet {
            address 10.10.47.101/32;
          }
        }
      }
    }
    routing-options {
      autonomous-system 0.65010;
    }
    protocols {
      rsvp {
        interface fe-0/1/0.0;
        interface at-0/2/0.0;
      }
      mpls {
        label-switched-path to-pe2 {
          to 192.168.9.1;
        }
        interface fe-0/1/0.0;
        interface at-0/2/0.0;
        interface lo0.0;
      }
      bgp {
        group group-mvpn {
```

```
        type internal;
        local-address 192.168.7.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.9.1;
        neighbor 192.168.8.1;
      }
    }
    ospf {
      traffic-engineering {
        shortcuts;
      }
      area 0.0.0.0 {
        interface at-0/2/0.0;
        interface lo0.0;
      }
    }
    pim {
      rp {
        local {
          address 192.168.7.1;
        }
      }
      interface at-0/2/0.0 {
        mode sparse;
        version 2;
      }
      interface lo0.0 {
        mode sparse;
        version 2;
      }
    }
}
policy-options {
  policy-statement bgp-to-ospf {
    from protocol bgp;
    then accept;
  }
}
routing-instances {
  vpn-a {
    instance-type vrf;
    interface lo0.1;
    interface fe-0/1/0.0;
    route-distinguisher 65010:1;
    provider-tunnel {
      pim-asm {
        group-address 224.1.1.1;
      }
    }
    vrf-target target:2:1;
    protocols {
```

```
        ospf {
          export bgp-to-ospf;
          area 0.0.0.0 {
            interface all;
          }
        }
        pim {
          rp {
            local {
              address 10.10.47.101;
              group-ranges {
                224.1.1.1/32;
              }
            }
          }
          interface lo0.1 {
            mode sparse;
            version 2;
          }
          interface fe-0/1/0.0 {
            mode sparse;
            version 2;
          }
        }
        mvpn;
      }
    }
  }
```

The relevant sample configuration for Router P follows.

Router P

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.8.1/32 {
          primary;
        }
      }
    }
  }
  at-0/2/0 {
    atm-options {
      pic-type atm1;
      vpi 0 {
        maximum-vcs 256;
      }
    }
    unit 0 {
      vci 0.128;
      family inet {
        address 10.0.78.6/32 {
          destination 10.0.78.5;
        }
      }
      family mpls;
```

```
        }
    }
    at-0/2/1 {
        atm-options {
            pic-type atm1;
            vpi 0 {
                maximum-vcs 256;
            }
        }
        unit 0 {
            vci 0.128;
            family inet {
                address 10.0.89.5/32 {
                    destination 10.0.89.6;
                }
            }
            family mpls;
        }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
        interface at-0/2/0.0;
        interface at-0/2/1.0;
    }
    mpls {
        interface at-0/2/0.0;
        interface at-0/2/1.0;
    }
    bgp {
        group group-mvpn {
            type internal;
            local-address 192.168.8.1;
            family inet {
                unicast;
            }
            family inet-mvpn {
                signaling;
            }
            neighbor 192.168.9.1;
            neighbor 192.168.7.1;
        }
    }
    ospf {
        traffic-engineering {
            shortcuts;
        }
        area 0.0.0.0 {
            interface lo0.0;
            interface all;
            interface fxp0.0 {
                disable;
            }
```

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

The relevant sample configuration for Router PE2 follows.

Router PE2

```
interfaces {
  lo0 {
    unit 0 {
      family inet {
        address 192.168.9.1/32 {
          primary;
        }
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
      family inet {
        address 10.0.90.13/30;
      }
    }
  }
  at-0/2/1 {
    atm-options {
      pic-type atm1;
      vpi 0 {
        maximum-vcs 256;
      }
    }
    unit 0 {
      vci 0.128;
      family inet {
        address 10.0.89.6/32 {
          destination 10.0.89.5;
        }
      }
      family mpls;
    }
```

```
    }
    lo0 {
      unit 1 {
        family inet {
          address 10.10.47.100/32;
        }
      }
    }
}
routing-options {
    autonomous-system 0.65010;
}
protocols {
    rsvp {
      interface fe-0/1/0.0;
      interface at-0/2/1.0;
    }
    mpls {
      label-switched-path to-pe1 {
        to 192.168.7.1;
      }
      interface lo0.0;
      interface fe-0/1/0.0;
      interface at-0/2/1.0;
    }
    bgp {
      group group-mvpn {
        type internal;
        local-address 192.168.9.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.7.1;
        neighbor 192.168.8.1;
      }
    }
    ospf {
      traffic-engineering {
        shortcuts;
      }
      area 0.0.0.0 {
        interface lo0.0;
        interface at-0/2/1.0;
      }
    }
    pim {
      rp {
        static {
          address 192.168.7.1 {
            version 2;
          }
        }
      }
```

```
      interface lo0.0 {
        mode sparse;
        version 2;
      }
      interface at-0/2/1.0 {
        mode sparse;
        version 2;
      }
    }
  }
  policy-options {
    policy-statement bgp-to-ospf {
      from protocol bgp;
      then accept;
    }
  }
  routing-instances {
    vpn-a {
      instance-type vrf;
      interface fe-0/1/0.0;
      interface lo0.1;
      route-distinguisher 65010:2;
      provider-tunnel {
        pim-asm {
          group-address 224.1.1.1;
        }
      }
      vrf-target target:2:1;
      protocols {
        ospf {
          export bgp-to-ospf;
          area 0.0.0.0 {
            interface all;
          }
        }
        pim {
          rp {
            static {
              address 10.10.47.101;
            }
          }
          interface fe-0/1/0.0 {
            mode sparse;
            version 2;
          }
          interface lo0.1 {
            mode sparse;
            version 2;
          }
        }
        mvpn;
      }
    }
  }
```

The relevant sample configuration for Router CE2 follows.

Router CE2

```
interfaces {
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.1/32 {
                    primary;
                }
            }
        }
    }
    fe-0/1/0 {
        unit 0 {
            family inet {
                address 10.0.90.14/30;
            }
        }
    }
    fe-1/3/0 {
        unit 0 {
            family inet {
                address 10.10.11.1/24;
            }
            family inet6 {
                address fe80::205:85ff:fe88:ccdb/64;
            }
        }
    }
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface fe-0/1/0.0;
            interface lo0.0;
            interface fe-1/3/0.0;
        }
    }
    pim {
        rp {
            static {
                address 10.10.47.101 {
                    version 2;
                }
            }
        }
        interface all {
            mode sparse;
            version 2;
        }
    }
}
```


CHAPTER 3

Configuring Multicast VPN Extranets

This chapter covers these topics:

MBGP Multicast VPN Extranets Overview

A multicast VPN (MVPN) extranet enables service providers to forward IP multicast traffic originating in one VPN routing and forwarding (VRF) instance to receivers in a different VRF instance. This capability is also know as *overlapping* MVPNs.

The MVPN extranet feature supports the following traffic flows:

- A receiver in one VRF can receive multicast traffic from a source connected to a different router in a different VRF.
- A receiver in one VRF can receive multicast traffic from a source connected to the same router in a different VRF.
- A receiver in one VRF can receive multicast traffic from a source connected to a different router in the same VRF.
- A receiver in one VRF can be prevented from receiving multicast traffic from a specific source in a different VRF.

MBGP Multicast VPN Extranets Application

An MVPN extranet is useful in the following applications.

Mergers and Data Sharing

An MVPN extranet is useful when there are business partnerships between different enterprise VPN customers that require them to be able to communicate with one another. For example, a wholesale company might want to broadcast inventory to its contractors and resellers. An MVPN extranet is also useful when companies merge and one set of VPN sites needs to receive content from another VPN. The enterprises involved in the

merger are different VPN customers from the service provider point of view. The MVPN extranet makes the connectivity possible.

Video Distribution

Another use for MVPN extranets is video multicast distribution from a video headend to receiving sites. Sites within a given multicast VPN might be in different organizations. The receivers can subscribe to content from a specific content provider.

The PE routers on the MVPN provider network learn about the sources and receivers using MVPN mechanisms. These PE routers can use selective trees as the multicast distribution mechanism in the backbone. The network carries traffic belonging only to a specified set of one or more multicast groups, from one or more multicast VPNs. As a result, this model facilitates the distribution of content from multiple providers on a selective basis if desired.

Financial Services

A third use for MVPN extranets is enterprise and financial services infrastructures. The delivery of financial data, such as financial market updates, stock ticker values, and financial TV channels, is an example of an application that must deliver the same data stream to hundreds and potentially thousands of end users. The content distribution mechanisms largely rely on multicast within the financial provider network. In this case, there could also be an extensive multicast topology within brokerage firms and banks networks to enable further distribution of content and for trading applications. Financial service providers require traffic separation between customers accessing the content, and MVPN extranets provide this separation.

Related Documentation

- Example: Configuring MBGP Multicast VPN Extranets on page 51
- MBGP Multicast VPN Extranets Configuration Guidelines on page 50

MBGP Multicast VPN Extranets Configuration Guidelines

When configuring MVPN extranets, keep the following in mind:

- If there is more than one VRF routing instance on a provider edge (PE) router that has receivers interested in receiving multicast traffic from the same source, virtual tunnel (VT) interfaces must be configured on all instances.
- For auto-RP operation, the mapping agent must be configured on at least two PEs in the extranet network.
- For asymmetrically configured extranets using auto-RP, when one VRF instance is the only instance that imports routes from all other extranet instances, the mapping agent must be configured in the VRF that can receive all RP discovery messages from all VRF instances, and mapping-agent election should be disabled.
- For bootstrap router (BSR) operation, the candidate and elected BSRs can be on PE, CE, or C routers. The PE router that connects the BSR to the MVPN extranets must have configured provider tunnels or other physical interfaces configured in the routing instance. The only case not supported is when the BSR is on a CE or C router connected

to a PE routing instance that is part of an extranet but does not have configured provider tunnels and does not have any other interfaces besides the one connecting to the CE router.

- RSVP-TE point-to-multipoint LSPs must be used for the provider tunnels.
- PIM dense mode is not supported in the MVPN extranets VRF instances.

Related Documentation

- Example: Configuring MBGP Multicast VPN Extranets on page 51
- MBGP Multicast VPN Extranets Overview on page 49

Example: Configuring MBGP Multicast VPN Extranets

This example provides a step-by-step procedure to configure multicast VPN extranets using static rendezvous points. It is organized in the following sections:

Requirements

This example uses the following hardware and software components:

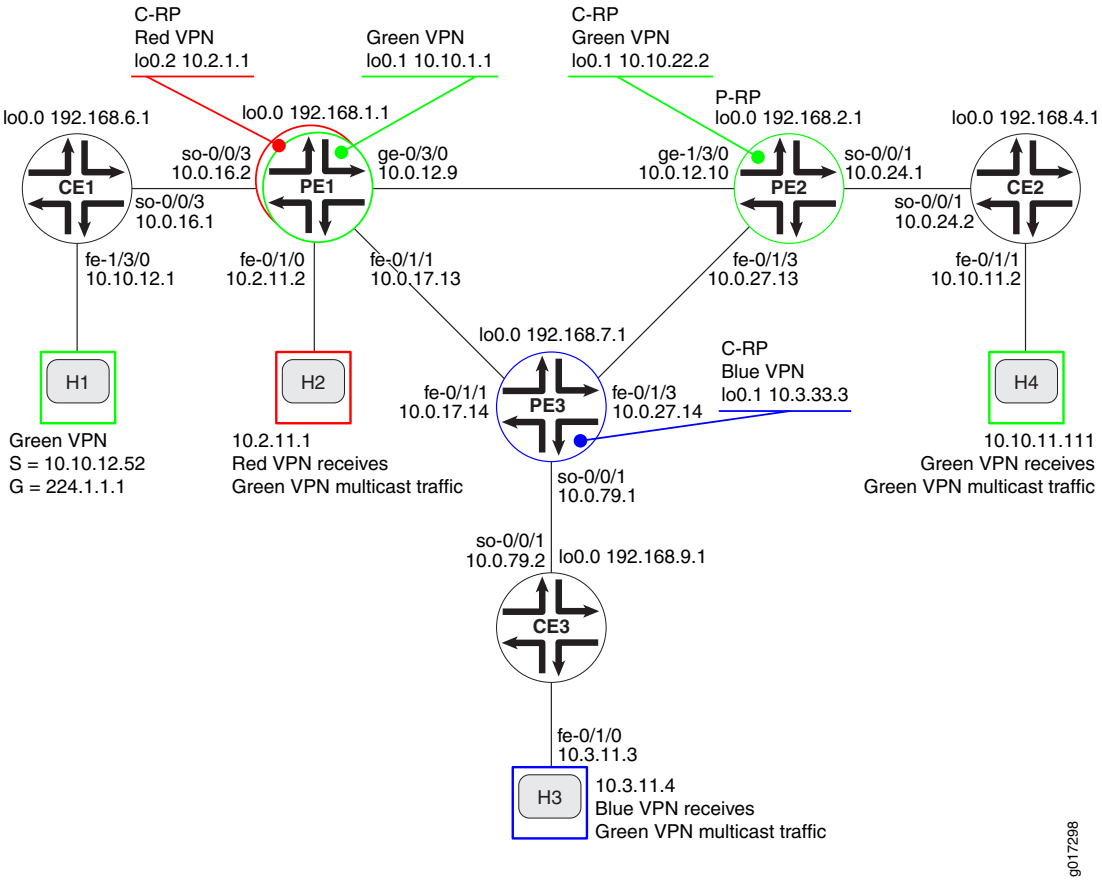
- Junos OS Release 9.5 or later
- Six M Series, T Series, TX Series, or MX Series Juniper routers
- One adaptive services PIC or MultiServices PIC in each of the M Series or T Series routers acting as PE routers
- One host system capable of sending multicast traffic and supporting the Internet Group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

Overview and Topology

In the network topology shown in Figure 3 on page 52:

- Host H1 is the source for group 244.1.1.1 in the green VPN.
- The multicast traffic originating at source H1 can be received by host H4 connected to router CE2 in the green VPN.
- The multicast traffic originating at source H1 can be received by host H3 connected to router CE3 in the blue VPN.
- The multicast traffic originating at source H1 can be received by host H2 directly connected to router PE1 in the red VPN.
- Any host can be a sender site or receiver site.

Figure 3: MVPN Extranets Topology Diagram



Configuration

NOTE: In any configuration session, it is good practice to verify periodically that the configuration can be committed using the commit check command.

In this example, the router being configured is identified using the following command prompts:

- **CE1** identifies the customer edge 1 (CE1) router
- **PE1** identifies the provider edge 1 (PE1) router
- **CE2** identifies the customer edge 2 (CE2) router
- **PE2** identifies the provider edge 2 (PE2) router
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router

Configuring multicast VPN extranets, involves the following tasks:

Configuring Interfaces

Step-by-Step Procedure

1. On each router, configure an IP address on the loopback logical interface 0 (**lo0.0**).

   ```
   user@CE1# set interfaces lo0 unit 0 family inet address 192.168.6.1/32 primary

   user@PE1# set interfaces lo0 unit 0 family inet address 192.168.1.1/32 primary
   user@PE2# set interfaces lo0 unit 0 family inet address 192.168.2.1/32 primary

   user@CE2# set interfaces lo0 unit 0 family inet address 192.168.4.1/32 primary

   user@PE3# set interfaces lo0 unit 0 family inet address 192.168.7.1/32 primary

   user@CE3# set interfaces lo0 unit 0 family inet address 192.168.9.1/32 primary
   ```

   Use the **show interfaces terse** command to verify that the correct IP address is configured on the loopback interface.

2. On the PE and CE routers, configure the IP address and protocol family on the Fast Ethernet and Gigabit Ethernet interfaces. Specify the **inet** address family type.

   ```
   user@CE1# set interfaces fe-1/3/0 unit 0 family inet address 10.10.12.1/24

   user@PE1# set interfaces fe-0/1/0 unit 0 description "to H2"
   user@PE1# set interfaces fe-0/1/0 unit 0 family inet address 10.2.11.2/30
   user@PE1# set interfaces fe-0/1/1 unit 0 description "to PE3 fe-0/1/1.0"
   user@PE1# set interfaces fe-0/1/1 unit 0 family inet address 10.0.17.13/30
   user@PE1# set interfaces ge-0/3/0 unit 0 family inet address 10.0.12.9/30

   user@PE2# set interfaces fe-0/1/3 unit 0 description "to PE3 fe-0/1/3.0"
   user@PE2# set interfaces fe-0/1/3 unit 0 family inet address 10.0.27.13/30
   user@PE2# set interfaces ge-1/3/0 unit 0 description "to PE1 ge-0/3/0.0"
   user@PE2# set interfaces ge-1/3/0 unit 0 family inet address 10.0.12.10/30
   ```

```
user@CE2# set interfaces fe-0/1/1 unit 0 description "to H4"
user@CE2# set interfaces fe-0/1/1 unit 0 family inet address 10.10.11.2/24

user@PE3# set interfaces fe-0/1/1 unit 0 description "to PE1 fe-0/1/1.0"
user@PE3# set interfaces fe-0/1/1 unit 0 family inet address 10.0.17.14/30
user@PE3# set interfaces fe-0/1/3 unit 0 description "to PE2 fe-0/1/3.0"
user@PE3# set interfaces fe-0/1/3 unit 0 family inet address 10.0.27.14/30

user@CE3# set interfaces fe-0/1/0 unit 0 description "to H3"
user@CE3# set interfaces fe-0/1/0 unit 0 family inet address 10.3.11.3/24
```

Use the **show interfaces terse** command to verify that the correct IP address and address family type are configured on the interfaces.

3. On the PE and CE routers, configure the SONET interfaces. Specify the **inet** address family type, and local IP address.

```
user@CE1# set interfaces so-0/0/3 unit 0 description "to PE1 so-0/0/3.0;"
user@CE1# set interfaces so-0/0/3 unit 0 family inet address 10.0.16.1/30

user@PE1# set interfaces so-0/0/3 unit 0 description "to CE1 so-0/0/3.0"
user@PE1# set interfaces so-0/0/3 unit 0 family inet address 10.0.16.2/30

user@PE2# set interfaces so-0/0/1 unit 0 description "to CE2 so-0/0/1:0.0"
user@PE2# set interfaces so-0/0/1 unit 0 family inet address 10.0.24.1/30

user@CE2# set interfaces so-0/0/1 unit 0 description "to PE2 so-0/0/1"
user@CE2# set interfaces so-0/0/1 unit 0 family inet address 10.0.24.2/30

user@PE3# set interfaces so-0/0/1 unit 0 description "to CE3 so-0/0/1.0"
user@PE3# set interfaces so-0/0/1 unit 0 family inet address 10.0.79.1/30

user@CE3# set interfaces so-0/0/1 unit 0 description "to PE3 so-0/0/1"
user@CE3# set interfaces so-0/0/1 unit 0 family inet address 10.0.79.2/30
```

Use the **show configuration interfaces** command to verify that the correct IP address and address family type are configured on the interfaces.

4. On each router, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

5. Use the **ping** command to verify unicast connectivity between each:

   - CE router and the attached host
   - CE router and the directly attached interface on the PE router
   - PE router and the directly attached interfaces on the other PE routers

Configuring an IGP in the Core

Step-by-Step Procedure

On the PE routers, configure an interior gateway protocol such as OSPF or IS-IS. This example shows how to configure OSPF.

1. Specify the **lo0.0** and SONET core-facing logical interfaces.

```
user@PE1# set protocols ospf area 0.0.0.0 interface ge-0/3/0.0 metric 100
user@PE1# set protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 100
user@PE1# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@PE1# set protocols ospf area 0.0.0.0 interface fxp0.0 disable

user@PE2# set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@PE2# set protocols ospf area 0.0.0.0 interface ge-1/3/0.0 metric 100
user@PE2# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@PE2# set protocols ospf area 0.0.0.0 interface fxp0.0 disable

user@PE3# set protocols ospf area 0.0.0.0 interface lo0.0 passive
user@PE3# set protocols ospf area 0.0.0.0 interface fe-0/1/3.0 metric 100
user@PE3# set protocols ospf area 0.0.0.0 interface fe-0/1/1.0 metric 100
user@PE3# set protocols ospf area 0.0.0.0 interface fxp0.0 disable
```

2. On the PE routers, configure a router ID.

```
user@PE1# set routing-options router-id 192.168.1.1

user@PE2# set routing-options router-id 192.168.2.1

user@PE3# set routing-options router-id 192.168.7.1
```

   Use the **show ospf overview** and **show configuration protocols ospf** commands to verify that the correct interfaces have been configured for the OSPF protocol.

3. On the PE routers, configure OSPF traffic engineering support. Enabling traffic engineering extensions supports the Constrained Shortest Path First algorithm, which is needed to support Resource Reservation Protocol - Traffic Engineering (RSVP-TE) point-to-multipoint label-switched paths (LSPs). If you are configuring IS-IS, traffic engineering is supported without any additional configuration.

```
user@PE1# set protocols ospf traffic-engineering

user@PE2# set protocols ospf traffic-engineering

user@PE3# set protocols ospf traffic-engineering
```

   Use the **show ospf overview** and **show configuration protocols ospf** commands to verify that traffic engineering support is enabled for the OSPF protocol.

4. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit
```

```
commit complete
```

5. On the PE routers, verify that the OSPF neighbors form adjacencies.

```
user@PE1> show ospf neighbors

Address          Interface              State     ID               Pri  Dead
10.0.17.14       fe-0/1/1.0             Full      192.168.7.1      128    32
10.0.12.10       ge-0/3/0.0             Full      192.168.2.1      128    33
```

Verify that the neighbor state with the other two PE routers is **Full**.

Configuring BGP in the Core

Step-by-Step Procedure

1. On the PE routers, configure BGP. Configure the BGP local autonomous system number.

```
user@PE1# set routing-options autonomous-system 65000

user@PE2# set routing-options autonomous-system 65000

user@PE3# set routing-options autonomous-system 65000
```

2. Configure the BGP peer groups. Configure the local address as the **lo0.0** address on the router. The neighbor addresses are the **lo0.0** addresses of the other PE routers.

   The **unicast** statement enables the router to use BGP to advertise network layer reachability information (NLRI). The **signaling** statement enables the router to use BGP as the signaling protocol for the VPN.

```
user@PE1# set protocols bgp group group-mvpn type internal
user@PE1# set protocols bgp group group-mvpn local-address 192.168.1.1
user@PE1# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE1# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE1# set protocols bgp group group-mvpn neighbor 192.168.2.1
user@PE1# set protocols bgp group group-mvpn neighbor 192.168.7.1

user@PE2# set protocols bgp group group-mvpn type internal
user@PE2# set protocols bgp group group-mvpn local-address 192.168.2.1
user@PE2# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE2# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE2# set protocols bgp group group-mvpn neighbor 192.168.1.1
user@PE2# set protocols bgp group group-mvpn neighbor 192.168.7.1

user@PE3# set protocols bgp group group-mvpn type internal
user@PE3# set protocols bgp group group-mvpn local-address 192.168.7.1
user@PE3# set protocols bgp group group-mvpn family inet-vpn unicast
user@PE3# set protocols bgp group group-mvpn family inet-mvpn signaling
user@PE3# set protocols bgp group group-mvpn neighbor 192.168.1.1
user@PE3# set protocols bgp group group-mvpn neighbor 192.168.2.1
```

3. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds
```

```
user@host> commit

commit complete
```

4. On the PE routers, verify that the BGP neighbors form a peer session.

```
user@PE1> show bgp group

Group Type: Internal    AS: 65000                 Local AS: 65000
  Name: group-mvpn      Index: 0                  Flags: Export Eval
  Holdtime: 0
  Total peers: 2        Established: 2
  192.168.2.1+54883
  192.168.7.1+58933
  bgp.l3vpn.0: 0/0/0/0
  bgp.mvpn.0: 0/0/0/0

Groups: 1 Peers: 2   External: 0   Internal: 2   Down peers: 0  Flaps: 0
Table        Tot Paths Act Paths Suppressed  History Damp State  Pending
bgp.l3vpn.0          0          0          0        0          0        0
bgp.mvpn.0           0          0          0        0          0        0
```

Verify that the peer state for the other two PE routers is **Established** and that the **lo0.0** addresses of the other PE routers are shown as peers.

Configuring LDP

Step-by-Step Procedure

1. On the PE routers, configure LDP to support unicast traffic. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces between the PE routers. Also configure LDP specifying the **lo0.0** interface. As a best practice, disable LDP on the **fxp0** interface.

```
user@PE1# set protocols ldp deaggregate
user@PE1# set protocols ldp interface fe-0/1/1.0
user@PE1# set protocols ldp interface ge-0/3/0.0
user@PE1# set protocols ldp interface fxp0.0 disable
user@PE1# set protocols ldp interface lo0.0

user@PE2# set protocols ldp deaggregate
user@PE2# set protocols ldp interface fe-0/1/3.0
user@PE2# set protocols ldp interface ge-1/3/0.0
user@PE2# set protocols ldp interface fxp0.0 disable
user@PE2# set protocols ldp interface lo0.0

user@PE3# set protocols ldp deaggregate
user@PE3# set protocols ldp interface fe-0/1/1.0
user@PE3# set protocols ldp interface fe-0/1/3.0
user@PE3# set protocols ldp interface fxp0.0 disable
user@PE3# set protocols ldp interface lo0.0
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit
```

```
commit complete
```

3. On the PE routers, use the **show ldp route** command to verify the LDP route.

```
user@PE1> show ldp route

Destination          Next-hop intf/lsp              Next-hop address
 10.0.12.8/30        ge-0/3/0.0
 10.0.12.9/32
 10.0.17.12/30       fe-0/1/1.0
 10.0.17.13/32
 10.0.27.12/30       fe-0/1/1.0                     10.0.17.14
                     ge-0/3/0.0                     10.0.12.10
 192.168.1.1/32      lo0.0
 192.168.2.1/32      ge-0/3/0.0                     10.0.12.10
 192.168.7.1/32      fe-0/1/1.0                     10.0.17.14
 224.0.0.5/32
 224.0.0.22/32
```

Verify that a next-hop interface and next-hop address have been established for each remote destination in the core network. Notice that local destinations do not have next-hop interfaces, and remote destinations outside the core do not have next-hop addresses.

Configuring RSVP

Step-by-Step Procedure

1. On the PE routers, configure RSVP. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces that participate in the LSP. Also specify the **lo0.0** interface. As a best practice, disable RSVP on the **fxp0** interface.

```
user@PE1# set protocols rsvp interface ge-0/3/0.0
user@PE1# set protocols rsvp interface fe-0/1/1.0
user@PE1# set protocols rsvp interface lo0.0
user@PE1# set protocols rsvp interface fxp0.0 disable

user@PE2# set protocols rsvp interface fe-0/1/3.0
user@PE2# set protocols rsvp interface ge-1/3/0.0
user@PE2# set protocols rsvp interface lo0.0
user@PE2# set protocols rsvp interface fxp0.0 disable

user@PE3# set protocols rsvp interface fe-0/1/3.0
user@PE3# set protocols rsvp interface fe-0/1/1.0
user@PE3# set protocols rsvp interface lo0.0
user@PE3# set protocols rsvp interface fxp0.0 disable
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

Verify these steps using the **show configuration protocols rsvp** command. You can verify the operation of RSVP only after the LSP is established.

Configuring MPLS

Step-by-Step Procedure

1. On the PE routers, configure MPLS. Specify the core-facing Fast Ethernet and Gigabit Ethernet interfaces that participate in the LSP. As a best practice, disable MPLS on the fxp0 interface.

   ```
   user@PE1# set protocols mpls interface ge-0/3/0.0
   user@PE1# set protocols mpls interface fe-0/1/1.0
   user@PE1# set protocols mpls interface fxp0.0 disable

   user@PE2# set protocols mpls interface fe-0/1/3.0
   user@PE2# set protocols mpls interface ge-1/3/0.0
   user@PE2# set protocols mpls interface fxp0.0 disable

   user@PE3# set protocols mpls interface fe-0/1/3.0
   user@PE3# set protocols mpls interface fe-0/1/1.0
   user@PE3# set protocols mpls interface fxp0.0 disable
   ```

   Use the **show configuration protocols mpls** command to verify that the core-facing Fast Ethernet and Gigabit Ethernet interfaces are configured for MPLS.

2. On the PE routers, configure the core-facing interfaces associated with the LSP. Specify the **mpls** address family type.

   ```
   user@PE1# set interfaces fe-0/1/1 unit 0 family mpls
   user@PE1# set interfaces ge-0/3/0 unit 0 family mpls

   user@PE2# set interfaces fe-0/1/3 unit 0 family mpls
   user@PE2# set interfaces ge-1/3/0 unit 0 family mpls

   user@PE3# set interfaces fe-0/1/3 unit 0 family mpls
   user@PE3# set interfaces fe-0/1/1 unit 0 family mpls
   ```

   Use the **show mpls interface** command to verify that the core-facing interfaces have the MPLS address family configured.

3. On the PE routers, commit the configuration:

   ```
   user@host> commit check

   configuration check succeeds

   user@host> commit

   commit complete
   ```

   You can verify the operation of MPLS after the LSP is established.

Configuring the VRF Routing Instances

Step-by-Step Procedure

1. On Router PE1 , configure the routing instance for the green and red VPNs. Specify the **vrf** instance type and specify the customer-facing SONET interfaces.

Configure a virtual tunnel (VT) interface on all MVPN routing instances on each PE where hosts in different instances need to receive multicast traffic from the same source.

```
user@PE1# set routing-instances green instance-type vrf
user@PE1# set routing-instances green interface so-0/0/3.0
user@PE1# set routing-instances green interface vt-1/2/0.1 multicast
user@PE1# set routing-instances green interface lo0.1

user@PE1# set routing-instances red instance-type vrf
user@PE1# set routing-instances red interface fe-0/1/0.0
user@PE1# set routing-instances red interface vt-1/2/0.2
user@PE1# set routing-instances red interface lo0.2
```

Use the **show configuration routing-instances green** and **show configuration routing-instances red** commands to verify that the virtual tunnel interfaces have been correctly configured.

2. On Router PE2 , configure the routing instance for the green VPN. Specify the **vrf** instance type and specify the customer-facing SONET interfaces.

```
user@PE2# set routing-instances green instance-type vrf
user@PE2# set routing-instances green interface so-0/0/1.0
user@PE2# set routing-instances green interface vt-1/2/0.1
user@PE2# set routing-instances green interface lo0.1
```

Use the **show configuration routing-instances green** command.

3. On Router PE3, configure the routing instance for the blue VPN. Specify the **vrf** instance type and specify the customer-facing SONET interfaces.

```
user@PE3# set routing-instances blue instance-type vrf
user@PE3# set routing-instances blue interface so-0/0/1.0
user@PE3# set routing-instances blue interface vt-1/2/0.3
user@PE3# set routing-instances blue interface lo0.1
```

Use the **show configuration routing-instances blue** command to verify that the instance type has been configured correctly and that the correct interfaces have been configured in the routing instance.

4. On Router PE1, configure a route distinguisher for the green and red routing instances. A route distinguisher allows the router to distinguish between two identical IP prefixes used as VPN routes.

TIP: To help in troubleshooting, this example shows how to configure the route distinguisher to match the router ID. This allows you to associate a route with the router that advertised it.

```
user@PE1# set routing-instances green route-distinguisher 192.168.1.1:1
user@PE1# set routing-instances red route-distinguisher 192.168.1.1:2
```

5. On Router PE2, configure a route distinguisher for the green routing instance.

```
user@PE2# set routing-instances green route-distinguisher 192.168.2.1:1
```

6. On Router PE3, configure a route distinguisher for the blue routing instance.

```
user@PE3# set routing-instances blue route-distinguisher 192.168.7.1:3
```

7. On the PE routers, configure the VPN routing instance for multicast support.

```
user@PE1# set routing-instances green protocols mvpn
user@PE1# set routing-instances red protocols mvpn

user@PE2# set routing-instances green protocols mvpn

user@PE3# set routing-instances blue protocols mvpn
```

   Use the **show configuration routing-instance** command to verify that the route distinguisher is configured correctly and that the MVPN Protocol is enabled in the routing instance.

8. On the PE routers, configure an IP address on additional loopback logical interfaces. These logical interfaces are used as the loopback addresses for the VPNs.

```
user@PE1# set interfaces lo0 unit 1 description "green VRF loopback"
user@PE1# set interfaces lo0 unit 1 family inet address 10.10.1.1/32
user@PE1# set interfaces lo0 unit 2 description "red VRF loopback"
user@PE1# set interfaces lo0 unit 2 family inet address 10.2.1.1/32

user@PE2# set interfaces lo0 unit 1 description "green VRF loopback"
user@PE2# set interfaces lo0 unit 1 family inet address 10.10.22.2/32

user@PE3# set interfaces lo0 unit 1 description "blue VRF loopback"
user@PE3# set interfaces lo0 unit 1 family inet address 10.3.33.3/32
```

   Use the **show interfaces terse** command to verify that the loopback logical interfaces are correctly configured.

9. On the PE routers, configure virtual tunnel interfaces. These interfaces are used in VRF instances where multicast traffic arriving on a provider tunnel needs to be forwarded to multiple VPNs.

```
user@PE1# set interfaces vt-1/2/0 unit 1 description "green VRF multicast vt"
user@PE1# set interfaces vt-1/2/0 unit 1 family inet
user@PE1# set interfaces vt-1/2/0 unit 2 description "red VRF unicast and multicast
  vt"
user@PE1# set interfaces vt-1/2/0 unit 2 family inet
user@PE1# set interfaces vt-1/2/0 unit 3 description "blue VRF multicast vt"
user@PE1# set interfaces vt-1/2/0 unit 3 family inet

user@PE2# set interfaces vt-1/2/0 unit 1 description "green VRF unicast and
  multicast vt"
user@PE2# set interfaces vt-1/2/0 unit 1 family inet
user@PE2# set interfaces vt-1/2/0 unit 3 description "blue VRF unicast and multicast
  vt"
user@PE2# set interfaces vt-1/2/0 unit 3 family inet

user@PE3# set interfaces vt-1/2/0 unit 3 description "blue VRF unicast and multicast
  vt"
user@PE3# set interfaces vt-1/2/0 unit 3 family inet
```

Use the **show interfaces terse** command to verify that the virtual tunnel interfaces have the correct address family type configured.

10. On the PE routers, configure the provider tunnel.

```
user@PE1# set routing-instances green provider-tunnel rsvp-te
  label-switched-path-template default-template
user@PE1# set routing-instances red provider-tunnel rsvp-te
  label-switched-path-template default-template

user@PE2# set routing-instances green provider-tunnel rsvp-te
  label-switched-path-template default-template

user@PE3# set routing-instances blue provider-tunnel rsvp-te
  label-switched-path-template default-template
```

Use the **show configuration routing-instance** command to verify that the provider tunnel is configured to use the default LSP template.

NOTE: You cannot commit the configuration for the VRF instance until you configure the VRF target in the next section.

Configuring MVPN Extranet Policy

Step-by-Step Procedure

1. On the PE routers, define the VPN community name for the route targets for each VPN. The community names are used in the VPN import and export policies.

```
user@PE1# set policy-options community green-com members target:65000:1
user@PE1# set policy-options community red-com members target:65000:2
user@PE1# set policy-options community blue-com members target:65000:3

user@PE2# set policy-options community green-com members target:65000:1
user@PE2# set policy-options community red-com members target:65000:2
user@PE2# set policy-options community blue-com members target:65000:3

user@PE3# set policy-options community green-com members target:65000:1
user@PE3# set policy-options community red-com members target:65000:2
user@PE3# set policy-options community blue-com members target:65000:3
```

Use the **show policy-options** command to verify that the correct VPN community name and route target are configured.

2. On the PE routers, configure the VPN import policy. Include the community name of the route targets that you want to accept. Do not include the community name of the route targets that you do not want to accept. For example, omit the community name for routes from the VPN of a multicast sender from which you do not want to receive multicast traffic.

```
user@PE1# set policy-options policy-statement green-red-blue-import term t1 from
  community green-com
user@PE1# set policy-options policy-statement green-red-blue-import term t1 from
  community red-com
```

```
user@PE1# set policy-options policy-statement green-red-blue-import term t1 from
  community blue-com
user@PE1# set policy-options policy-statement green-red-blue-import term t1 then
  accept
user@PE1# set policy-options policy-statement green-red-blue-import term t2 then
  reject

user@PE2# set policy-options policy-statement green-red-blue-import term t1 from
  community green-com
user@PE2# set policy-options policy-statement green-red-blue-import term t1 from
  community red-com
user@PE2# set policy-options policy-statement green-red-blue-import term t1 from
  community blue-com
user@PE2# set policy-options policy-statement green-red-blue-import term t1 then
  accept
user@PE2# set policy-options policy-statement green-red-blue-import term t2 then
  reject

user@PE3# set policy-options policy-statement green-red-blue-import term t1 from
  community green-com
user@PE3# set policy-options policy-statement green-red-blue-import term t1 from
  community red-com
user@PE3# set policy-options policy-statement green-red-blue-import term t1 from
  community blue-com
user@PE3# set policy-options policy-statement green-red-blue-import term t1 then
  accept
user@PE3# set policy-options policy-statement green-red-blue-import term t2 then
  reject
```

Use the **show policy green-red-blue-import** command to verify that the VPN import policy is correctly configured.

3. On the PE routers, apply the VRF import policy. In this example, the policy is defined in a **policy-statement** policy, and target communities are defined under the **[edit policy-options]** hierarchy level.

```
user@PE1# set routing-instances green vrf-import green-red-blue-import
user@PE1# set routing-instances red vrf-import green-red-blue-import

user@PE2# set routing-instances green vrf-import green-red-blue-import

user@PE3# set routing-instances blue vrf-import green-red-blue-import
```

Use the **show configuration routing-instances** command to verify that the correct VRF import policy has been applied.

4. On the PE routers, configure VRF export targets. The **vrf-target** statement and **export** option cause the routes being advertised to be labeled with the target community.

For Router PE3, the **vrf-target** statement is included without specifying the **export** option. If you do not specify the **import** or **export** options, default VRF import and export policies are generated that accept imported routes and tag exported routes with the specified target community.

NOTE: You must configure the same route target on each PE router for a given VPN routing instance.

```
user@PE1# set routing-instances green vrf-target export target:65000:1
user@PE1# set routing-instances red vrf-target export target:65000:2

user@PE2# set routing-instances green vrf-target export target:65000:1

user@PE3# set routing-instances blue vrf-target target:65000:3
```

Use the **show configuration routing-instances** command to verify that the correct VRF export targets have been configured.

5. On the PE routers, configure automatic exporting of routes between VRF instances. When you include the **auto-export** statement, the **vrf-import** and **vrf-export** policies are compared across all VRF instances. If there is a common route target community between the instances, the routes are shared. In this example, the **auto-export** statement must be included under all instances that need to send traffic to and receive traffic from another instance located on the same router.

```
user@PE1# set routing-instances green routing-options auto-export
user@PE1# set routing-instances red routing-options auto-export

user@PE2# set routing-instances green routing-options auto-export

user@PE3# set routing-instances blue routing-options auto-export
```

6. On the PE routers, configure the load balance policy statement. While load balancing leads to better utilization of the available links, it is not required for MVPN extranets. It is included here as a best practice.

```
user@PE1# set policy-options policy-statement load-balance then load-balance
  per-packet

user@PE2# set policy-options policy-statement load-balance then load-balance
  per-packet

user@PE3# set policy-options policy-statement load-balance then load-balance
  per-packet
```

Use the **show policy-options** command to verify that the load balance policy statement has been correctly configured.

7. On the PE routers, apply the load balance policy.

```
user@PE1# set routing-options forwarding-table export load-balance

user@PE2# set routing-options forwarding-table export load-balance

user@PE3# set routing-options forwarding-table export load-balance
```

8. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

9. On the PE routers, use the **show rsvp neighbor** command to verify that the RSVP neighbors are established.

```
user@PE1> show rsvp neighbor

RSVP neighbor: 2 learned
Address            Idle Up/Dn LastChange HelloInt HelloTx/Rx MsgRcvd
10.0.17.14            5  1/0       43:52        9   293/293  247
10.0.12.10            0  1/0       50:15        9   336/336  140
```

Verify that the other PE routers are listed as RSVP neighbors.

10. On the PE routers, display the MPLS LSPs.

```
user@PE1> show mpls lsp p2mp

Ingress LSP: 2 sessions
P2MP name: 192.168.1.1:1:mvpn:green, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
192.168.2.1     192.168.1.1     Up     0 *                      192.168.2.1:192.168.1.1:1:mvpn:green
192.168.7.1     192.168.1.1     Up     0 *                      192.168.7.1:192.168.1.1:1:mvpn:green
P2MP name: 192.168.1.1:2:mvpn:red, P2MP branch count: 2
To              From            State Rt P     ActivePath       LSPname
192.168.2.1     192.168.1.1     Up     0 *                      192.168.2.1:192.168.1.1:2:mvpn:red
192.168.7.1     192.168.1.1     Up     0 *                      192.168.7.1:192.168.1.1:2:mvpn:red
Total 4 displayed, Up 4, Down 0

Egress LSP: 2 sessions
P2MP name: 192.168.2.1:1:mvpn:green, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.1     192.168.2.1     Up       0  1 SE  299888        3 192.168.1.1:192.168.2.1:1:mvpn:green
P2MP name: 192.168.7.1:3:mvpn:blue, P2MP branch count: 1
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.1     192.168.7.1     Up       0  1 SE  299872        3 192.168.1.1:192.168.7.1:3:mvpn:blue
Total 2 displayed, Up 2, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

In this display from Router PE1, notice that there are two ingress LSPs for the green VPN and two for the red VPN configured on this router. Verify that the state of each ingress LSP is **up**. Also notice that there is one egress LSP for each of the green and blue VPNs. Verify that the state of each egress LSP is **up**.

TIP: The LSP name displayed in the show mpls lsp p2mp command output can be used in the ping mpls rsvp <*lsp-name*> multipath command.

Configuring CE-PE BGP

Step-by-Step Procedure

1. On the PE routers, configure the BGP export policy. The BGP export policy is used to allow static routes and routes that originated from directly attached interfaces to be exported to BGP.

```
user@PE1# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@PE1# set policy-options policy-statement BGP-export term t1 then accept
user@PE1# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@PE1# set policy-options policy-statement BGP-export term t2 then accept

user@PE2# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@PE2# set policy-options policy-statement BGP-export term t1 then accept
user@PE2# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@PE2# set policy-options policy-statement BGP-export term t2 then accept

user@PE3# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@PE3# set policy-options policy-statement BGP-export term t1 then accept
user@PE3# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@PE3# set policy-options policy-statement BGP-export term t2 then accept
```

   Use the **show policy BGP-export** command to verify that the BGP export policy is correctly configured.

2. On the PE routers, configure the CE to PE BGP session. Use the IP address of the SONET interface as the neighbor address. Specify the autonomous system number for the VPN network of the attached CE router.

```
user@PE1# set routing-instances green protocols bgp group PE-CE export
  BGP-export
user@PE1# set routing-instances green protocols bgp group PE-CE neighbor 10.0.16.1
  peer-as 65001

user@PE2# set routing-instances green protocols bgp group PE-CE export
  BGP-export
user@PE2# set routing-instances green protocols bgp group PE-CE neighbor
  10.0.24.2 peer-as 65009

user@PE3# set routing-instances blue protocols bgp group PE-CE export BGP-export
user@PE3# set routing-instances blue protocols bgp group PE-CE neighbor 10.0.79.2
  peer-as 65003
```

3. On the CE routers, configure the BGP local autonomous system number.

```
user@CE1# set routing-options autonomous-system 65001

user@CE2# set routing-options autonomous-system 65009

user@CE3# set routing-options autonomous-system 65003
```

4. On the CE routers, configure the BGP export policy. The BGP export policy is used to allow static routes and routes that originated from directly attached interfaces to be exported to BGP.

```
user@CE1# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@CE1# set policy-options policy-statement BGP-export term t1 then accept
user@CE1# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@CE1# set policy-options policy-statement BGP-export term t2 then accept

user@CE2# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@CE2# set policy-options policy-statement BGP-export term t1 then accept
user@CE2# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@CE2# set policy-options policy-statement BGP-export term t2 then accept

user@CE3# set policy-options policy-statement BGP-export term t1 from protocol
  direct
user@CE3# set policy-options policy-statement BGP-export term t1 then accept
user@CE3# set policy-options policy-statement BGP-export term t2 from protocol
  static
user@CE3# set policy-options policy-statement BGP-export term t2 then accept
```

   Use the **show policy BGP-export** command to verify that the BGP export policy is correctly configured.

5. On the CE routers, configure the CE-to-PE BGP session. Use the IP address of the SONET interface as the neighbor address. Specify the autonomous system number of the core network. Apply the BGP export policy.

```
user@CE1# set protocols bgp group PE-CE export BGP-export
user@CE1# set protocols bgp group PE-CE neighbor 10.0.16.2 peer-as 65000

user@CE2# set protocols bgp group PE-CE export BGP-export
user@CE2# set protocols bgp group PE-CE neighbor 10.0.24.1 peer-as 65000

user@CE3# set protocols bgp group PE-CE export BGP-export
user@CE3# set protocols bgp group PE-CE neighbor 10.0.79.1 peer-as 65000
```

6. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

7. On the PE routers, use the **show bgp group pe-ce** command to verify that the BGP neighbors form a peer session.

```
user@PE1> show bgp group pe-ce
```

```
Group Type: External                                    Local AS: 65000
  Name: PE-CE           Index: 1                        Flags: <>
  Export: [ BGP-export ]
  Holdtime: 0
  Total peers: 1        Established: 1
  10.0.16.1+60500
  green.inet.0: 2/3/3/0
```

Verify that the peer state for the CE routers is **Established** and that the IP address configured on the peer SONET interface is shown as the peer.

Configuring PIM on the PE Routers

Step-by-Step Procedure

1. On the PE routers, enable an instance of PIM in each VPN. Configure the **lo0.1**, **lo0.2**, and customer-facing SONET and Fast Ethernet interfaces. Specify the mode as **sparse**.

```
user@PE1# set routing-instances green protocols pim interface lo0.1 mode sparse
user@PE1# set routing-instances green protocols pim interface so-0/0/3.0 mode
  sparse
user@PE1# set routing-instances red protocols pim interface lo0.2 mode sparse
user@PE1# set routing-instances red protocols pim interface fe-0/1/0.0 mode
  sparse

user@PE2# set routing-instances green protocols pim interface lo0.1 mode sparse
user@PE2# set routing-instances green protocols pim interface so-0/0/1.0 mode
  sparse

user@PE3# set routing-instances blue protocols pim interface lo0.1 mode sparse
user@PE3# set routing-instances blue protocols pim interface so-0/0/1.0 mode
  sparse
```

2. On the PE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the PE routers, use the **show pim interfaces instance green** command and substitute the appropriate VRF instance name to verify that the PIM interfaces are **up**.

```
user@PE1> show pim interfaces instance green

Instance: PIM.green

Name               Stat Mode       IP V State NbrCnt JoinCnt DR address
lo0.1              Up   Sparse      4 2 DR         0       0 10.10.1.1
lsi.0              Up   SparseDense 4 2 P2P        0       0
pe-1/2/0.32769     Up   Sparse      4 2 P2P        0       0
so-0/0/3.0         Up   Sparse      4 2 P2P        1       2
vt-1/2/0.1         Up   SparseDense 4 2 P2P        0       0
lsi.0              Up   SparseDense 6 2 P2P        0       0
```

Also notice that the normal mode for the virtual tunnel interface and label-switched interface is **SparseDense**.

Configuring PIM on the CE Routers

Step-by-Step Procedure

1. On the CE routers, configure the customer-facing and core-facing interfaces for PIM. Specify the mode as **sparse**.

```
user@CE1# set protocols pim interface fe-1/3/0.0 mode sparse
user@CE1# set protocols pim interface so-0/0/3.0 mode sparse

user@CE2# set protocols pim interface fe-0/1/1.0 mode sparse
user@CE2# set protocols pim interface so-0/0/1.0 mode sparse

user@CE3# set protocols pim interface fe-0/1/0.0 mode sparse
user@CE3# set protocols pim interface so-0/0/1.0 mode sparse
```

   Use the **show pim interfaces** command to verify that the PIM interfaces have been configured to use sparse mode.

2. On the CE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

3. On the CE routers, use the **show pim interfaces** command to verify that the PIM interface status is **up**.

```
user@CE1> show pim interfaces

Instance: PIM.master

Name               Stat Mode       IP V State NbrCnt JoinCnt DR address
fe-1/3/0.0         Up   Sparse      4 2 DR         0       0 10.10.12.1
pe-1/2/0.32769     Up   Sparse      4 2 P2P        0       0
so-0/0/3.0         Up   Sparse      4 2 P2P        1       1
```

Configuring the Rendezvous Points

Step-by-Step Procedure

1. Configure Router PE1 to be the rendezvous point for the red VPN instance of PIM. Specify the local **lo0.2** address.

```
user@PE1# set routing-instances red protocols pim rp local address 10.2.1.1
```

2. Configure Router PE2 to be the rendezvous point for the green VPN instance of PIM. Specify the **lo0.1** address of Router PE2.

```
user@PE2# set routing-instances green protocols pim rp local address 10.10.22.2
```

3. Configure Router PE3 to be the rendezvous point for the blue VPN instance of PIM. Specify the local **lo0.1**.

```
user@PE3# set routing-instances blue protocols pim rp local address 10.3.33.3
```

4. On the PE1, CE1, and CE2 routers, configure the static rendezvous point for the green VPN instance of PIM. Specify the **lo0.1** address of Router PE2.

```
user@PE1# set routing-instances green protocols pim rp static address 10.10.22.2

user@CE1# set protocols pim rp static address 10.10.22.2

user@CE2# set protocols pim rp static address 10.10.22.2
```

5. On Router CE3, configure the static rendezvous point for the blue VPN instance of PIM. Specify the **lo0.1** address of Router PE3.

```
user@CE3# set protocols pim rp static address 10.3.33.3
```

6. On the CE routers, commit the configuration:

```
user@host> commit check

configuration check succeeds

user@host> commit

commit complete
```

7. On the PE routers, use the **show pim rps instance <*instance-name*>** command and substitute the appropriate VRF instance name to verify that the RPs have been correctly configured.

```
user@PE1> show pim rps instance <instance-name>

Instance: PIM.green
Address family INET
RP address              Type       Holdtime Timeout Groups Group prefixes
10.10.22.2              static             0    None      1 224.0.0.0/4

Address family INET6
```

Verify that the correct IP address is shown as the RP.

8. On the CE routers, use the **show pim rps** command to verify that the RP has been correctly configured.

```
user@CE1> show pim rps

Instance: PIM.master
Address family INET
RP address              Type       Holdtime Timeout Groups Group prefixes
10.10.22.2              static             0    None      1 224.0.0.0/4

Address family INET6
```

Verify that the correct IP address is shown as the RP.

9. On Router PE1, use the **show route table green.mvpn.0 | find 1** command to verify that the type-1 routes have been received from the PE2 and PE3 routers.

```
user@PE1> show route table green.mvpn.0 | find 1
```

```
green.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:192.168.1.1:1:192.168.1.1/240
                   *[MVPN/70] 03:38:09, metric2 1
                      Indirect
1:192.168.1.1:2:192.168.1.1/240
                   *[MVPN/70] 03:38:05, metric2 1
                      Indirect
1:192.168.2.1:1:192.168.2.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
1:192.168.7.1:3:192.168.7.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.7.1
                      AS path: I
                    > to 10.0.17.14 via fe-0/1/1.0
```

10. On Router PE1, use the **show route table green.mvpn.0 | find 5** command to verify that the type-5 routes have been received from Router PE2.

```
user@PE1> show route table green.mvpn.0 | find 5

5:192.168.2.1:1:32:10.10.12.52:32:224.1.1.1/240
                   *[BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
```

11. On Router PE1, use the **show route table green.mvpn.0 | find 7** command to verify that the type-7 routes have been received from Router PE2.

```
user@PE1> show route table green.mvpn.0 | find 7

7:192.168.1.1:1:65000:32:10.10.12.52:32:224.1.1.1/240
                   *[MVPN/70] 03:22:47, metric2 1
                      Multicast (IPv4)
                    [PIM/105] 03:34:18
                      Multicast (IPv4)
                    [BGP/170] 03:12:18, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 10.0.12.10 via ge-0/3/0.0
```

12. On Router PE1, use the **show route advertising-protocol bgp 192.168.2.1 table green.mvpn.0 detail** command to verify that the routes advertised by Router PE2 use the PMSI attribute set to RSVP-TE.

```
user@PE1> show route advertising-protocol bgp 192.168.2.1 table green.mvpn.0 detail

green.mvpn.0: 7 destinations, 9 routes (7 active, 1 holddown, 0 hidden)
* 1:192.168.1.1:1:192.168.1.1/240 (1 entry, 1 announced)
 BGP group group-mvpn type Internal
     Route Distinguisher: 192.168.1.1:1
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65000] I
     Communities: target:65000:1
     PMSI: Flags 0:RSVP-TE:label[0:0:0]:Session_13[192.168.1.1:0:56822:192.168.1.1]
```

Testing MVPN Extranets

Step-by-Step Procedure

1. Start the multicast receiver device connected to Router CE2.
2. Start the multicast sender device connected to Router CE1.
3. Verify that the receiver receives the multicast stream.
4. On Router PE1, display the provider tunnel to multicast group mapping by using the **show mvpn c-multicast** command.

```
user@PE1> show mvpn c-multicast

MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: green
  C-mcast IPv4 (S:G)            Ptnl                              St
  10.10.12.52/32:224.1.1.1/32   RSVP-TE P2MP:192.168.1.1, 56822,192.168.1.1         RM
  0.0.0.0/0:239.255.255.250/32
MVPN instance:

Legend for provider tunnel
I-P-tnl -- inclusive provider tunnel S-P-tnl -- selective provider tunnel

Legend for c-multicast routes properties (Pr)
DS -- derived from (*, c-g)          RM -- remote VPN route
Instance: red
  C-mcast IPv4 (S:G)            Ptnl                              St
  10.10.12.52/32:224.1.1.1/32                                            DS
  0.0.0.0/0:224.1.1.1/32
```

5. On Router PE2, use the **show route table green.mvpn.0 | find 6** command to verify that the type-6 routes have been created as a result of receiving PIM join messages.

```
user@PE2> show route table green.mvpn.0 | find 6

6:192.168.2.1:1:65000:32:10.10.22.2:32:224.1.1.1/240
                   *[PIM/105] 04:01:23
                      Multicast (IPv4)
6:192.168.2.1:1:65000:32:10.10.22.2:32:239.255.255.250/240

                   *[PIM/105] 22:39:46
                      Multicast (IPv4)
```

NOTE: The multicast address 239.255.255.250 shown in the preceding step is not related to this example. This address is sent by some host machines.

6. Start the multicast receiver device connected to Router CE3.
7. Verify that the receiver is receiving the multicast stream.

8. On Router PE2, use the **show route table green.mvpn.0 | find 6** command to verify that the type-6 routes have been created as a result of receiving PIM join messages from the multicast receiver device connected to Router CE3.

```
user@PE2> show route table green.mvpn.0 | find 6

6:192.168.2.1:1:65000:32:10.10.22.2:32:239.255.255.250/240
                   *[PIM/105] 06:43:39
                      Multicast (IPv4)
```

9. Start the multicast receiver device directly connected to Router PE1.

10. Verify that the receiver is receiving the multicast stream.

11. On Router PE1, use the **show route table green.mvpn.0 | find 6** command to verify that the type-6 routes have been created as a result of receiving PIM join messages from the directly connected multicast receiver device.

```
user@PE1> show route table green.mvpn.0 | find 6

6:192.168.1.1:2:65000:32:10.2.1.1:32:224.1.1.1/240
                   *[PIM/105] 00:02:32
                      Multicast (IPv4)
6:192.168.1.1:2:65000:32:10.2.1.1:32:239.255.255.250/240
                   *[PIM/105] 00:05:49
                      Multicast (IPv4)
```

NOTE: The multicast address 255.255.255.250 shown in the step above is not related to this example.

Results The configuration and verification parts of this example have been completed. The following section is for your reference.

The relevant sample configuration for Router CE1 follows.

Router CE1

```
interfaces {
  so-0/0/3 {
    unit 0 {
      description "to PE1 so-0/0/3.0";
      family inet {
        address 10.0.16.1/30;
      }
    }
  }
  fe-1/3/0 {
    unit 0 {
      family inet {
        address 10.10.12.1/24;
      }
    }
  }
  lo0 {
    unit 0 {
      description "CE1 Loopback";
```

```
        family inet {
          address 192.168.6.1/32 {
            primary;
          }
          address 127.0.0.1/32;
        }
      }
    }
}
routing-options {
    autonomous-system 65001;
    router-id 192.168.6.1;
    forwarding-table {
      export load-balance;
    }
}
protocols {
    bgp {
      group PE-CE {
        export BGP-export;
        neighbor 10.0.16.2 {
          peer-as 65000;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.10.22.2;
        }
      }
      interface fe-1/3/0.0 {
        mode sparse;
      }
      interface so-0/0/3.0 {
        mode sparse;
      }
    }
}
policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement load-balance {
      then {
        load-balance per-packet;
      }
    }
}
```

The relevant sample configuration for Router PE1 follows.

Router PE1

```
interfaces {
  so-0/0/3 {
    unit 0 {
      description "to CE1 so-0/0/3.0";
      family inet {
        address 10.0.16.2/30;
      }
    }
  }
  fe-0/1/0 {
    unit 0 {
      description "to H2";
      family inet {
        address 10.2.11.2/30;
      }
    }
  }
    fe-0/1/1 {
      unit 0 {
        description "to PE3 fe-0/1/1.0";
        family inet {
          address 10.0.17.13/30;
        }
        family mpls;
      }
    }
  ge-0/3/0 {
    unit 0 {
      description "to PE2 ge-1/3/0.0";
      family inet {
        address 10.0.12.9/30;
      }
      family mpls;
    }
  }
  vt-1/2/0 {
    unit 1 {
      description "green VRF multicast vt";
      family inet;
    }
    unit 2 {
      description "red VRF unicast and multicast vt";
      family inet;
    }
    unit 3 {
      description "blue VRF multicast vt";
      family inet;
    }
  }
  lo0 {
    unit 0 {
      description "PE1 Loopback";
      family inet {
        address 192.168.1.1/32 {
```

```
            primary;
          }
          address 127.0.0.1/32;
        }
      }
      unit 1 {
        description "green VRF loopback";
        family inet {
          address 10.10.1.1/32;
        }
      }
      unit 2 {
        description "red VRF loopback";
        family inet {
          address 10.2.1.1/32;
        }
      }
    }
}
routing-options {
    autonomous-system 65000;
    router-id 192.168.1.1;
    forwarding-table {
      export load-balance;
    }
}
protocols {
    rsvp {
      interface ge-0/3/0.0;
      interface fe-0/1/1.0;
      interface lo0.0;
      interface fxp0.0 {
        disable;
      }
    }
    mpls {
      interface ge-0/3/0.0;
      interface fe-0/1/1.0;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group group-mvpn {
        type internal;
        local-address 192.168.1.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.2.1;
        neighbor 192.168.7.1;
      }
    }
```

```
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface ge-0/3/0.0 {
          metric 100;
        }
        interface fe-0/1/1.0 {
          metric 100;
        }
        interface lo0.0 {
          passive;
        }
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      deaggregate;
      interface ge-0/3/0.0;
      interface fe-0/1/1.0;
      interface fxp0.0 {
        disable;
      }
      interface lo0.0;
    }
}
policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement green-red-blue-import {
      term t1 {
        from community [ green-com red-com blue-com ];
        then accept;
      }
      term t2 {
        then reject;
      }
    }
    policy-statement load-balance {
      then {
        load-balance per-packet;
      }
    }
    community green-com members target:65000:1;
    community red-com members target:65000:2;
    community blue-com members target:65000:3;
}
```

```
routing-instances {
   green {
      instance-type vrf;
      interface so-0/0/3.0;
      interface vt-1/2/0.1 {
         multicast;
      }
      interface lo0.1;
      route-distinguisher 192.168.1.1:1;
      provider-tunnel {
         rsvp-te {
            label-switched-path-template {
               default-template;
            }
         }
      }
      vrf-import green-red-blue-import;
      vrf-target export target:65000:1;
      vrf-table-label;
      routing-options {
         auto-export;
      }
      protocols {
         bgp {
            group PE-CE {
               export BGP-export;
               neighbor 10.0.16.1 {
                  peer-as 65001;
               }
            }
         }
         pim {
            rp {
               static {
                  address 10.10.22.2;
               }
            }
            interface so-0/0/3.0 {
               mode sparse;
            }
            interface lo0.1 {a
            mode sparse;
         }
      }
      mvpn;
   }
   red {
      instance-type vrf;
      interface fe-0/1/0.0;
      interface vt-1/2/0.2;
      interface lo0.2;
      route-distinguisher 192.168.1.1:2;
      provider-tunnel {
         rsvp-te {
            label-switched-path-template {
               default-template;
```

```
        }
      }
    }
    vrf-import green-red-blue-import;
    vrf-target export target:65000:2;
    routing-options {
      auto-export;
    }
    protocols {
      pim {
        rp {
          local {
            address 10.2.1.1;
          }
        }
        interface fe-0/1/0.0 {
          mode sparse;
        }
        interface lo0.2 {
          mode sparse;
        }
      }
      mvpn;
    }
  }
}
```

The relevant sample configuration for Router PE2 follows.

Router PE2

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to CE2 so-0/0/1:0.0";
      family inet {
        address 10.0.24.1/30;
      }
    }
  }
  fe-0/1/3 {
    unit 0 {
      description "to PE3 fe-0/1/3.0";
      family inet {
        address 10.0.27.13/30;
      }
      family mpls;
    }
    vt-1/2/0 {
      unit 1 {
        description "green VRF unicast and multicast vt";
        family inet;
      }
      unit 3 {
        description "blue VRF unicast and multicast vt";
        family inet;
      }
    }
```

```
}
ge-1/3/0 {
   unit 0 {
      description "to PE1 ge-0/3/0.0";
      family inet {
         address 10.0.12.10/30;
      }
      family mpls;
   }
}
lo0 {
   unit 0 {
      description "PE2 Loopback";
      family inet {
         address 192.168.2.1/32 {
            primary;
         }
         address 127.0.0.1/32;
      }
   }
   unit 1 {
      description "green VRF loopback";
      family inet {
         address 10.10.22.2/32;
      }
   }
}
routing-options {
   router-id 192.168.2.1;
   autonomous-system 65000;
   forwarding-table {
      export load-balance;
   }
}
protocols {
   rsvp {
      interface fe-0/1/3.0;
      interface ge-1/3/0.0;
      interface lo0.0;
      interface fxp0.0 {
         disable;
      }
   }
   mpls {
      interface fe-0/1/3.0;
      interface ge-1/3/0.0;
      interface fxp0.0 {
         disable;
      }
   }
   bgp {
      group group-mvpn {
         type internal;
         local-address 192.168.2.1;
         family inet-vpn {
            unicast;
```

```
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.1.1;
        neighbor 192.168.7.1;
      }
    }
    ospf {
      traffic-engineering;
      area 0.0.0.0 {
        interface fe-0/1/3.0 {
          metric 100;
        }
        interface ge-1/3/0.0 {
          metric 100;
        }
        interface lo0.0 {
          passive;
        }
        interface fxp0.0 {
          disable;
        }
      }
    }
    ldp {
      deaggregate;
      interface fe-0/1/3.0;
      interface ge-1/3/0.0;
      interface fxp0.0 {
        disable;
      }
      interface lo0.0;
    }
  }
  policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
      term t2 {
        from protocol static;
        then accept;
      }
    }
    policy-statement green-red-blue-import {
      term t1 {
        from community [ green-com red-com blue-com ];
        then accept;
      }
      term t2 {
        then reject;
      }
    }
    policy-statement load-balance {
```

```
        then {
          load-balance per-packet;
        }
      }
      community green-com members target:65000:1;
      community red-com members target:65000:2;
      community blue-com members target:65000:3;
    }
    routing-instances {
      green {
        instance-type vrf;
        interface so-0/0/1.0;
        interface vt-1/2/0.1;
        interface lo0.1;
        route-distinguisher 192.168.2.1:1;
        provider-tunnel {
          rsvp-te {
            label-switched-path-template {
              default-template;
            }
          }
        }
        vrf-import green-red-blue-import;
        vrf-target export target:65000:1;
        routing-options {
          auto-export;
        }
        protocols {
          bgp {
            group PE-CE {
              export BGP-export;
              neighbor 10.0.24.2 {
                peer-as 65009;
              }
            }
          }
          pim {
            rp {
              local {
                address 10.10.22.2;
              }
            }
            interface so-0/0/1.0 {
              mode sparse;
            }
            interface lo0.1 {
              mode sparse;
            }
          }
          mvpn;
        }
      }
    }
  }
```

The relevant sample configuration for Router CE2 follows.

Router CE2

```
interfaces {
  fe-0/1/1 {
    unit 0 {
      description "to H4";
      family inet {
        address 10.10.11.2/24;
      }
    }
  }
  so-0/0/1 {
    unit 0 {
      description "to PE2 so-0/0/1";
      family inet {
        address 10.0.24.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      description "CE2 Loopback";
      family inet {
        address 192.168.4.1/32 {
          primary;
        }
        address 127.0.0.1/32;
      }
    }
  }
}
routing-options {
  router-id 192.168.4.1;
  autonomous-system 65009;
  forwarding-table {
    export load-balance;
  }
}
protocols {
  bgp {
    group PE-CE {
      export BGP-export;
      neighbor 10.0.24.1 {
        peer-as 65000;
      }
    }
  }
  pim {
    rp {
      static {
        address 10.10.22.2;
      }
    }
    interface so-0/0/1.0 {
      mode sparse;
    }
    interface fe-0/1/1.0 {
      mode sparse;
```

```
    }
  }
}
policy-options {
  policy-statement BGP-export {
    term t1 {
      from protocol direct;
      then accept;
    }
    term t2 {
      from protocol static;
      then accept;
    }
  }
  policy-statement load-balance {
    then {
      load-balance per-packet;
    }
  }
}
```

The relevant sample configuration for Router PE3 follows.

Router PE3

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to CE3 so-0/0/1.0";
      family inet {
        address 10.0.79.1/30;
      }
    }
  }
  fe-0/1/1 {
    unit 0 {
      description "to PE1 fe-0/1/1.0";
      family inet {
        address 10.0.17.14/30;
      }
      family mpls;
    }
  }
  fe-0/1/3 {
    unit 0 {
      description "to PE2 fe-0/1/3.0";
      family inet {
        address 10.0.27.14/30;
      }
      family mpls;
    }
  }
  vt-1/2/0 {
    unit 3 {
      description "blue VRF unicast and multicast vt";
      family inet;
    }
  }
```

```
    lo0 {
      unit 0 {
        description "PE3 Loopback";
        family inet {
          address 192.168.7.1/32 {
            primary;
          }
          address 127.0.0.1/32;
        }
      }
      unit 1 {
        description "blue VRF loopback";
        family inet {
          address 10.3.33.3/32;
        }
      }
    }
}
routing-options {
    router-id 192.168.7.1;
    autonomous-system 65000;
    forwarding-table {
      export load-balance;
    }
}
protocols {
    rsvp {
      interface fe-0/1/3.0;
      interface fe-0/1/1.0;
      interface lo0.0;
      interface fxp0.0 {
        disable;
      }
    }
    mpls {
      interface fe-0/1/3.0;
      interface fe-0/1/1.0;
      interface fxp0.0 {
        disable;
      }
    }
    bgp {
      group group-mvpn {
        type internal;
        local-address 192.168.7.1;
        family inet-vpn {
          unicast;
        }
        family inet-mvpn {
          signaling;
        }
        neighbor 192.168.1.1;
        neighbor 192.168.2.1;
      }
    }
    ospf {
```

```
        traffic-engineering;
        area 0.0.0.0 {
          interface fe-0/1/3.0 {
            metric 100;
          }
          interface fe-0/1/1.0 {
            metric 100;
          }
          interface lo0.0 {
            passive;
          }
          interface fxp0.0 {
            disable;
          }
        }
      }
      ldp {
        deaggregate;
        interface fe-0/1/3.0;
        interface fe-0/1/1.0;
        interface fxp0.0 {
          disable;
        }
        interface lo0.0;
      }
    }
    policy-options {
      policy-statement BGP-export {
        term t1 {
          from protocol direct;
          then accept;
        }
        term t2 {
          from protocol static;
          then accept;
        }
      }
      policy-statement green-red-blue-import {
        term t1 {
          from community [ green-com red-com blue-com ];
          then accept;
        }
        term t2 {
          then reject;
        }
      }
      policy-statement load-balance {
        then {
          load-balance per-packet;
        }
      }
      community green-com members target:65000:1;
      community red-com members target:65000:2;
      community blue-com members target:65000:3;
    }
    routing-instances {
```

```
    blue {
      instance-type vrf;
      interface vt-1/2/0.3;
      interface so-0/0/1.0;
      interface lo0.1;
      route-distinguisher 192.168.7.1:3;
      provider-tunnel {
        rsvp-te {
          label-switched-path-template {
            default-template;
          }
        }
      }
      vrf-import green-red-blue-import;
      vrf-target target:65000:3;
      routing-options {
        auto-export;
      }
      protocols {
        bgp {
          group PE-CE {
            export BGP-export;
            neighbor 10.0.79.2 {
              peer-as 65003;
            }
          }
        }
        pim {
          rp {
            local {
              address 10.3.33.3;
            }
          }
          interface so-0/0/1.0 {
            mode sparse;
          }
          interface lo0.1 {
            mode sparse;
          }
        }
        mvpn ;
      }
    }
  }
```

The relevant sample configuration for Router CE3 follows.

Router CE3

```
interfaces {
  so-0/0/1 {
    unit 0 {
      description "to PE3";
      family inet {
        address 10.0.79.2/30;
      }
    }
  }
```

```
    fe-0/1/0 {
      unit 0 {
        description "to H3";
        family inet {
          address 10.3.11.3/24;
        }
      }
    }
    lo0 {
      unit 0 {
        description "CE3 loopback";
        family inet {
          address 192.168.9.1/32 {
            primary;
          }
          address 127.0.0.1/32;
        }
      }
    }
}
routing-options {
    router-id 192.168.9.1;
    autonomous-system 65003;
    forwarding-table {
      export load-balance;
    }
}
protocols {
    bgp {
      group PE-CE {
        export BGP-export;
        neighbor 10.0.79.1 {
          peer-as 65000;
        }
      }
    }
    pim {
      rp {
        static {
          address 10.3.33.3;
        }
      }
      interface so-0/0/1.0 {
        mode sparse;
      }
      interface fe-0/1/0.0 {
        mode sparse;
      }
    }
}
policy-options {
    policy-statement BGP-export {
      term t1 {
        from protocol direct;
        then accept;
      }
```

```
    term t2 {
      from protocol static;
      then accept;
    }
  }
  policy-statement load-balance {
    then {
      load-balance per-packet;
    }
  }
}
```

Related Documentation

- MBGP Multicast VPN Extranets Configuration Guidelines on page 50
- MBGP Multicast VPN Extranets Overview on page 49

For More Information

For additional information about multicast over Layer 3 VPNs, see the following resources:

- *Junos Multicast Protocols Configuration Guide*
- *Junos VPNs Configuration Guide*
- *RFC 2547, BGP/MPLS VPNs*
- *RFC 3618, Multicast Source Discovery Protocol (MSDP)*
- *RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs)*
- Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs* (expires April 2004)

CHAPTER 4

Configuring Draft Rosen VPNs

Draft-rosen multicast VPN with provider tunnels operating in any-source multicast (ASM) mode is a legacy feature. We recommend that you implement multiprotocol BGP-based multicast VPNs for multicast VPNs. Use this section to maintain your already existing draft-rosen multicast VPNs with provider tunnels operating in ASM mode. To implement this type of draft-rosen IPv4 multicast for a Layer 3 VPN, configure the following:

Dual PIM Draft-Rosen Multicast VPN Operation

The operation of draft-rosen multicast within a Layer 3 VPN domain with provider tunnels operating in any-source (ASM) multicast mode occurs in multiple stages, which are shown in Figure 4 on page 92 and described on the following pages.

Figure 4: Multicast Over Layer 3 VPN Operation

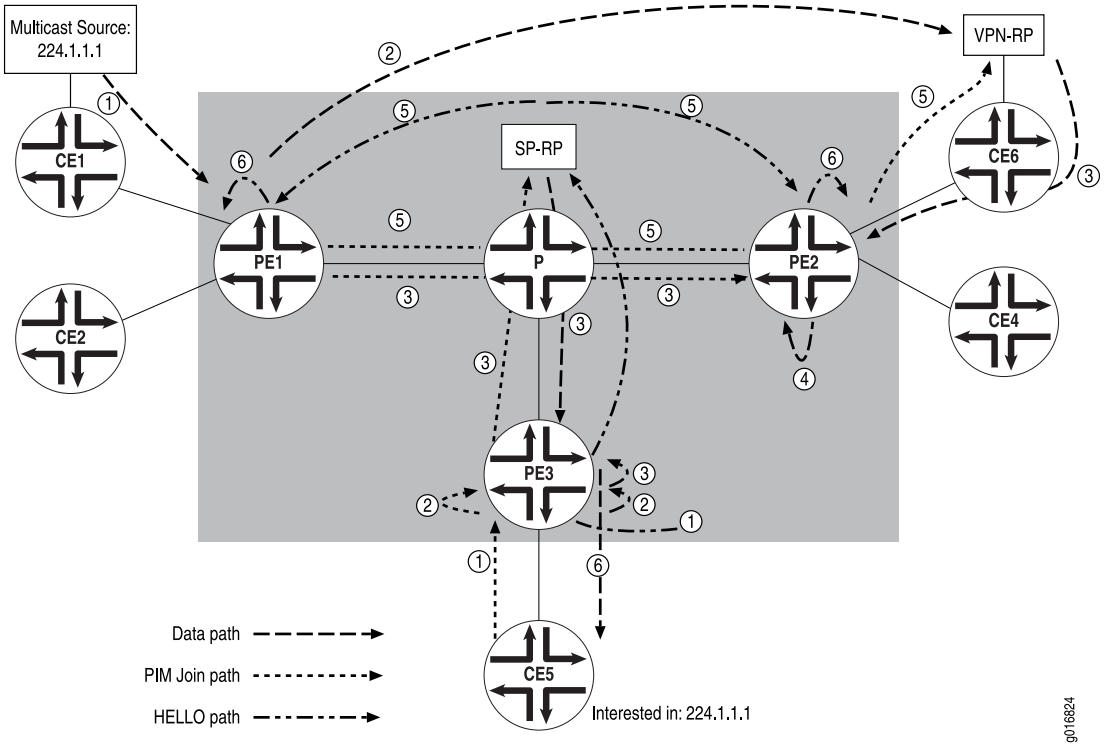


Figure 4 on page 92 shows the various stages that multicast packets pass through in a Layer 3 VPN environment.

- Stage 1: PIM hello messages

1. PIM is configured as part of a VPN routing instance and the configuration is committed. For Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, a virtual multicast tunnel interface (**mt-*fpc/pic/port.abcde***) is created if a Tunnel Services Physical Interface Card (PIC) is installed on the router. On Juniper Networks MX Series Ethernet Services Routers, you can create a virtual multicast tunnel interface by including the **tunnel-services** statement at the [**edit chassis fpc *slot-number* pic *number***] hierarchy level. For more information about configuring tunnel interfaces on MX Series routers, see the *Junos System Basics Configuration Guide*. The virtual multicast tunnel interface is used to communicate between the PIM instance within the VRF and the master PIM instance.
2. A PIM hello message is sent from the VRF across the **mt** interface. When this happens, a GRE header is prepended to the PIM hello message with fields containing the VPN group address and the loopback address of the PE router.
3. A PIM register header is prepended to the hello message as the packet is looped through the **pe** (PIM encapsulation) interface. This header contains the destination address of the SP-RP and the loopback address of the PE router.
4. The packet is sent to the SP-RP.

5. The SP-RP de-encapsulates the top header off the packet as it travels through the **pd** (PIM de-encapsulation) interface and sends the remaining GRE encapsulated hello message to all of the PE routers.
6. The master PIM instance on the PE router handles the GRE encapsulated packet. Because the VPN group address is contained in the packet, the master PIM instance de-encapsulates the packet and sends the hello message over the **mt** interface to reach the desired VPN group address within the VRF.

- Stage 2: PIM join message

1. Router CE5 is interested in receiving from multicast source **224.1.1.1**, so a PIM Join message is sent from Router CE5 to Router PE3.
2. The PIM Join message is sent through the **mt** interface and a GRE header is prepended to it. The GRE header contains the VPN group ID and the loopback address of Router PE3.
3. The GRE encapsulated Join message is sent to other PE routers.
4. Router PE2 receives the packet. Because the VPN C-RP is behind Router PE2, Router PE2 sends the packet through the **mt** interface, which strips off the GRE header.
5. The PIM Join message is now sent to the VPN C-RP.

- Stage 3: Multicast forwarding

1. The source behind Router CE1 is sending to group **224.1.1.1**. The designated router (DR) behind the CE router encapsulates this packet into a PIM register.
2. Because the packet already has the PIM register header, it is forwarded to the VPN C-RP by unicast routing over the Layer 3 VPN.
3. The VPN C-RP de-encapsulates the data packet and sends it out the downstream interfaces (which include the return path interface leading to Router PE3). Router P also forwards the packet to Router PE3.
4. The data packet is sent through the **mt** interface on Router PE2. In the process, the GRE header is prepended to the packet.
5. The packet is sent to the PE routers with GRE header intact.
6. The "interested" PE routers strip the GRE header off the packet and forward it to the CE routers that requested the PIM join. If there are no PIM-join messages for this group at this site, the PE router drops the packet.

When PIM is configured within a routing instance, two **mt** interfaces are created:

- **mt-[*xxxxx*]**(*xxxxx*) range is 32768 through 49151 for **mt-encap**
- **mt-[*yyyyy*]**(*yyyyy*) range is 49152 through 65535 for **mt-decap**

PIM is run only on the **mt-encap** interface. The **mt-decap** interface is used to populate downstream interface information.

Configuring BGP, MPLS, RSVP, and an IGP on the PE and Core Routers for Draft Rosen VPNs

To send multicast traffic across a Layer 3 VPN, you must configure network protocols to handle *intradomain routing* (an interior gateway protocol [IGP], such as Open Shortest Path First [OSPF] or Intermediate System-to-Intermediate System [IS-IS]), *interdomain routing* (Border Gateway Protocol [BGP]), *label switching* (Multiprotocol Label Switching [MPLS]), and *path signaling* (Resource Reservation Protocol [RSVP]). For more information about these protocols and examples of how to configure these protocols to support a Layer 3 VPN, see the *Junos VPNs Configuration Guide*.

NOTE: 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 does not work because PIM cannot transmit upstream interface information for multicast sources behind remote PE routers 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.

Creating a Unique Logical Loopback Interface for the Routing Instance for Draft Rosen VPNs

To facilitate the PIM protocol within a Layer 3 VPN, configure a unique loopback interface for the routing instance at the **[edit interfaces lo0 unit]** hierarchy level:

```
[edit interfaces]
lo0 {
   unit 1 {
      family inet {
         address  ip-address;
      }
   }
}
```

Configuring the Master PIM Instance on the PE Router in the Service Provider Network

To configure the master PIM instance that communicates with other PIM neighbors and the SP-RP within the service provider network, include the **pim** statement at the **[edit protocols]** hierarchy level. The example shown enables PIM sparse mode.

```
[edit protocols]
pim {
   rp {
      static {
         address ip-address;
      }
   }
   interface all {
```

```
        mode sparse;
        version 2;
    }
}
```

Configuring PIM and the VPN Group Address in a Routing Instance

The configuration syntax for PIM in a Layer 3 instance is available at the **[edit routing-instances protocols pim]** hierarchy level. It is similar to the global PIM configuration syntax found at the **[edit protocols pim]** hierarchy level.

In Junos OS Release 5.3 and later, you can include the **vpn-group-address** statement at the **[edit routing-instances *instance-name* protocols pim]** hierarchy level. You include this statement within the routing instance and specify the multicast group address for a particular VPN. Only one **vpn-group-address** statement can be configured per VPN, and this address should be unique on a per-VPN basis. To review how the VPN group address is used within GRE packet headers, see Stage 2 in "Dual PIM Multicast VPNs: Draft Rosen" on page 4.

Keep in mind that each PE router contains two entries of PIM: one for the master instance of PIM that connects through the service provider network and a second for the routing instance that connects to the CE router. The RP listed within the routing instance is the VPN C-RP, whereas the RP in the master PIM instance is an SP-RP. The following sample configuration shows a PE router with PIM enabled for sparse-dense mode in the VPN instance.

```
[edit]
routing-instances {
    instance-name {
        .....
        protocols {
            .....
            pim {
                vpn-group-address group-address;
                rp {
                    static {
                        address ip-address;
                    }
                }
                interface interface-name {
                    mode sparse-dense;
                    version 2;
                }
                interface lo0.1 {
                    mode sparse-dense;
                    version 2;
                }
            }
        }
    }
}
```

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

NOTE: In Junos OS Release 5.5 and later, you can configure PIM dense mode with the dense statement at the [edit routing-instances pim mode] hierarchy level. Sparse mode is available at this same hierarchy level in Junos OS Release 5.3 and later.

Option: Configuring PIM Sparse Mode Graceful Restart for a Layer 3 VPN

Graceful restart permits a routing platform to continue forwarding multicast traffic to neighbors while the routing protocol process restarts. To enable graceful restart for PIM sparse mode in a Layer 3 VPN, include the **graceful-restart** statement at both the **[edit routing-options]** and **[edit routing-instances *instance-name* routing-options]** hierarchy levels. To disable graceful restart in a Layer 3 VPN, include the **disable** statement at the **[edit routing-instances *instance-name* protocols pim graceful-restart]** hierarchy level.

```
[edit]
routing-options {
   graceful-restart {
      disable;
      restart-duration seconds;
   }
}
routing-instances {
   instance-name {
      .....
      protocols {
         pim {
            graceful-restart {
               disable;
               restart-duration seconds;
            }
         }
      }
      routing-options {
         graceful-restart {
            disable;
            restart-duration seconds;
         }
      }
   }
}
```

For more information about PIM sparse mode graceful restart in a Layer 3 VPN, see the *Junos OS High Availability Configuration Guide* or the *Junos Multicast Protocols Configuration Guide*.

Option: Configuring Multicast Distribution Trees for Data

By using date multicast distribution trees (MDTs) in a Layer 3 VPN, you can prevent multicast packets from being flooded unnecessarily to specified provider edge (PE) routers within a VPN group. This option is primarily useful for PE routers in your Layer 3 VPN multicast network that have no receivers for the multicast traffic from a particular source.

When a PE router directly connected to the multicast source receives Layer 3 VPN multicast traffic exceeding a configured threshold, a new data MDT tunnel is established between the PE router connected to the site where the multicast source is and its remote PE router neighbors. Neighbors that do not have receivers for the multicast traffic ignore the new tunnel. Conversely, neighbors that do have receivers for the multicast traffic link to the data MDT tunnel, which is created when the site exceeds a traffic rate threshold. If the multicast traffic level drops back below the threshold, the data MDT is torn down automatically and traffic flows back across the default Layer 3 VPN PIM tunnel.

To specify when the PE router directly connected to the multicast source should create a new data MDT, you must configure the maximum threshold value by including the **rate** statement at the **[edit routing-instances *instance-name* protocols pim mdt threshold group *group-address* source *source-address*]** hierarchy level. The data rate is specified in kilobits per second (Kbps). To specify the maximum number of data MDTs that can be created for a single routing instance, include the **tunnel-limit** statement at the **[edit routing-instances *instance-name* protocols pim mdt]** hierarchy level. To specify the multicast group IP address range used when a new data MDT needs to be initiated on the PE router, include the **group-range** statement at the **[edit routing-instances *instance-name* protocols pim mdt]** hierarchy level.

```
[edit routing-instances instance-name protocols pim]
mdt {
  group-range multicast-prefix;
  threshold {
    group group-address {
      source source-address {
        rate threshold-rate;
      }
    }
  }
  tunnel-limit limit;
}
```

NOTE: Because MDTs applies to VPNs and VRF routing instances, you cannot configure MDT statements in the master routing instance. If you configure MDTs in the master routing instance, the configuration commit operation will fail.

For more information about MDT, see the *Junos Multicast Protocols Configuration Guide*.

Option: Configuring MSDP Within a Layer 3 VPN

MSDP, defined in RFC 3618, allows a PIM-enabled network to connect multicast routing domains. It typically runs on the same router as a PIM sparse-mode rendezvous point (RP). Each MSDP router establishes adjacencies with internal and external MSDP peers similar to adjacency establishment for BGP peers. MSDP peer routers inform each other about active sources within the domain. When the peers detect active sources, they send explicit join messages to the active source.

You can configure MSDP in the master instance of a routing platform, or in the following types of routing instances:

- Forwarding
- No forwarding
- Virtual router
- VPLS
- VRF

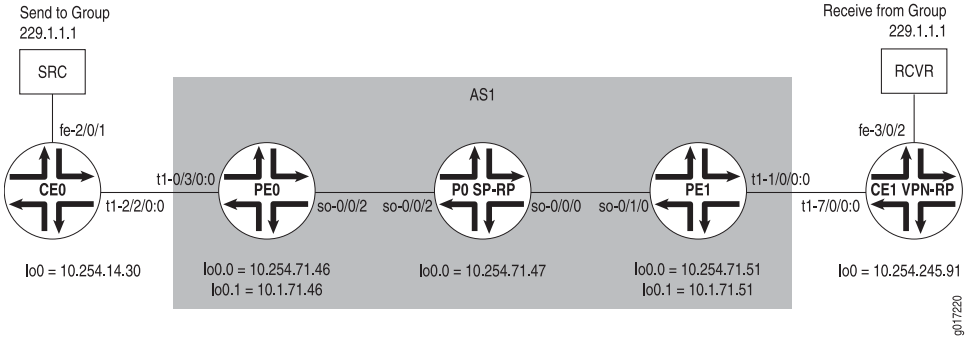
To configure MSDP in a Layer 3 VPN, include the **msdp** statement at the **[edit routing-instances *instance-name* protocols]** hierarchy level and specify local and peer addresses. You must also configure PIM sparse mode in the routing instance and specify a rendezvous point.

```
[edit routing-instances instance-name protocols]
pim {
   rp {
      local {
         address ip-address;
      }
   }
   interface interface-name;
}
msdp {
   local-address local-ip-address;
   peer peer-ip-address;
}
```

To view information about the operation of MSDP within a Layer 3 VPN instance, issue the **show msdp instance**, **show msdp statistics instance**, **show msdp source instance**, and **show msdp source-active instance** commands. For more information about MSDP, see the *Junos Multicast Protocols Configuration Guide*.

Example: Basic IPv4 Multicast over a Layer 3 VPN Configuration

Figure 5: Basic IPv4 Multicast over a Layer 3 VPN Topology Diagram



In Figure 5 on page 99, the multicast source sends to group **229.1.1.1**, and the receiver listens to the same group address. The VPN C-RP is located at Router CE1, whereas the SP-RP is located at Router P0. The routing instances are named VPN-A on both routers PE0 and PE1.

Router CE0

```
[edit]
protocols {
  pim {
    rp {
      dense-groups {
        229.0.0.0/8;
      }
      static {
        address 10.254.245.91;
      }
    }
    interface all {
      mode sparse-dense;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

In this example, the **interface all** statement is configured. If the topology requires only a few interfaces to be configured for PIM, then loopback interface **lo0** must also be one of the configured interfaces.

Router PE0

```
[edit]
protocols {
  pim {
    rp {
      static {
        address 10.254.71.47;
      }
    }
    interface all {
```

```
        mode sparse;
        version 2;
      }
      interface fxp0.0 {
        disable;
      }
    }
  }
```

Router PE0 also requires a standard VPN configuration, along with the PIM instance configuration. The **vpn-group-address** command is the only new PIM statement with PIM used exclusively with a routing instance multicast configuration.

Router PE0

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface t1-0/3/0:0.0;
    interface lo0.1
    route-distinguisher 10.254.71.46:100;
    vrf-import VPNA-import;
    vrf-export VPNA-export;
    protocols {
      ospf {
        export bgp-to-ospf;
        area 0.0.0.0 {
          interface t1-0/3/0:0.0;
          interface lo0.1;
        }
      }
      pim {
        dense-groups {
          229.0.0.0/8;
        }
        vpn-group-address 239.1.1.1;
        rp {
          static {
            address 10.254.245.91;
          }
        }
        interface t1-0/3/0:0.0 {
          mode sparse-dense;
          version 2;
        }
        interface lo0.1 {
          mode sparse-dense;
          version 2;
        }
      }
    }
  }
}
```

Router P0

```
[edit]
protocols {
  pim {
```

```
    rp {
      local {
        address 10.254.71.47;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Again, if the configuration calls for specific interfaces to be configured for PIM, loopback interface **lo0** must be included as one of the interfaces running PIM.

Router PE1

```
[edit]
protocols {
  pim {
    rp {
      static {
        address 10.254.71.47;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
routing-instances {
  VPN-A {
    instance-type vrf;
    interface t1-1/0/0:0.0;
    interface lo0.1;
    route-distinguisher 10.254.71.51:100;
    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 10.254.245.91;
              }
            }
            interface t1-1/0/0:0.0 {
              mode sparse-dense;
              version 2;
            }
            interface lo0.1 {
              mode sparse-dense;
              version 2;
            }
          }
        }
      }
    }
```

Router CE1

```
[edit]
protocols {
  pim {
    dense-groups {
      229.0.0.0/8;
    }
    rp {
      local {
        address 10.254.245.91;
      }
    }
    interface all {
      mode sparse-dense;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Verifying Your Work

To verify correct operation of basic IPv4 multicast over a Layer 3 VPN, use the following commands:

- **show pim**
- **show pim rps**
- **show pim rps instance** *instance-name*
- **show pim join**
- **show pim join extensive**
- **show pim join extensive instance** *instance-name*
- **show multicast route extensive**

- **show multicast next-hops**
- **show interfaces mt-*fpc/pic/port* extensive**

The following sections show the output of these commands used with the configuration example:

RP Information

You can view PIM information for the master instance with the **show pim** command. You can see information on the PIM routing instance with the **show pim (rps | join extensive) instance *instance-name*** command. Output verifying the SP-RP (**10.254.71.47**) as well as the VPN C-RP (**10.254.245.91**) follows.

```
user@PE0> show pim rps
Instance: PIM.master
Family: INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
10.254.71.47    static           0    None             1 224.0.0.0/4
Family: INET6
RP address      Type      Holdtime Timeout Active groups Group prefixes
user@PE0>  show pim rps instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP address      Type      Holdtime Timeout Active groups Group prefixes
10.254.245.91   static           0    None             0 224.0.0.0/4
Family: INET6
RP address      Type      Holdtime Timeout Active groups Group prefixes
```

PIM Information Before Multicast Transmission

With the configuration properly set, the backbone PIM sessions should be established before any traffic is forwarded. In the output below, the routers were configured, but the traffic source was not transmitting and the receiver was not requesting to be part of a group. Notice that there is no PIM join information for the routing instances yet.

Router PE0

```
user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 166
    Downstream Neighbors:
```

```
        Interface: so-0/0/2.0
            192.168.296.70 State: Join   Flags: S    Timeout: 204
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Join to Source
    Keepalive timeout: 166
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
user@PE0>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
```

Router P0

```
user@P0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.34 State: Join   Flags: SRW  Timeout: 186
        Interface: so-0/0/2.0
            192.168.296.69 State: Join   Flags: SRW  Timeout: 198
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 170
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.34 State: Join   Flags: S    Timeout: 186
        Interface: so-0/0/2.0             (pruned)
            192.168.296.69 State: Prune  Flags: SR   Timeout: 198
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/0/0.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 170
    Downstream Neighbors:
        Interface: so-0/0/0.0             (pruned)
            192.168.296.34 State: Prune  Flags: SR   Timeout: 186
        Interface: so-0/0/2.0
            192.168.296.69 State: Join   Flags: S    Timeout: 198
```

Router PE1

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/1/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
```

```
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt-pending
    Upstream interface: so-0/1/0.0
    Upstream State: Join to Source
    Keepalive timeout: 180
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 180
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.33 State: Join   Flags: S    Timeout: 168
```

Successful PIM Join Verification

In the remaining output for this example, the **show pim join** output shows group participation. Also displayed is the output from the **show multicast route extensive** and **show multicast next-hop** commands. The join output for PIM within a VPN will reference the group **229.1.1.1**, while the service provider side of the network will reference the join information for group **239.1.1.1** (which is the VPN group ID). In the **show multicast route extensive** output, you can view the group, sender, and upstream interface toward the sender.

Router CE0

```
user@CE0> show pim join
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: fe-2/0/1.0
Instance: PIM.master Family: INET6
```

```
user@CE0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  120   8010       0          360
    Upstream interface: fe-2/0/1.0
    Session name: Unknown
    Forwarding rate: 1 kbps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
```

```
user@CE0> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
120            2          1 t1-2/2/0:0.0
```

Router PE0

```
user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
```

```
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.46 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 188
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.70 State: Join   Flags: S    Timeout: 180
Instance: PIM.master Family: INET6


user@PE0> show interfaces mt-1/1/0 extensive
Physical interface: mt-1/1/0, Enabled, Physical link is Up
  Interface index: 37, SNMP ifIndex: 45, Generation: 36
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              2887970                    0 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                31896                    0 pps
   Output packets:                    0                    0 pps
  Logical interface mt-1/1/0.32769 (Index 43) (SNMP ifIndex 0) (Generation 46)
    Flags: Point-To-Point SNMP-Traps
    IP-Header 239.1.1.1:10.254.71.46:47:df:64:0000000800000000
    Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                    0
     Output bytes  :                 2396
     Input  packets:                    0
     Output packets:                   34
    Local statistics:
     Input  bytes  :                    0
     Output bytes  :                 2396
     Input  packets:                    0
     Output packets:                   34
    Transit statistics:
     Input  bytes  :                    0                    0 bps
     Output bytes  :                    0                    0 bps
     Input  packets:                    0                    0 pps
     Output packets:                    0                    0 pps
    Protocol inet, MTU: 4446, Generation: 79, Route table: 3
      Flags: None
  Logical interface mt-1/1/0.49154 (Index 44) (SNMP ifIndex 0) (Generation 47)
    Flags: Point-To-Point SNMP-Traps Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                 1550
     Output bytes  :                    0
     Input  packets:                   33
     Output packets:                    0
    Local statistics:
     Input  bytes  :                 1550
     Output bytes  :                    0
```

```
     Input  packets:                  33
     Output packets:                   0
    Transit statistics:
     Input  bytes  :                   0                    0 bps
     Output bytes  :                   0                    0 bps
     Input  packets:                   0                    0 pps
     Output packets:                   0                    0 pps
    Protocol inet, MTU: Unlimited, Generation: 80, Route table: 3
      Flags: None
```

```
user@PE0> show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-0/3/0:0.0
    Downstream interfaces:
        mt-1/1/0.32769
Instance: PIM.VPN-A Family: INET6
```

```
user@PE0> show pim join
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/2.0
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse
    Upstream interface: local
Instance: PIM.master Family: INET6
user@PE0>  show pim join instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-0/3/0:0.0
Instance: PIM.VPN-A Family: INET6
```

```
user@PE0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  86    9174       0          360
    Upstream interface: local
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  96    36         0          360
    Upstream interface: so-0/0/2.0
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
```

```
user@PE0> show multicast route extensive instance VPN-A
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  85    9408       0          360
    Upstream interface: t1-0/3/0:0.0
```

```
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
```

```
user@PE0> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
86             2          1 so-0/0/2.0
85             2          1 mt-1/1/0.32769
96             2          1 mt-1/1/0.49154
Family: INET6
```

Router P0

```
user@P0> show pim join
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: local
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt
    Upstream interface: so-0/0/2.0
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/0/0.0
Instance: PIM.master Family: INET6
```

```
user@P0> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  127   9906       195        360
    Upstream interface: so-0/0/2.0
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  126   135        23         359
    Upstream interface: so-0/0/0.0
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
```

```
user@P0> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
127            2          1 so-0/0/0.0
126            2          1 so-0/0/2.0
Family: INET6
```

Router PE1

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.47
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/1/0.0
```

```
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.51 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.46
    Flags: sparse,spt-pending
    Upstream interface: so-0/1/0.0
    Upstream State: Join to Source
    Keepalive timeout: 199
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            10.1.71.51 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 79
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.33 State: Join   Flags: S    Timeout: 174
        Interface: register to RP 10.254.71.47 on pe-1/1/0.32769
Instance: PIM.master Family: INET6


user@PE1>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: mt-1/1/0.32769
    Downstream interfaces:
        t1-1/0/0:0.0
Instance: PIM.VPN-A Family: INET6


user@PE1>  show interfaces mt-1/1/0 extensive
Physical interface: mt-1/1/0, Enabled, Physical link is Up
  Interface index: 38, SNMP ifIndex: 45, Generation: 37
  Type: Multicast-GRE, Link-level type: GRE, MTU: Unlimited, Speed: 800mbps
  Hold-times     : Up 0 ms, Down 0 ms
  Device flags   : Present Running
  Interface flags: SNMP-Traps
  Statistics last cleared: Never
  Traffic statistics:
   Input  bytes  :              2265256                 7568 bps
   Output bytes  :                    0                    0 bps
   Input  packets:                24981                   10 pps
   Output packets:                    0                    0 pps
  Logical interface mt-1/1/0.32769 (Index 45) (SNMP ifIndex 0) (Generation 46)
    Flags: Point-To-Point SNMP-Traps
    IP-Header 239.1.1.1:10.254.71.51:47:df:64:0000000800000000
    Encapsulation: GRE-NULL
    Traffic statistics:
     Input  bytes  :                    0
     Output bytes  :                10934
     Input  packets:                    0
     Output packets:                  153
    Local statistics:
     Input  bytes  :                    0
     Output bytes  :                10934
```

```
    Input  packets:                    0
    Output packets:                  153
   Transit statistics:
    Input  bytes  :                    0                    0 bps
    Output bytes  :                    0                    0 bps
    Input  packets:                    0                    0 pps
    Output packets:                    0                    0 pps
   Protocol inet, MTU: 4418, Generation: 77, Route table: 1
     Flags: None
  Logical interface mt-1/1/0.49154 (Index 46) (SNMP ifIndex 0) (Generation 47)
    Flags: Point-To-Point SNMP-Traps Encapsulation: GRE-NULL
    Traffic statistics:
    Input  bytes  :              1820512
    Output bytes  :                    0
    Input  packets:                19848
    Output packets:                    0
    Local statistics:
    Input  bytes  :                 5536
    Output bytes  :                    0
    Input  packets:                  120
    Output packets:                    0
    Transit statistics:
    Input  bytes  :              1814976                 7568 bps
    Output bytes  :                    0                    0 bps
    Input  packets:                19728                   10 pps
    Output packets:                    0                    0 pps
    Protocol inet, MTU: Unlimited, Generation: 78, Route table: 1
     Flags: None
```

```
user@PE1> show multicast route extensive
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
239.1.1.1       10.254.71.46   /32 A   F  76    11014      0          360
    Upstream interface: so-0/1/0.0
    Session name: Administratively Scoped
    Forwarding rate: 1 kBps (10 pps)
239.1.1.1       10.254.71.51   /32 A   F  103   1          0          360
    Upstream interface: local
    Session name: Administratively Scoped
    Forwarding rate: 0 kBps (0 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
```

```
user@PE1> show multicast route extensive instance VPN-A
Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  99    10976      4          360
    Upstream interface: mt-1/1/0.49154
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
```

```
user@PE1> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
75             2          1 so-0/1/0.0
99             2          1 t1-1/0/0:0.0
76             2          1 mt-1/1/0.49154
```

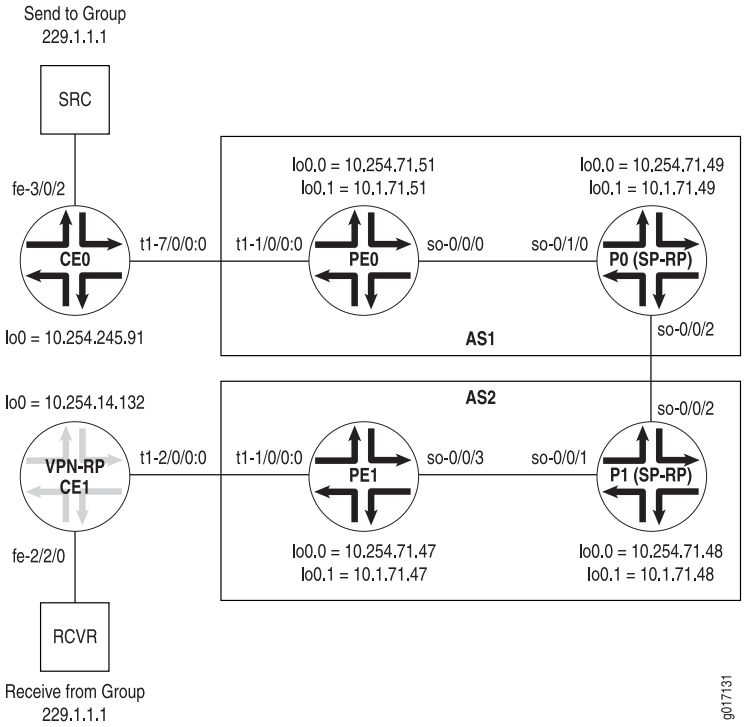
```
Family: INET6
```

Router CE1

```
user@CE1> show pim join
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.34
    Flags: dense
    Upstream interface: t1-7/0/0:0.0
Instance: PIM.master Family: INET6


user@CE1> show multicast route extensive
          2          1 fe-3/0/2.0

Family: INET
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout
229.1.1.1       192.168.295.34 /32 A   F  120   8010       0          360
    Upstream interface: t1-7/0/0:0.0
    Session name: Unknown
    Forwarding rate: 1 kBps (10 pps)
Family: INET6
Group           Source prefix     Act Pru NHid  Packets    IfMismatch Timeout


user@CE1> show multicast next-hops
Family: INET
ID      Refcount  KRefcount Downstream interface
120
```

Example: IPv4 Multicast with Interprovider VPNs Configuration

Figure 6: IPv4 Multicast with Interprovider VPNs Topology Diagram



Interprovider VPNs are also mentioned in RFC 4364. An example is shown in Figure 6 on page 112. The topology is slightly different; the main difference is the addition of MSDP between the two provider core transit (P) routers. In this limited topology, each P router is an SP-RP for the local autonomous system (AS), and Router CE1 is the VPN C-RP. VPN-A is the name of the routing instance on routers PE0 and PE1.

Router CE0

```
[edit]
protocols {
  pim {
    dense-groups {
      229.0.0.0/8;
    }
    rp {
      static {
        address 10.254.14.132;
      }
    }
    interface all {
      mode sparse-dense;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
```

```
}
```

Router PE0

```
[edit]
protocols {
  pim {
    rp {
      static {
        address 10.254.71.49;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
routing-instances {
  VPN-A {
    protocols {
      pim {
        dense-groups {
          229.0.0.0/8;
        }
        vpn-group-address 239.1.1.1;
        rp {
          static {
            address 10.254.14.132;
          }
        }
        interface t1-1/0/0:0.0 {
          mode sparse-dense;
          version 2;
        }
        interface lo0.1 {
          mode sparse-dense;
          version 2;
        }
      }
    }
  }
}
```

Router P0

```
[edit]
protocols {
  ...
  msdp {
    peer 10.254.71.48 {
      local-address 10.254.71.49;
    }
  }
  ...
  pim {
```

```
    rp {
      local {
        address 10.254.71.49;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Router P1

```
[edit]
protocols {
  ...
  msdp {
    peer 10.254.71.49 {
      local-address 10.254.71.48;
    }
  }
  ...
  pim {
    rp {
      local {
        address 10.254.71.48;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Router PE1

```
[edit]
protocols {
  pim {
    rp {
      static {
        address 10.254.71.48;
      }
    }
    interface all {
      mode sparse;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
```

```
    }
  }
  routing-instances {
    VPN-A {
      protocols {
        pim {
          dense-groups {
            229.0.0.0/8;
          }
          vpn-group-address 239.1.1.1;
          rp {
            static {
              address 10.254.14.132;
            }
          }
          interface t1-1/0/0:0.0 {
            mode sparse-dense;
            version 2;
          }
          interface lo0.1 {
            mode sparse-dense;
            version 2;
          }
        }
      }
    }
  }
```

Router CE1

```
[edit]
protocols {
  pim {
    dense-groups {
      229.0.0.0/8;
    }
    rp {
      local {
        address 10.254.14.132;
      }
    }
    interface all {
      mode sparse-dense;
      version 2;
    }
    interface fxp0.0 {
      disable;
    }
  }
}
```

Verifying Your Work

The **show** commands used to verify proper functionality of multicast in an interprovider environment are the same ones used with the first Layer 3 VPN multicast example (see “Verifying Your Work” on page 102).

The following output provides details for RP and the PIM join information:

Router CE0 Status

```
user@CE0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:21:35
Holdtime: 0
Device Index: 119
Subunit: 32769
Interface: pe-6/0/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6
```

```
user@CE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: fe-3/0/2.0
    Downstream interfaces:
        t1-7/0/0:0.0
Instance: PIM.master Family: INET6
```

Router PE0 Status

```
user@PE0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.49
Learned via: static configuration
Time Active: 00:22:07
Holdtime: 0
Device Index: 34
Subunit: 32769
Interface: pe-1/1/0.32769
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
```

```
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.51    10.254.71.51    10.254.71.49    Suppress      20
Family: INET6

user@PE0> show pim rps extensive instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:22:22
Holdtime: 0
Device Index: 34
Subunit: 32771
Interface: pe-1/1/0.32771
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6

user@PE0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.49
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/0.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/0.0
    Upstream State: Join to Source
    Keepalive timeout: 198
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 198
    Downstream Neighbors:
        Interface: so-0/0/0.0
            192.168.296.42 State: Join   Flags: S    Timeout: 176
Instance: PIM.master Family: INET6

user@PE0> show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: t1-1/0/0:0.0
    Downstream interfaces:
```

```
        mt-1/1/0.32769
Instance: PIM.VPN-A Family: INET6
```

Router P0 Status

```
user@P0> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.49
Learned via: static configuration
Time Active: 00:30:43
Holdtime: 0
Device Index: 33
Subunit: 32768
Interface: pd-1/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.51    10.254.71.51    10.254.71.49    Receive
Family: INET6


user@P0> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.49
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.41 State: Join   Flags: SRW  Timeout: 184
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 207
    Downstream Neighbors:
        Interface: so-0/1/0.0
            192.168.296.41 State: Join   Flags: S    Timeout: 184
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt
    Upstream interface: so-0/1/0.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 207
    Downstream Neighbors:
        Interface: so-0/0/2.0
            192.168.296.73 State: Join   Flags: S    Timeout: 186
        Interface: so-0/1/0.0             (pruned)
            192.168.296.41 State: Prune  Flags: SR   Timeout: 184
Instance: PIM.master Family: INET6
```

Router P1 Status

```
user@P1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.71.48
Learned via: static configuration
Time Active: 06:26:56
Holdtime: 0
Device Index: 32
Subunit: 32768
Interface: pd-1/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.47    10.254.71.47    10.254.71.48    Receive        0
Family: INET6
```

```
user@P1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.48
    Flags: sparse,rptree,wildcard
    Upstream interface: local
    Upstream State: Local RP
    Downstream Neighbors:
        Interface: so-0/0/1.0
            192.168.296.50 State: Join   Flags: SRW  Timeout: 174
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse,spt
    Upstream interface: so-0/0/1.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 196
    Downstream Neighbors:
        Interface: so-0/0/1.0             (pruned)
            192.168.296.50 State: Prune  Flags: SR   Timeout: 174
        Interface: so-0/0/2.0
            192.168.296.74 State: Join   Flags: S    Timeout: 178
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/2.0
    Upstream State: Local RP, Join to Source
    Keepalive timeout: 196
    Downstream Neighbors:
        Interface: so-0/0/1.0
            192.168.296.50 State: Join   Flags: S    Timeout: 174
Instance: PIM.master Family: INET6
```

Router PE1 Status

```
user@PE1> show pim rps extensive
```

```
Instance: PIM.master
Family: INET
RP: 10.254.71.48
Learned via: static configuration
Time Active: 00:25:13
Holdtime: 0
Device Index: 34
Subunit: 32770
Interface: pe-1/1/0.32770
Group Ranges:
        224.0.0.0/4
Active groups using RP:
        239.1.1.1
        total 1 groups active
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
239.1.1.1       10.254.71.47    10.254.71.47    10.254.71.48    Suppress      42
Family: INET6
```

```
user@PE1> show pim rps extensive instance VPN-A
Instance: PIM.VPN-A
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:25:17
Holdtime: 0
Device Index: 34
Subunit: 32771
Interface: pe-1/1/0.32771
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6
```

```
user@PE1> show pim join extensive
Instance: PIM.master Family: INET
Group: 239.1.1.1
    Source: *
    RP: 10.254.71.48
    Flags: sparse,rptree,wildcard
    Upstream interface: so-0/0/3.0
    Upstream State: Join to RP
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: SRW  Timeout: Infinity
Group: 239.1.1.1
    Source: 10.254.71.47
    Flags: sparse
    Upstream interface: local
    Upstream State: Local Source, Prune to RP
    Keepalive timeout: 173
    Downstream Neighbors:
        Interface: so-0/0/3.0
            192.168.296.49 State: Join   Flags: S    Timeout: 199
Group: 239.1.1.1
    Source: 10.254.71.51
    Flags: sparse,spt-pending
    Upstream interface: so-0/0/3.0
```

```
    Upstream State: Join to Source
    Keepalive timeout: 173
    Downstream Neighbors:
        Interface: mt-1/1/0.32769
            0.0.0.0 State: Join   Flags: S    Timeout: Infinity
Instance: PIM.master Family: INET6
```

```
user@PE1>  show pim join extensive instance VPN-A
Instance: PIM.VPN-A Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: mt-1/1/0.32769
    Downstream interfaces:
        t1-1/0/0:0.0
Instance: PIM.VPN-A Family: INET6
```

Router CE1 Status

```
user@CE1> show pim rps extensive
Instance: PIM.master
Family: INET
RP: 10.254.14.132
Learned via: static configuration
Time Active: 00:28:22
Holdtime: 0
Device Index: 69
Subunit: 32768
Interface: pd-3/1/0.32768
Group Ranges:
        224.0.0.0/4
Active groups using RP:
Register State for RP:
Group           Source          FirstHop        RP Address      State    Timeout
Family: INET6
```

```
user@CE1>  show pim join extensive
Instance: PIM.master Family: INET
Group: 229.1.1.1
    Source: 192.168.295.38
    Flags: dense
    Upstream interface: t1-2/0/0:0.0
    Downstream interfaces:
        fe-2/2/0.0
Instance: PIM.master Family: INET6
```

For More Information

For additional information about multicast over Layer 3 VPNs, see the following resources:

- *Junos OS Multicast Protocols Configuration Guide*
- *Junos OS VPNs Configuration Guide*
- *RFC 2547, BGP/MPLS VPNs*
- *RFC 3618, Multicast Source Discovery Protocol (MSDP)*

- *RFC 4364, BGP/MPLS IP Virtual Private Networks (VPNs)*
- Internet draft draft-rosen-vpn-mcast-06.txt, *Multicast in MPLS/BGP VPNs* (expires April 2004)

PART 2

Index

Index

