JUNIPER NETWORKS

Network Configuration Example

Configuring MBGP Multicast VPN Extranets

Release

11.1

Published: 2011-01-19

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

Network Configuration Example Configuring MBGP MVPN Extranets
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

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 5
- MBGP Multicast VPN Extranets Configuration Guidelines on page 3

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 5
- MBGP Multicast VPN Extranets Overview on page 1

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:

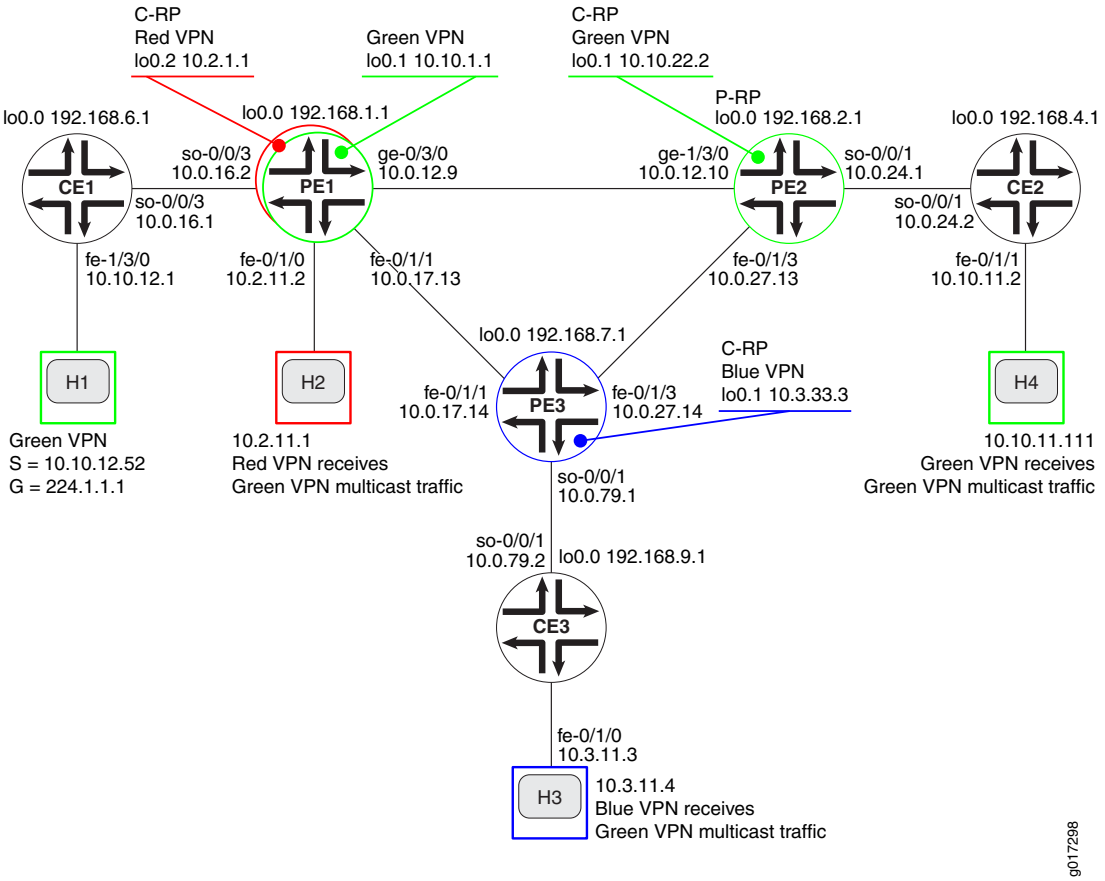
- Junos OS Release 9.5 or later
- Six M Series, T Series, TX Series, or MX Series Juniper routers
- One adaptive services PIC or MultiServices PIC in each of the M Series or T Series routers acting as PE routers
- One host system capable of sending multicast traffic and supporting the Internet Group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

Overview and Topology

In the network topology shown in Figure 1 on page 6:

- 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 1: 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 3
- MBGP Multicast VPN Extranets Overview on page 1