

Network Configuration Example

Interconnecting a Layer 2 VPN with a Layer 2 VPN

Release
11.2



Published: 2011-05-17

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

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Network Configuration Example Interconnecting a Layer 2 VPN with a Layer 2 VPN

Release 11.2

Copyright © 2011, Juniper Networks, Inc.

All rights reserved.

Revision History

April 2011—R1 Junos OS 11.2

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 présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés 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

Introduction	1
Layer 2 VPN Overview	3
Layer 2 VPN Applications	5
Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN	7
Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN	9

Introduction

This configuration example provides configuration and verification commands for interconnecting a Layer 2 VPN with a Layer 2 VPN using the Layer 2 interworking (iw0) interface. The Junos OS can make use of a Tunnel Services PIC to loop packets out and back from the Packet Forwarding Engine (PFE), to link together Layer 2 networks. The Layer 2 interworking interface avoids the need for the Tunnel Services PIC and overcomes the limitation of bandwidth constraints imposed by the Tunnel Services PIC.

Layer 2 VPN Overview

As the need to link different Layer 2 services to one another for expanded service offerings grows, Layer 2 MPLS VPN services are increasingly in demand.

Implementing a Layer 2 VPN on a router is similar to implementing a VPN using a Layer 2 technology, such as Asynchronous Transfer Mode (ATM). However, for a Layer 2 VPN on a router, traffic is forwarded to the router in a Layer 2 format. It is carried by Multiprotocol Label Switching (MPLS) over the service provider's network, and then converted back to Layer 2 format at the receiving site. You can configure different Layer 2 formats at the sending and receiving sites. The security and privacy of an MPLS Layer 2 VPN are equal to those of an ATM or Frame Relay VPN. The service provisioned with Layer 2 VPNs is also known as Virtual Private Wire Service (VPWS).

On a Layer 2 VPN, routing typically occurs on the customer edge (CE) router. The CE router connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The provider edge (PE) router receiving the traffic sends the traffic across the service provider's network to the PE router connected to the receiving site. The PE routers do not need to store or process the customer's routes; they only need to be configured to send data to the appropriate tunnel. For a Layer 2 VPN, customers need to configure their own routers to carry all Layer 3 traffic. The service provider needs to know only how much traffic the Layer 2 VPN will need to carry. The service provider's routers carry traffic between the customer's sites using Layer 2 VPN interfaces. The VPN topology is determined by policies configured on the PE routers.

Because Layer 2 VPNs use BGP as the signaling protocol, they have a simpler design and require less overhead than traditional VPNs over Layer 2 circuits. BGP signaling also enables autodiscovery of Layer 2 VPN peers. Layer 2 VPNs are similar to BGP or MPLS VPNs and VPLS in many respects; all three types of services employ BGP for signaling.

Related Documentation

- Layer 2 VPN Applications on page 5
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN on page 7
- Interconnecting Layer 2 VPNs with Layer 3 VPNs Overview
- Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN on page 9
- Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN
- Example: Interconnecting a Layer 2 VPN with a Layer 3 VPN

Layer 2 VPN Applications

Implementing a Layer 2 VPN includes the following benefits:

- Terminating a Layer 2 VPN into a Layer 2 VPN using the interworking (iw0) software interface eliminates the limitation of bandwidth on the tunnel interfaces used for these configuration scenarios. Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 VPN to another Layer 2 VPN, Junos OS is used to link both the Layer 2 VPN routes.
- Layer 2 VPNs enable the sharing of a provider's core network infrastructure between IP and Layer 2 VPN services, reducing the cost of providing those services. A Layer 2 MPLS VPN allows you to provide Layer 2 VPN service over an existing IP and MPLS backbone.
- From a service provider's point of view, a Layer 2 MPLS VPN allows the use of a single Layer 3 VPN (such as RFC 2547bis), MPLS traffic engineering, and Differentiated Services (DiffServ).
- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 VPN service. You can configure the PE router to run any Layer 3 protocol in addition to the Layer 2 protocols. Customers who prefer to maintain control over most of the administration of their own networks might want Layer 2 VPN connections with their service provider instead of a Layer 3 VPN.

Related Documentation

- [Layer 2 VPN Overview on page 3](#)
- [Using the Layer 2 Interworking Interface to Interconnect a Layer 2 Circuit to a Layer 2 VPN](#)
- [Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN on page 7](#)
- [Example: Interconnecting a Layer 2 Circuit with a Layer 2 VPN](#)
- [Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN on page 9](#)
- [Example: Interconnecting a Layer 2 VPN with a Layer 3 VPN](#)

Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN

Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 VPN to another Layer 2 VPN, the Layer 2 Interworking interface uses Junos OS to stitch together both Layer 2 VPN routes.

To configure the interworking interface, include the **iw0** statement. The **iw0** statement is configured at the **[edit interfaces]** hierarchy level.

```
[edit interfaces]
iw0 {
  unit 0 {
    peer 1;
  }
  unit 1 {
    peer 0;
  }
}
```

The configuration of an interworking (iw) interface is similar to the configuration of a logical tunnel (lt) interface. In this example, the logical interfaces must be associated with the endpoints of both Layer 2 VPN connections terminating on this router. To make the association, include the **interfaces** statement and specify **iw0** as the interface name. Include the statement at the **[edit routing-instances *routing-instances-name* protocols l2vpn site *site-name*]** hierarchy level for each routing instance. The **routing-instances** statement is configured at the **[edit routing-instances]** hierarchy level.

```
[edit routing-instances]
L2VPN-PE1 {
  instance-type l2vpn;
  interface iw0.0;
  route-distinguisher 65000:3;
  vrf-target target:65000:2;
  protocols {
    l2vpn {
      encapsulation-type ethernet;
      site CE3 {
        site-identifier 3;
        interface iw0.0 {
          remote-site-id 1;
        }
      }
    }
  }
}
L2VPN-PE5 {
  instance-type l2vpn;
  interface iw0.1;
  route-distinguisher 65000:33;
  vrf-target target:65000:2;
  protocols {
    l2vpn {
      encapsulation-type ethernet;
      site CE3 {
```

```
        site-identifier 3;  
        interface iw0.1 {  
            remote-site-id 5;  
        }  
    }  
}
```

In addition to the **iw0** interface configuration, Layer 2 interworking **l2iw** protocols need to be configured. Without the **l2iw** configuration, the **l2iw** routes will not be formed, regardless of whether any **iw** interfaces are present. Within the **l2iw** protocols, only trace options can be configured in the standard fashion. The minimum configuration necessary for the feature to work is shown below:

```
[edit]  
protocols {  
    l2iw;  
}
```

**Related
Documentation**

- [Layer 2 VPN Overview on page 3](#)
- [Layer 2 VPN Applications on page 5](#)
- [Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN on page 9](#)

Example: Interconnecting a Layer 2 VPN with a Layer 2 VPN

This example provides a step-by-step procedure for interconnecting and verifying a Layer 2 VPN with a Layer 2 VPN. It contains the following sections:

- Requirements on page 9
- Overview and Topology on page 9
- Configuration on page 10

Requirements

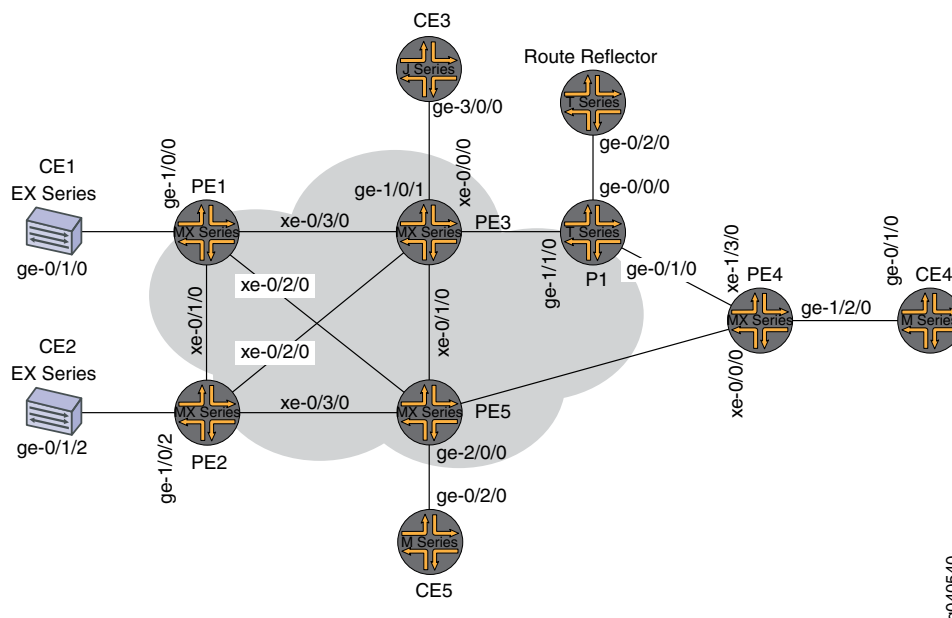
This example uses the following hardware and software components:

- Junos OS Release 9.3 or later
- 2 MX Series routers
- 2 M Series routers
- 1 T Series router
- 1 EX Series router
- 1 J Series router

Overview and Topology

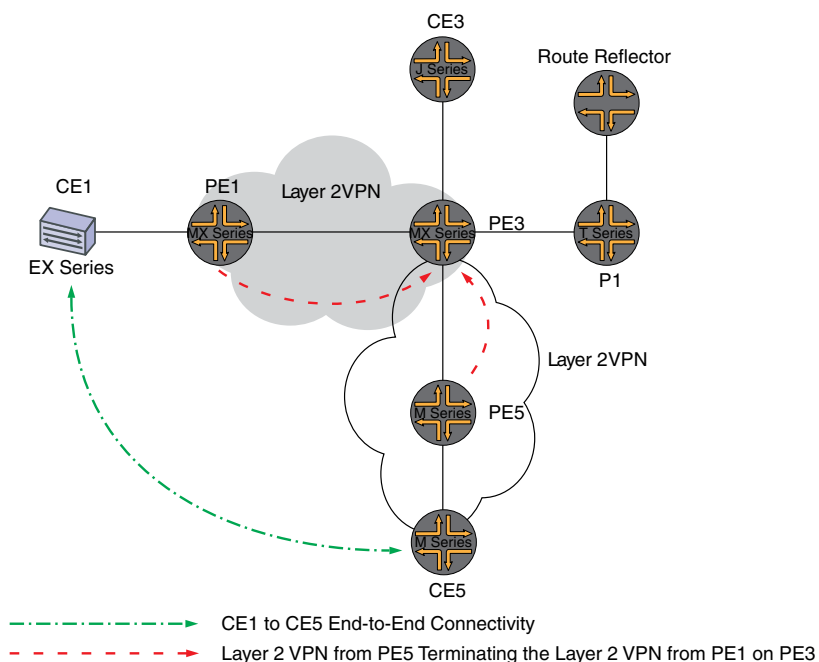
The physical topology of the Layer 2 VPN to Layer 2 VPN connection example is shown in Figure 1 on page 9.

Figure 1: Physical Topology of a Layer 2 VPN to Layer 2 VPN Connection



The logical topology of a Layer 2 VPN to Layer 2 VPN connection is shown in Figure 2 on page 10.

Figure 2: Logical Topology of a Layer 2 VPN to Layer 2 VPN Connection



9040542

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
- **CE3** identifies the customer edge 3 (CE3) router
- **PE3** identifies the provider edge 3 (PE3) router
- **CE5** identifies the customer edge 5 (CE5) router
- **PE5** identifies the provider edge 5 (PE5) router

This example is organized in the following sections:

- Configuring Protocols on the PE and P Routers on page 11
- Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3 on page 16
- Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3 on page 18

Configuring Protocols on the PE and P Routers

Step-by-Step Procedure All of the PE routers and P routers are configured with OSPF as the IGP protocol. The MPLS, LDP, and BGP protocols are enabled on all of the interfaces except **fxp0.0**. Core-facing interfaces are enabled with the MPLS address and inet address.

1. Configure all the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp0.0**. The following configuration snippet shows the protocol configuration for Router PE1:

```
[edit]
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  bgp {
    group RR {
      type internal;
      local-address 1.1.1.1;
      family l2vpn {
        signaling;
      }
      neighbor 7.7.7.7;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface all;
      interface fxp0.0 {
        disable;
      }
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
```

2. Configure the PE and P routers with OSPF as the IGP. Enable the MPLS, LDP, and BGP protocols on all interfaces except **fxp0.0**. The following configuration snippet shows the protocol configuration for Router PE3:

```
[edit]
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
```

```
}
bgp {
  group RR {
    type internal;
    local-address 3.3.3.3;
    family l2vpn {
      signaling;
    }
    neighbor 7.7.7.7;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
ldp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
}
```

Step-by-Step Procedure

Configuring the Layer 2 VPN Protocol and Interfaces

1. On Router PE1, configure the **ge-1/0/0** interface encapsulation. To configure the interface encapsulation, include the **encapsulation** statement and specify the **ethernet-ccc** option (vlan-ccc encapsulation is also supported). Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality. To configure the logical interface family, include the **family** statement and specify the **ccc** option. The encapsulation should be configured the same way for all routers in the Layer 2 VPN domain.

```
[edit interfaces]
ge-1/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
```

2. On Router PE1, configure the Layer 2 VPN protocols. Configure the remote site ID as 3. Site ID 3 represents Router PE3 (Hub-PE). To configure the Layer 2 VPN

protocols, include the **l2vpn** statement at the **[edit routing-instances routing-instances-name protocols]** hierarchy level. Layer 2 VPNs use BGP as the signaling protocol.

```
[edit routing-instances]
L2VPN {
  instance-type l2vpn;
  interface ge-1/0/0.0;
  route-distinguisher 65000:1;
  vrf-target target:65000:2;
  protocols {
    l2vpn {
      encapsulation-type ethernet;
      site CE1 {
        site-identifier 1;
        interface ge-1/0/0.0 {
          remote-site-id 3;
        }
      }
    }
  }
}
```

3. On Router PE5, configure the **ge-2/0/0** interface encapsulation by including the **encapsulation** statement and specify the **ethernet-ccc** option. Configure the **ge-1/0/0.0** logical interface family for circuit cross-connect functionality by including the **family** statement and specifying the **ccc** option.

```
[edit interfaces]
ge-2/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 5.5.5.5/32;
    }
  }
}
```

4. On Router PE5, configure the Layer 2 VPN protocols by including the **l2vpn** statement at the **[edit routing-instances routing-instances-name protocols]** hierarchy level. Configure the remote site ID as **3**.

```
[edit routing-instances]
L2VPN {
  instance-type l2vpn;
  interface ge-2/0/0.0;
  route-distinguisher 65000:5;
  vrf-target target:65000:2;
  protocols {
    l2vpn {
      encapsulation-type ethernet;
```

```
site CE5 {  
  site-identifier 5;  
  interface ge-2/0/0.0 {  
    remote-site-id 3;  
  }  
}  
}
```

5. On Router PE3, configure the **iw0** interface with two logical interfaces. To configure the **iw0** interface, include the **interfaces** statement and specify **iw0** as the interface name. For the unit 0 logical interface, include the **peer-unit** statement and specify the logical interface **unit 1** as the peer l interface. For the unit 1 logical interface, include the **peer-unit** statement and specify the logical interface **unit 0** as the peer interface.

```
[edit interfaces]  
iw0 {  
  unit 0 {  
    encapsulation ethernet-ccc;  
    peer-unit 1;  
  }  
  unit 1 {  
    encapsulation ethernet-ccc;  
    peer-unit 0;  
  }  
}
```

6. On Router PE3, configure the edge-facing **ge-1/0/1** interface encapsulation by including the **encapsulation** statement and specifying the **ethernet-ccc** option.

```
[edit interfaces]  
ge-1/0/1 {  
  encapsulation ethernet-ccc;  
  unit 0 {  
    family ccc;  
  }  
}
```

7. On Router PE3, configure the logical loopback interface. The loopback interface is used to establish the targeted LDP sessions to Routers PE1 and PE5.

```
[edit interfaces]  
lo0 {  
  unit 0 {  
    family inet {  
      address 3.3.3.3/32;  
    }  
  }  
}
```

8. On Router PE3, enable the Layer 2 interworking protocol. To enable the Layer 2 interworking protocol, include the **l2iw** statement at the **[edit protocols]** hierarchy level.

```
[edit protocols]
```

```
l2iw;
```

9. On Router PE3, configure two Layer 2 VPN routing instances to terminate the Layer 2 VPN virtual circuits from Routers PE1 and PE5, as shown.

```
[edit routing-instances]
L2VPN-PE1 {
  instance-type l2vpn;
  interface iw0.0;
  route-distinguisher 65000:3;
  vrf-target target:65000:2;
  protocols {
    l2vpn {
      encapsulation-type ethernet;
      site CE3 {
        site-identifier 3;
        interface iw0.0 {
          remote-site-id 1;
        }
      }
    }
  }
}
L2VPN-PE5 {
  instance-type l2vpn;
  interface iw0.1;
  route-distinguisher 65000:33;
  vrf-target target:65000:2;
  protocols {
    l2vpn {
      encapsulation-type ethernet;
      site CE3 {
        site-identifier 3;
        interface iw0.1 {
          remote-site-id 5;
        }
      }
    }
  }
}
```

Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3

Step-by-Step Procedure

1. BGP is used for control plane signaling in a Layer 2 VPN. On Router PE1, use the **show bgp** command to verify that the BGP control plane for the Layer 2 VPN, has established a neighbor relationship with the router reflector that has IP address **7.7.7.7**.

Three Layer 2 VPN routes are received from the route reflector from each PE router in the topology.

```
user@PE1> show bgp summary
```

```
Groups: 1 Peers: 1 Down peers: 0
Table      Tot Paths  Act Paths Suppressed  History  Damp State   Pending
bgp.l2vpn.0      3          3          0          0          0          0
Peer           AS      InPkt    OutPkt    OutQ   Flaps  Last Up/Dwn
State|#Active/Received/Accepted/Damped...
7.7.7.7         65000      190      192       0       0    1:24:40 Establ
  bgp.l2vpn.0: 3/3/3/0
  L2VPN.l2vpn.0: 3/3/3/0
```

2. On Router PE1, use the **show route** command to verify that the BGP Layer 2 VPN routes are stored in the **L2VPN.l2vpn.0** routing table for each PE router.

```
user@PE1> show route table L2VPN.l2vpn.0
```

```
L2VPN.l2vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
65000:1:1:3/96
    *[L2VPN/170/-101] 01:31:53, metric2 1
    Indirect
65000:3:3:1/96
    *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
    AS path: I
    > to 10.10.1.2 via xe-0/3/0.0
65000:5:5:3/96
    *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
    AS path: I
    > to 10.10.3.2 via xe-0/2/0.0
65000:33:3:5/96
    *[BGP/170] 01:24:58, localpref 100, from 7.7.7.7
    AS path: I
    > to 10.10.1.2 via xe-0/3/0.0
```

3. On Router PE1, use the **show ldp session** command to verify that targeted LDP sessions are established to the PE routers in the network and that the state is **Operational**.

```
user@PE1> show ldp session
```

Address	State	Connection	Hold time
2.2.2.2	Operational	Open	24
3.3.3.3	Operational	Open	22
5.5.5.5	Operational	Open	28

4. On Router PE1, use the **show l2vpn connections** command to verify that the Layer 2 VPN to site 3 on Router PE3 (Hub-PE) is **Up**.

```
user@PE1> show l2vpn connections
```


Layer-2 VPN connections:

Legend for connection status (St)

EI -- encapsulation invalid	NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch	WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down	NP -- interface hardware not present
CM -- control-word mismatch	-> -- only outbound connection is up
CN -- circuit not provisioned	<- -- only inbound connection is up
OR -- out of range	Up -- operational
OL -- no outgoing label	Dn -- down
LD -- local site signaled down	CF -- call admission control failure
RD -- remote site signaled down	SC -- local and remote site ID collision
LN -- local site not designated	LM -- local site ID not minimum designated
RN -- remote site not designated	RM -- remote site ID not minimum designated
XX -- unknown connection status	IL -- no incoming label
MM -- MTU mismatch	MI -- Mesh-Group ID not available
BK -- Backup connection	ST -- Standby connection
PF -- Profile parse failure	PB -- Profile busy

Legend for interface status

Up -- operational
Dn -- down

Instance: L2VPN

Local site: CE1 (1)

connection-site	Type	St	Time last up	# Up trans
3	rmt	Up	Jan 5 18:08:25 2010	1
Remote PE: 3.3.3.3, Negotiated control-word: Yes (Null)				
Incoming label: 800000, Outgoing label: 800000				
Local interface: ge-1/0/0.0, Status: Up, Encapsulation: ETHERNET				
5	rmt	OR		

- On Router PE1, use the **show route** command to verify that the **mpls.0** routing table is populated with the Layer 2 VPN routes used to forward the traffic using an LDP label. Notice that in this example, the router is pushing label **8000000**.

```
user@PE1> show route table mpls.0
```

```
[edit]
```

```
mpls.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0          *[MPLS/0] 1w1d 11:36:44, metric 1
           Receive
1          *[MPLS/0] 1w1d 11:36:44, metric 1
           Receive
2          *[MPLS/0] 1w1d 11:36:44, metric 1
           Receive
300432     *[LDP/9] 3d 04:25:02, metric 1
           > to 10.10.2.2 via xe-0/1/0.0, Pop
300432(S=0) *[LDP/9] 3d 04:25:02, metric 1
           > to 10.10.2.2 via xe-0/1/0.0, Pop
300768     *[LDP/9] 3d 04:25:02, metric 1
           > to 10.10.3.2 via xe-0/2/0.0, Pop
300768(S=0) *[LDP/9] 3d 04:25:02, metric 1
           > to 10.10.3.2 via xe-0/2/0.0, Pop
300912     *[LDP/9] 3d 04:25:02, metric 1
           > to 10.10.3.2 via xe-0/2/0.0, Swap 299856
301264     *[LDP/9] 3d 04:24:58, metric 1
```

```

> to 10.10.1.2 via xe-0/3/0.0, Swap 308224
301312      *[LDP/9] 3d 04:25:01, metric 1
> to 10.10.1.2 via xe-0/3/0.0, Pop
301312(S=0) *[LDP/9] 3d 04:25:01, metric 1
> to 10.10.1.2 via xe-0/3/0.0, Pop
800000      *[L2VPN/7] 01:25:28
> via ge-1/0/0.0, Pop Offset: 4
ge-1/0/0.0  *[L2VPN/7] 01:25:28, metric 21
> to 10.10.1.2 via xe-0/3/0.0, Push 800000 Offset: -4

```

Verifying the Layer 2 VPN to Layer 2 VPN Connection on Router PE3

- Step-by-Step Procedure**
1. On Router PE3, use the **show l2vpn connections** command to verify that the Layer 2 VPN connections from Router PE1 and Router PE5 are **Up** and are using the **iw0** interface.

```
user@PE3> show l2vpn connections
```

```
Instance: L2VPN-PE1
Local site: CE3 (3)
```

connection-site	Type	St	Time last up	# Up trans
1	rmt	Up	Jan 5 18:08:22 2010	1

Remote PE: 1.1.1.1, Negotiated control-word: Yes (Null)
Incoming label: 800000, Outgoing label: 800000
Local interface: iw0.0, Status: Up, Encapsulation: ETHERNET

connection-site	Type	St	Time last up	# Up trans
5	rmt	OR		

```
Instance: L2VPN-PE5
Local site: CE3 (3)
```

connection-site	Type	St	Time last up	# Up trans
1	rmt	CN		
5	rmt	Up	Jan 5 18:08:22 2010	1

Remote PE: 5.5.5.5, Negotiated control-word: Yes (Null)
Incoming label: 800002, Outgoing label: 800000
Local interface: iw0.1, Status: Up, Encapsulation: ETHERNET

2. On Router PE3, use the **show ldp neighbor** command to verify that the targeted LDP session neighbor IP addresses are shown.

```
user@PE3> show ldp neighbor
```

Address	Interface	Label space ID	Hold time
1.1.1.1	lo0.0	1.1.1.1:0	44
2.2.2.2	lo0.0	2.2.2.2:0	42
4.4.4.4	lo0.0	4.4.4.4:0	31
5.5.5.5	lo0.0	5.5.5.5:0	44

3. On Router PE3, use the **show bgp summary** command to verify that the BGP control plane for the Layer 2 VPN, has established a neighbor relationship with the router reflector that has IP address 7.7.7.7.

```
user@PE3> show bgp summary
```

```
Groups: 1 Peers: 1 Down peers: 0
```

Table	Tot Paths	Act Paths	Suppressed	History	Damp	State	Pending
bgp.l2vpn.0	2	2	0	0	0	0	0

Peer AS InPkt OutPkt OutQ Flaps Last Up/Dwn

State|#Active/Received/Accepted/Damped...

```
7.7.7.7 65000 10092 10195 0 0 3d 4:23:27 Establ
```

bgp.l2vpn.0: 2/2/2/0

L2VPN-PE1.l2vpn.0: 2/2/2/0
 L2VPN-PE5.l2vpn.0: 2/2/2/0

4. On Router PE3, use the **show ldp session** command to verify that targeted LDP sessions are established to all of the PE routers in the network and that the state is **Operational**.

```
user@PE3> show ldp session
```

Address	State	Connection	Hold time
1.1.1.1	Operational	Open	24
2.2.2.2	Operational	Open	22
4.4.4.4	Operational	Open	20
5.5.5.5	Operational	Open	24

5. On Router PE3, use the **show route** command to verify that the **mpls.0** routing table is populated with the Layer 2 VPN routes used to forward the traffic using an LDP label. Notice that in this example, the router is swapping label **800000**. Also notice the two **iw0** interfaces that are used for the Layer 2 interworking routes.

```
user@PE3>show route table mpls.0
```

```
mpls.0: 16 destinations, 18 routes (16 active, 2 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
0          *[MPLS/0] 1w1d 11:50:14, metric 1
            Receive
1          *[MPLS/0] 1w1d 11:50:14, metric 1
            Receive
2          *[MPLS/0] 1w1d 11:50:14, metric 1
            Receive
308160     *[LDP/9] 3d 04:38:45, metric 1
            > to 10.10.1.1 via xe-0/3/0.0, Pop
308160(S=0) *[LDP/9] 3d 04:38:45, metric 1
            > to 10.10.1.1 via xe-0/3/0.0, Pop
308176     *[LDP/9] 3d 04:38:44, metric 1
            > to 10.10.6.2 via xe-0/1/0.0, Pop
308176(S=0) *[LDP/9] 3d 04:38:44, metric 1
            > to 10.10.6.2 via xe-0/1/0.0, Pop
308192     *[LDP/9] 00:07:18, metric 1
            > to 10.10.20.1 via xe-0/0/0.0, Swap 601649
            > to 10.10.6.2 via xe-0/1/0.0, Swap 299856
308208     *[LDP/9] 3d 04:38:44, metric 1
            > to 10.10.5.1 via xe-0/2/0.0, Pop
308208(S=0) *[LDP/9] 3d 04:38:44, metric 1
            > to 10.10.5.1 via xe-0/2/0.0, Pop
308224     *[LDP/9] 3d 04:38:42, metric 1
            > to 10.10.20.1 via xe-0/0/0.0, Pop
308224(S=0) *[LDP/9] 3d 04:38:42, metric 1
            > to 10.10.20.1 via xe-0/0/0.0, Pop
800000     *[L2IW/6] 01:39:13, metric2 1
            > to 10.10.6.2 via xe-0/1/0.0, Swap 800000
            [L2VPN/7] 01:39:13
            > via iw0.0, Pop   Offset: 4
800002     *[L2IW/6] 01:39:13, metric2 1
            > to 10.10.1.1 via xe-0/3/0.0, Swap 800000
            [L2VPN/7] 01:39:13
            > via iw0.1, Pop   Offset: 4
iw0.0     *[L2VPN/7] 01:39:13, metric2 1
            > to 10.10.1.1 via xe-0/3/0.0, Push 800000 Offset: -4
```

```
iw0.1      *[L2VPN/7] 01:39:13, metric2 1
> to 10.10.6.2 via xe-0/1/0.0, Push 800000 Offset: -4
```

**Step-by-Step
Procedure**

Testing Layer 2 VPN to Layer 2 VPN Connectivity (CE1 to CE5)

1. On Router CE1, use the **ping** command to test connectivity to Router CE5. Notice that the response time is in milliseconds, confirming that the ping response is returned.

```
user@CE1>ping 40.40.40.11

PING 40.40.40.11 (40.40.40.11): 56 data bytes
64 bytes from 40.40.40.11: icmp_seq=1 ttl=64 time=22.425 ms
64 bytes from 40.40.40.11: icmp_seq=2 ttl=64 time=1.299 ms
64 bytes from 40.40.40.11: icmp_seq=3 ttl=64 time=1.032 ms
64 bytes from 40.40.40.11: icmp_seq=4 ttl=64 time=1.029 ms
```

2. On Router CE5, use the **ping** command to test connectivity to Router CE1. Notice that the response time is in milliseconds, confirming that the ping response is returned.

```
user@CE5>ping 40.40.40.1

PING 40.40.40.1 (40.40.40.1): 56 data bytes
64 bytes from 40.40.40.1: icmp_seq=0 ttl=64 time=1.077 ms
64 bytes from 40.40.40.1: icmp_seq=1 ttl=64 time=0.957 ms
64 bytes from 40.40.40.1: icmp_seq=2 ttl=64 time=1.057 ms 1.017 ms
```

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

The relevant sample configuration for Router PE1 follows.

```
Router PE1  chassis {
              dump-on-panic;
              fpc 1 {
                pic 3 {
                  tunnel-services {
                    bandwidth 1g;
                  }
                }
              }
              network-services ethernet;
            }
            interfaces {
              xe-0/1/0 {
                unit 0 {
                  family inet {
                    address 10.10.2.1/30;
                  }
                  family mpls;
                }
              }
              xe-0/2/0 {
                unit 0 {
                  family inet {
                    address 10.10.3.1/30;
                  }
                }
              }
            }
```

```

    }
    family mpls;
  }
}
xe-0/3/0 {
  unit 0 {
    family inet {
      address 10.10.1.1/30;
    }
    family mpls;
  }
}
ge-1/0/0 {
  encapsulation ethernet-ccc;
  unit 0 {
    family ccc;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
}
routing-options {
  static {
    route 172.0.0.0/8 next-hop 172.19.59.1;
  }
  autonomous-system 65000;
}
protocols {
  mpls {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
bgp {
  group RR {
    type internal;
    local-address 1.1.1.1;
    family l2vpn {
      signaling;
    }
    neighbor 7.7.7.7;
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}

```

```
    }
  }
  ldp {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
routing-instances {
  L2VPN {
    instance-type l2vpn;
    interface ge-1/0/0.0;
    route-distinguisher 65000:1;
    vrf-target target:65000:2;
    protocols {
      l2vpn {
        encapsulation-type ethernet;
        site CE1 {
          site-identifier 1;
          interface ge-1/0/0.0 {
            remote-site-id 3;
          }
        }
      }
    }
  }
}
```

The relevant sample configuration for Router PE3 follows.

```
Router PE3  chassis {
               dump-on-panic;
               fpc 1 {
                 pic 3 {
                   tunnel-services {
                     bandwidth 1g;
                   }
                 }
               }
               network-services ethernet;
             }
            interfaces {
              xe-0/0/0 {
                unit 0 {
                  family inet {
                    address 10.10.20.2/30;
                  }
                  family mpls;
                }
              }
              xe-0/1/0 {
                unit 0 {
                  family inet {
                    address 10.10.6.1/30;
                  }
                }
              }
            }
          }
```

```
        family mpls;
    }
}
xe-0/2/0 {
    unit 0 {
        family inet {
            address 10.10.5.2/30;
        }
        family mpls;
    }
}
xe-0/3/0 {
    unit 0 {
        family inet {
            address 10.10.1.2/30;
        }
        family mpls;
    }
}
ge-1/0/1 {
    encapsulation ethernet-ccc;
    unit 0 {
        family ccc;
    }
}
iw0 {
    unit 0 {
        encapsulation ethernet-ccc;
        peer-unit 1;
    }
    unit 1 {
        encapsulation ethernet-ccc;
        peer-unit 0;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 3.3.3.3/32;
        }
    }
}
}
routing-options {
    static {
        route 172.0.0.0/8 next-hop 172.19.59.1;
    }
    autonomous-system 65000;
}
protocols {
    l2iw;
    mpls {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
```

```
}
bgp {
  group RR {
    type internal;
    local-address 3.3.3.3;
    family l2vpn {
      signaling;
    }
    neighbor 7.7.7.7;
  }
}
ospf {
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
}
ldp {
  interface all;
  interface fxp0.0 {
    disable;
  }
}
}
routing-instances {
  L2VPN-PE1 {
    instance-type l2vpn;
    interface iw0.0;
    route-distinguisher 65000:3;
    vrf-target target:65000:2;
    protocols {
      l2vpn {
        encapsulation-type ethernet;
        site CE3 {
          site-identifier 3;
          interface iw0.0 {
            remote-site-id 1;
          }
        }
      }
    }
  }
  L2VPN-PE5 {
    instance-type l2vpn;
    interface iw0.1;
    route-distinguisher 65000:33;
    vrf-target target:65000:2;
    protocols {
      l2vpn {
        encapsulation-type ethernet;
        site CE3 {
          site-identifier 3;
          interface iw0.1 {
            remote-site-id 5;
          }
        }
      }
    }
  }
}
```



```
}  
}  
}  
}  
}
```

**Related
Documentation**

- Layer 2 VPN Overview on page 3
- Layer 2 VPN Applications on page 5
- Using the Layer 2 Interworking Interface to Interconnect a Layer 2 VPN to a Layer 2 VPN on page 7

