

Network Configuration Example

Implementing Interprovider Layer 3 VPN Option A

Release

10.4



Published: 2010-10-08

Revision 1

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 Implementing Interprovider Layer 3 VPN Option A

Release 10.4

Copyright © 2010, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Arun Kumar

Editing: Nancy Kurahashi, Katie Smith, Marilyn Kerr, Roy Spencer

Illustration: Faith Bradford, Dawn Spencer

Cover Design: Edmonds Design

Revision History

October 2010—R1 Junos 10.4

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

Overview	1
Applications	1
Implementation	2
Example: Configuring Interprovider Layer 3 VPN Option A	3

Overview

This document describes one of four recommended interprovider and carrier-of-carriers solutions for situations in which the customer of a VPN service provider might be another service provider rather than an end customer. The customer service provider depends on the virtual private network (VPN) service provider (SP) to deliver a VPN transport service between the customer service provider's points of presence (POPs) or regional networks.

If the customer service provider's sites have different autonomous system (AS) numbers, then the VPN transit service provider supports carrier-of-carriers VPN service for the interprovider VPN service. This functionality might be used by a VPN customer who has connections to several different Internet service providers (ISPs), or different connections to the same ISP in different geographic regions, each of which has a different AS number.

Applications

A customer might require VPN services for different sites, yet the same SP is not available for all of those sites.

RFC 4364 suggests several methods to resolve this problem, including:

- Interprovider VRF-to-VRF connections at the AS boundary routers (ASBR) (not very scalable). This option is presented in *Implementing Interprovider Layer 3 VPN Option A*.
- Interprovider EBGp redistribution of labeled VPN-IPv4 routes from AS to neighboring AS (somewhat scalable). This option is presented in *Implementing Interprovider Layer 3 VPN Option B*.
- Interprovider multihop EBGp redistribution of labeled VPN-IPv4 routes between source and destination ASs, with EBGp redistribution of labeled IPv4 routes from AS to neighboring AS (very scalable). This option is presented in *Implementing Interprovider Layer 3 VPN Option C*.

Solutions might include elements of both the interprovider VPN solutions and the carrier-of-carriers solution. For example, a transit carrier might supply a service provider whose sites have different AS numbers, which makes the solution topology look like an interprovider solution (due to the different AS numbers). However, it is the same service for the transit carrier, so it really is a carrier-of-carriers service. This type of service solution is referred to as carrier-of-carriers VPN service for the interprovider VPN service.

In contrast, if the customer service provider's sites have the same AS number, then the VPN transit service provider delivers a carrier-of-carriers VPN service.

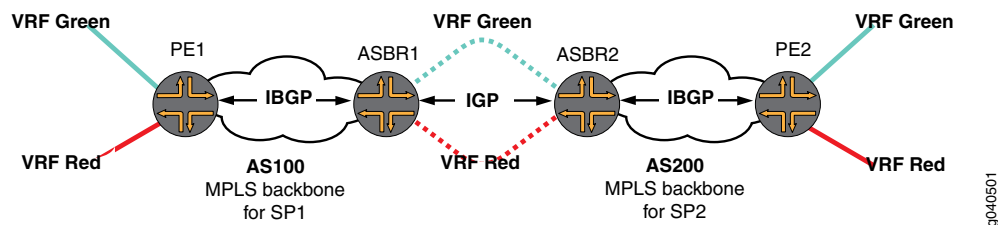
In addition to resolving the initial problem described above, carrier-of-carriers or interprovider VPN solutions may be used to solve other problems such as scalability and merging two service providers.

Implementation

This solution is the same as a regular VPN solution. There is no need to send MPLS packets to the neighboring AS. If SP1 and SP2 are connected to each other using a transit SP, the transit SP may provide a tunnel between SP1 and SP2 using a layer-2 VPN or any other IP tunneling technology.

The logical topology of the network is shown in Figure 1 on page 2.

Figure 1: Logical Topology of Interprovider Layer 3 VPN Option A



Related Documentation

- Example: Configuring Interprovider Layer 3 VPN Option A on page 3

Example: Configuring Interprovider Layer 3 VPN Option A

This example provides a step-by-step procedure to configure interprovider layer 3 VPN option A, which is one of the recommended implementations of MPLS VPN when that service is required by a customer that has more than one AS and but not all of the customer's ASs can be serviced by the same service provider. It is organized in the following sections:

- Requirements on page 3
- Configuration Overview and Topology on page 3
- Configuration on page 4

Requirements

This example uses the following hardware and software components:

- Junos OS Release 9.5 or later.
- Eight M Series, T Series, TX Series, or MX Series Juniper Networks routers.

Configuration Overview and Topology

This is the simplest and least scalable interprovider VPN solution to the problem of providing VPN services to a customer that has different sites, not all of which can use the same service provider (SP).

RFC 4364, section 10, refers to this method as Interprovider VRF-to-VRF connections at the AS border routers.

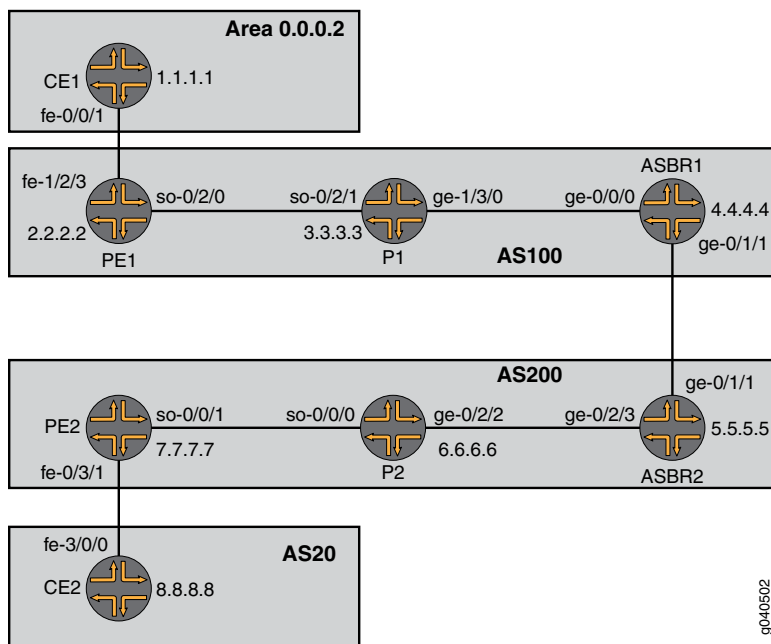
In this configuration:

- The VPN routing and forwarding (VRF) table in the ASBR of one AS is linked to the VRF table in the ASBR in the other AS. Each ASBR must contain a VRF instance for every VPN configured in both service provider networks. Then an IGP or BGP must be configured between the ASBRs. This has the disadvantage of limiting scalability.
- In this configuration, the autonomous system boundary routers (ASBRs) at both SPs are configured as regular PE routers, and provide MPLS L3 VPN service to the neighbor SP.
- Each PE router treats the other as if it were a customer edge (CE) router. ASBRs play the role of regular CE routers for the ASBR of the remote SP. ASBRs see each other as CE devices.
- A provider edge (PE) router in one autonomous system (AS) attaches directly to a PE router in another AS.
- The two PE routers are attached by multiple sub-interfaces, at least one for each of the VPNs whose routes need to be passed from AS to AS.
- The PE routers associate each sub-interface with a VPN routing and forwarding (VRF) table, and use EBGp to distribute unlabeled IPv4 addresses to each other.

- In this solution, all common VPNs defined at both PEs must also be defined at one or more ASBRs between the two SPs. This is not a very scalable methodology, especially when a transit SP is used by two regional SPs for interconnection.
- This is a procedure that is simple to configure and it does not require MPLS at the border between ASs. Additionally, it does not scale as well as other recommended procedures.

The topology of the network is shown in Figure 2 on page 4.

Figure 2: Physical Topology of Interprovider Layer 3 VPN Option A



Configuration



NOTE: The procedure presented here is written with the assumption that the reader is already familiar with MPLS MVPN configuration. This example focuses on explaining the unique configuration required for carrier-of-carriers solutions for VPN services to different sites.

To configure interprovider layer 3 VPN option A, perform the following tasks:

- Configuring Router CE1 on page 5
- Configuring Router PE1 on page 5
- Configuring Router P1 on page 8
- Configuring Router ASBR1 on page 9
- Configuring Router ASBR2 on page 11
- Configuring Router P2 on page 13
- Configuring Router PE2 on page 14

- Configuring Router CE2 on page 16
- Verifying the VPN Operation on page 17

Configuring Router CE1

- Step-by-Step Procedure**
1. On Router CE1, configure the IP address and protocol family on the Fast Ethernet interface for the link between Router CE1 and Router PE1. Specify the **inet** address family type.


```
[edit interfaces fe-0/0/1.0]
family inet {
  address 18.18.18.1/30;
}
```
 2. On Router CE1, configure the IP address and protocol family on the loopback interface. Specify the **inet** address family type.


```
[edit interfaces lo0]
unit 0 {
  family inet {
    address 1.1.1.1/32;
  }
}
```
 3. On Router CE1, configure an IGP. The IGP can be a static route, RIP, OSPF, ISIS, or EBGp. In this example we configure OSPF. Include the Fast Ethernet interface for the link between Router CE1 and Router PE1 and the logical loopback interface of Router CE1.


```
[edit protocols]
ospf {
  area 0.0.0.2 {
    interface fe-0/0/1.0;
    interface lo0.0;
  }
}
```

Configuring Router PE1

- Step-by-Step Procedure**
1. On Router PE1, configure IPv4 addresses on the SONET, Fast Ethernet, and logical loopback interfaces. Specify the **inet** address family on all of the interfaces. Specify the **mpls** address family on the SONET and Fast Ethernet interfaces.


```
[edit interfaces]
so-0/2/0 {
  unit 0 {
    family inet {
      address 19.19.19.1/30;
    }
    family mpls;
  }
}
fe-1/2/3 {
  unit 0 {
    family inet {
      address 18.18.18.2/30;
    }
  }
}
```

```
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 2.2.2.2/32;
        }
    }
}
```

2. On Router PE1, configure the routing instance for VPN2. Specify the **vrf** instance type and specify the customer-facing Fast Ethernet interface. Configure a route distinguisher to create a unique VPN-IPv4 address prefix. Apply the VRF import and export policies to enable the sending and receiving of route targets. Configure the OSPF protocol within the VRF. Specify the customer-facing Fast Ethernet interface and specify the export policy to export BGP routes into OSPF.

```
[edit routing-instances]
vpn2CE1 {
    instance-type vrf;
    interface fe-1/2/3.0;
    route-distinguisher 1:100;
    vrf-import vpnimport;
    vrf-export vpnexport;
    protocols {
        ospf {
            export bgp-to-ospf;
            area 0.0.0.2 {
                interface fe-1/2/3.0;
            }
        }
    }
}
```

3. On Router PE1, configure the RSVP and MPLS protocols to support the label-switched path (LSP). Configure the LSP to Router ASBR1 and specify the IP address of the logical loopback interface on Router ASBR1. Configure a BGP group. Specify the group type as **internal**. Specify the local address as the logical loopback interface on Router PE1. Specify the neighbor address as the logical loopback interface on Router ASBR1. Specify the **inet-vpn** address family and **unicast** traffic type to enable BGP to carry IPv4 network layer reachability information (NLRI) for VPN routes. Configure the OSPF protocol. Specify the core-facing SONET interface and specify the logical loopback interface on Router PE1.

```
[edit protocols]
rsvp {
    interface so-0/2/0.0;
    interface lo0.0;
}
mpls {
    label-switched-path To-ASBR1 {
        to 4.4.4.4;
    }
    interface so-0/2/0.0;
```

```

interface lo0.0;
}
bgp {
  group To_ASBR1 {
    type internal;
    local-address 2.2.2.2;
    neighbor 4.4.4.4 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-0/2/0.0;
    interface lo0.0;
  }
}

```

4. On Router PE1, configure the BGP local autonomous system number.

```

[edit routing-options]
autonomous-system 100;

```

5. On Router PE1, configure a policy to export the BGP routes into OSPF.

```

[edit policy-options]
policy-statement bgp-to-ospf {
  term 1 {
    from protocol bgp;
    then accept;
  }
  term 2 {
    then reject;
  }
}

```

6. On Router PE1, configure a policy to add the VRF route target to the routes being advertised for this VPN.

```

[edit policy-options]
policy-statement vpnexport {
  term 1 {
    from protocol ospf;
    then {
      community add test_comm;
      accept;
    }
  }
  term 2 {
    then reject;
  }
}

```

7. On Router PE1, configure a policy to import routes from BGP that have the **test_comm** community attached.

```
[edit policy-options]
  policy-statement vpnimport {
    term 1 {
      from {
        protocol bgp;
        community test_comm;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
```

8. On Router PE1, define the **test_comm** BGP community with a route target.

```
[edit policy-options]
  community test_comm members target:1:100;
```

Configuring Router P1

Step-by-Step Procedure

1. On Router P1, configure IP addresses for the SONET and Gigabit Ethernet interfaces. Enable the interfaces to process the **inet** and **mpls** address families. Configure the IP address for the **lo0.0** loopback interface and enable the interface to process the **inet** address family.

```
[edit interfaces]
  so-0/2/1 {
    unit 0 {
      family inet {
        address 19.19.2/30;
      }
      family mpls;
    }
  }
  ge-1/3/0 {
    unit 0 {
      family inet {
        address 20.20.20.1/30;
      }
      family mpls;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 3.3.3.3/32;
      }
    }
  }
```

2. On Router P1, configure the RSVP and MPLS protocols to support the LSP. Specify the SONET and Gigabit Ethernet interfaces.

Configure the OSPF protocol. Specify the SONET and Gigabit Ethernet interfaces and specify the logical loopback interface. Enable OSPF to support traffic engineering extensions.

```
[edit protocols]
rsvp {
  interface so-0/2/1.0;
  interface ge-1/3/0.0;
  interface lo0.0;
}
mpls {
  interface lo0.0;
  interface ge-1/3/0.0;
  interface so-0/2/1.0;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface ge-1/3/0.0;
    interface so-0/2/1.0;
    interface lo0.0;
  }
}
```

Configuring Router ASBR1

- Step-by-Step Procedure**
1. On Router ASBR1, configure IP addresses for the Gigabit Ethernet interfaces. Enable the interfaces to process the **inet** and **mpls** addresses families. Configure the IP addresses for the **lo0.0** loopback interface and enable the interface to process the **inet** address family.

```
[edit interfaces]
ge-0/0/0 {
  unit 0 {
    family inet {
      address 20.20.20.2/30;
    }
    family mpls;
  }
}
ge-0/1/1 {
  unit 0 {
    family inet {
      address 21.21.21.1/30;
    }
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 4.4.4.4/32;
    }
  }
}
```

2. On Router ASBR1, configure the **To_ASBR2** routing instance. Specify the **vrf** instance type and specify the core-facing Gigabit Ethernet interface. Configure a route distinguisher to create a unique VPN-IPv4 address prefix. Configure a route target for the VPN. Configure the BGP peer group within the VRF. Specify AS 200 as the peer AS and specify the IP address of the Gigabit Ethernet interface on Router ASBR2 as the neighbor address.

```
[edit routing instances]
To_ASBR2{
  instance-type vrf;
  interface ge-0/1/1.0;
  route-distinguisher 1:100;
  vrf-target target:1:100;
  protocols {
    bgp {
      group To_ASBR2 {
        type external;
        neighbor 21.21.21.2 {
          peer-as 200;
        }
      }
    }
  }
}
```

3. On Router ASBR1, configure the RSVP and MPLS protocols to support the LSP. Specify the Gigabit Ethernet interfaces.

Configure the OSPF protocol. Specify the SONET and Gigabit Ethernet interfaces and specify the logical loopback interface. Enable OSPF to support traffic engineering extensions.

```
[edit protocols]
rsvp {
  interface ge-0/0/0.0;
  interface lo0.0;
}
mpls {
  label-switched-path To_PE1 {
    to 2.2.2.2;
  }
  interface lo0.0;
  interface ge-0/0/0.0;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface ge-0/0/0.0;
    interface lo0.0;
  }
}
```

4. On Router ASBR1, create the **To_PE1** internal BGP peer group. Specify the local IP peer address as the local **lo0.0** address. Specify the neighbor IP peer address as the **lo0.0** interface address of Router PE1.

```
[edit protocols]
bgp {
  group To-PE1 {
    type internal;
    local-address 4.4.4.4;
    neighbor 2.2.2.2 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

5. On Router ASBR1, configure the BGP local autonomous system number.

```
[edit routing-options]
autonomous-system 100;
```

Configuring Router ASBR2

Step-by-Step Procedure

1. On Router ASBR2, configure IP addresses for the Gigabit Ethernet interfaces. Enable the interfaces to process the **inet** and **mpls** address families. Configure the IP address for the **lo0.0** loopback interface and enable the interface to process the **inet** address family.

```
[edit interfaces]
ge-0/1/1 {
  unit 0 {
    family inet {
      address 21.21.21.2/30;
    }
    family mpls;
  }
}
ge-0/2/3 {
  unit 0 {
    family inet {
      address 22.22.22.1/30;
    }
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 5.5.5.5/32;
    }
  }
}
```

2. On Router ASBR2, configure the **To_ASBR1** routing instance. Specify the **vrf** instance type and specify the core-facing Gigabit Ethernet interface. Configure a route distinguisher to create a unique VPN-IPv4 address prefix. Configure a route target for the VPN. Configure the BGP peer group within the VRF. Specify AS 100 as the peer AS and specify the IP address of the Gigabit Ethernet interface on Router ASBR1 as the neighbor address.

```
[edit routing-instances]
To_ASBR1 {
  instance-type vrf;
  interface ge-0/1/1.0;
  route-distinguisher 1:100;
  vrf-target target:1:100;
  protocols {
    bgp {
      group To_ASBR1 {
        type external;
        neighbor 21.21.21.1 {
          peer-as 100;
        }
      }
    }
  }
}
```

3. On Router ASBR2, configure the RSVP and MPLS protocols to support the LSP. Specify the Gigabit Ethernet interfaces.

Configure the OSPF protocol. Specify the SONET and Gigabit Ethernet interfaces and specify the logical loopback interface. Enable OSPF to support traffic engineering extensions.

```
[edit protocols]
rsvp {
  interface ge-0/2/3.0;
  interface lo0.0;
}
mpls {
  label-switched-path To_PE2 {
    to 7.7.7.7;
  }
  interface lo0.0;
  interface ge-0/2/3.0;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface ge-0/2/3.0;
    interface lo0.0;
  }
}
```

4. On Router ASBR2, create the **To-PE2** internal BGP peer group. Specify the local IP peer address as the local **lo0.0** address. Specify the neighbor IP peer address as the **lo0.0** interface address of Router PE2.

```
[edit protocols]
bgp {
  group To-PE2 {
    type internal;
    local-address 5.5.5.5;
    neighbor 7.7.7.7 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

```

    }
  }
}

```

5. On Router ASBR2, configure the BGP local autonomous system number.

```

[edit routing-options]
autonomous-system 200;

```

Configuring Router P2

Step-by-Step Procedure

1. On Router P2, configure IP addresses for the SONET and Gigabit Ethernet interfaces. Enable the interfaces to process the **inet** and **mpls** address families. Configure the IP address for the **lo0.0** loopback interface and enable the interface to process the **inet** address family.

```

[edit interfaces]
so-0/0/0 {
  unit 0 {
    family inet {
      address 23.23.23.1/30;
    }
    family mpls;
  }
}
ge-0/2/2 {
  unit 0 {
    family inet {
      address 22.22.22.2/30;
    }
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 6.6.6.6/32;
    }
  }
}

```

2. On Router P2, configure the RSVP and MPLS protocols to support the LSP. Specify the SONET and Gigabit Ethernet interfaces.

Configure the OSPF protocol. Specify the SONET and Gigabit Ethernet interfaces and specify the logical loopback interface. Enable OSPF to support traffic engineering extensions.

```

[edit protocols]
rsvp {
  interface so-0/0/0.0;
  interface ge-0/2/2.0;
  interface lo0.0;
}
mpls {
  interface lo0.0;
  interface ge-0/2/2.0;
}

```

```

        interface so-0/0/0.0;
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface ge-0/2/2.0;
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}

```

Configuring Router PE2

Step-by-Step Procedure

1. On Router PE2, configure IPv4 addresses on the SONET, Fast Ethernet, and logical loopback interfaces. Specify the **inet** address family on all of the interfaces. Specify the **mpls** address family on the SONET and Fast Ethernet interfaces.

```

[edit interfaces]
so-0/0/1 {
    unit 0 {
        family inet {
            address 23.23.23.2/30;
        }
        family mpls;
    }
}
fe-0/3/1 {
    unit 0 {
        family inet {
            address 24.24.24.1/30;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 7.7.7.7/32;
        }
    }
}
}

```

2. On Router PE2, configure the routing instance for VPN2. Specify the **vrf** instance type and specify the customer-facing Fast Ethernet interface. Configure a route distinguisher to create a unique VPN-IPv4 address prefix. Apply the VRF import and export policies to enable the sending and receiving of route targets. Configure the BGP peer group within the VRF. Specify AS 20 as the peer AS and specify the IP address of the Fast Ethernet interface on Router CE2 as the neighbor address.

```

[edit routing-instances]
vpn2CE2 {
    instance-type vrf;
    interface fe-0/3/1.0;
    route-distinguisher 1:100;
    vrf-import vpnimport;
    vrf-export vpnexport;
    protocols {

```

```

    bgp {
      group To_CE2 {
        peer-as 20;
        neighbor 24.24.24.2;
      }
    }
  }
}

```

3. On Router PE2, configure the RSVP and MPLS protocols to support the LSP. Configure the LSP to ASBR2 and specify the IP address of the logical loopback interface on Router ASBR2. Configure a BGP group. Specify the group type as **internal**. Specify the local address as the logical loopback interface on Router PE2. Specify the neighbor address as the logical loopback interface on the Router ASBR2. Specify the **inet-vpn** address family and **unicast** traffic type to enable BGP to carry IPv4 NLRI for VPN routes. Configure the OSPF protocol. Specify the core-facing SONET interface and specify the logical loopback interface on Router PE2.

```

[edit protocols]
rsvp {
  interface so-0/0/1.0;
  interface lo0.0;
}
mpls {
  label-switched-path To-ASBR2 {
    to 5.5.5.5;
  }
  interface so-0/0/1.0;
  interface lo0.0;
}
bgp {
  group To_ASBR2 {
    type internal;
    local-address 7.7.7.7;
    neighbor 5.5.5.5 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-0/0/1.0;
    interface lo0.0;
  }
}

```

4. On Router PE2, configure the BGP local autonomous system number.

```

[edit routing-options]
autonomous-system 200;

```

5. On Router PE2, configure a policy to add the VRF route target to the routes being advertised for this VPN.

```
[edit policy-options]
policy-statement vpnexport {
  term 1 {
    from protocol bgp;
    then {
      community add test_comm;
      accept;
    }
  }
  term 2 {
    then reject;
  }
}
```

6. On Router PE2, configure a policy to import routes from BGP that have the **test_comm** community attached.

```
[edit policy-options]
policy-statement vpnimport {
  term 1 {
    from {
      protocol bgp;
      community test_comm;
    }
    then accept;
  }
  term 2 {
    then reject;
  }
}
```

7. On Router PE2, define the **test_comm** BGP community with a route target.

```
[edit policy-options]
community test_comm members target:1:100;
```

Configuring Router CE2

Step-by-Step Procedure

1. On Router CE2, configure the IP address and protocol family on the Fast Ethernet interface for the link between Router CE2 and Router PE2. Specify the **inet** address family type.

```
[edit interfaces]
fe-3/0/0 {
  unit 0 {
    family inet {
      address 24.24.24.2/30;
    }
  }
}
```

2. On Router CE2, configure the IP address and protocol family on the loopback interface. Specify the **inet** address family type.

```
[edit interfaces lo0]
lo0 {
  unit 0 {
    family inet {
```



```

        address 8.8.8.8/32;
    }
}

```

- On Router CE2, configure an IGP. The IGP can be a static route, RIP, OSPF, ISIS, or EBGp. In this example, we configure EBGp. Specify AS **200** as the peer AS and specify the BGP neighbor IP address as the Fast Ethernet interface of Router PE2.

```

[edit protocols]
bgp {
  group To_PE2 {
    neighbor 24.24.24.1 {
      export myroutes;
      peer-as 200;
    }
  }
}

```

Verifying the VPN Operation

Step-by-Step Procedure

- Commit the configuration on each router.



NOTE: The MPLS labels shown in this example will be different than the labels used in your configuration.

- On Router PE1, display the routes for the **vpn2CE1** routing instance using the **show ospf route** command. Verify that the 1.1.1.1 route is learned from OSPF.

```
user@PE1> show ospf route instance vpn2CE1
```

Topology default Route Table:

Prefix	Path	Route	NH	Metric	NextHop	NextHop
	Type	Type	Type		Interface	addr/label
1.1.1.1	Intra	Router	IP	1	fe-1/2/3.0	18.18.18.1
1.1.1.1/32	Intra	Network	IP	1	fe-1/2/3.0	18.18.18.1
18.18.18.0/30	Intra	Network	IP	1	fe-1/2/3.0	18.18.18.1

- On Router PE1, use the **show route advertising-protocol** command to verify that Router PE1 advertises the 1.1.1.1 route to Router ASBR1 using MP-BGP with the VPN MPLS label.

```
user@PE1> show route advertising-protocol bgp 4.4.4.4 extensive
```

```

vpn2CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
* 1.1.1.1/32 (1 entry, 1 announced)
  BGP group To_PE1 type Internal
    Route Distinguisher: 1:100
    VPN Label: 299856
    Nexthop: Self
    Flags: Nexthop Change
    MED: 1
    Localpref: 100

```

```
AS path: [100] I
Communities: target:1:100 rte-type:0.0.0.2:1:0
```

4. On Router ASBR1, use the **show route receive-protocol** command to verify that the router receives and accepts the 1.1.1.1 route and places it in the **To_ASBR2.inet.0** routing table.

```
user@ASBR1> show route receive-protocol bgp 2.2.2.2 extensive

inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

To_ASBR2.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
* 1.1.1./32 (1 entry, 1 announced)
  Route Distinguisher: 1:100
  VPN Label: 299856
  Nexthop: 2.2.2.2
  MED: 1
  Localpref: 100
  AS path: I
  Communities: target:1:100 rte-type:0.0.0.2:1:0

MPLS.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

BGP.13VPN.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

* 1:100:1.1.1.1/32 (1 entry, 0 announced)
  Route Distinguisher: 1:100
  VPN Label: 299856
  Nexthop: 2.2.2.2
  MED: 1
  Localpref: 100
  AS path: I
  Communities: target:1:100 rte-type:0.0.0.2:1:0
```

5. On Router ASBR1, use the **show route advertising-protocol** command to verify that Router ASBR1 advertises the 1.1.1.1 route to Router ASBR2.

```
user@ASBR1> show route advertising-protocol bgp 21.21.21.2 extensive

To_ASBR2.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
* 1.1.1./32 (1 entry, 1 announced)
  BGP group To_ASBR2.inet.0 type External
  Nexthop: Self
  AS path: [100] I
  Communities: target:1:100 rte-type:0.0.0.2:1:0
```

6. On Router ASBR2, use the **show route receive-protocol** command to verify that the router receives and accepts the 1.1.1.1 route and places it in the **To_ASBR1.inet.0** routing table.

```
user@ASBR2> show route receive-protocol bgp 21.21.21.1 extensive

inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

To_ASBR1.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
* 1.1.1./32 (1 entry, 1 announced)
  Accepted
```

```

Nexthop: 21.21.21.1
AS path: 100 I
Communities: target:1:100 rte-type:0.0.0.2:1:0

```

```
MPLS.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
```

```
BGP.13VPN.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

7. On Router ASBR2, use the **show route advertising-protocol** command to verify that Router ASBR2 advertises the 1.1.1.1 route to Router PE2 in the **To-PE2** routing instance.

```
user@ASBR2> show route advertising-protocol bgp 7.7.7.7 extensive
```

```

To_ASBR1.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
* 1.1.1./32 (1 entry, 1 announced)
  BGP group To-PE2 type Internal
    Route Distinguisher: 1:100
    VPN Label: 299936
    Nexthop: Self
    Flags: Nexthop Change
    Localpref: 100
    AS path: [200] 100 I
    Communities: target:1:100 rte-type:0.0.0.2:1:0

```

8. On Router PE2, use the **show route receive-protocol** command to verify that the router receives and accepts the 1.1.1.1 route and places it in the **To_CE2.inet.0** routing table.

```
user@PE2> show route receive-protocol bgp 5.5.5.5 extensive
```

```
inet.0: 12 destinations, 13 routes (12 active, 0 holddown, 0 hidden)
```

```
inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
```

```
__juniper_private1__.inet.0: 14 destinations, 14 routes (8 active, 0 holddown, 6 hidden)
```

```
__juniper_private2__.inet.0: 1 destinations, 1 routes (0 active, 0 holddown, 1 hidden)
```

```
To_CE2.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
```

```

* 1.1.1./32 (1 entry, 1 announced)
  Accepted
    Route Distinguisher: 1:100
    VPN Label: 299936
    Nexthop: 5.5.5.5
    Localpref: 100
    AS path: 100 I
    AS path: Recorded
    Communities: target:1:100 rte-type:0.0.0.2:1:0

```

9. On Router PE2, use the **show route advertising-protocol** command to verify that Router PE2 advertises the 1.1.1.1 route to Router CE2 through the **To_CE2** peer group.

```
user@PE2> show route advertising-protocol bgp 24.24.24.2 extensive
```

```
To_CE2.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
```

```

* 1.1.1./32 (1 entry, 1 announced)
  BGP group To_CE2 type External
    Nexthop: Self

```

```
AS path: [200] 100 I
Communities: target:1:100 rte-type:0.0.0.2:1:0
```

10. On Router CE2, use the **show route** command to verify that Router CE2 receives the 1.1.1.1 route from Router PE2.

```
user@CE2> show route 1.1.1.1

inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32          *[BGP/170] 00:25:36, localpref 100
                    AS path: 200 100 I
                    > to 24.24.24.1 via fe-3/0/0.0
```

11. On Router CE2, use the **ping** command and specify 8.8.8.8 as the source of the ping packets to verify connectivity with Router CE1.

```
user@CE2> ping 1.1.1.1 source 8.8.8.8

PING 1.1.1.1 (1.1.1.1): 56 data bytes
64 bytes from 1.1.1.1: icmp_seq=0 ttl=58 time=4.672 ms
64 bytes from 1.1.1.1: icmp_seq=1 ttl=58 time=10.480 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=58 time=10.560 ms
```

12. On Router PE2, use the **show route** command to verify that the traffic is sent with an inner label of 299936 and a top label of 299776.

```
user@PE2> show route 1.1.1.1 detail

To_CE2.inet.0: 5 destinations, 6 routes (5 active, 0 holddown, 0 hidden)
1.1.1.1/32 (1 entry, 1 announced)
  *BGP Preference: 170/-101
    Route Distinguisher: 1:100
    Next hop type: Indirect
    Next-hop reference count: 6
    Source: 5.5.5.5
    Next hop type: Router, Next hop index: 648
    Next hop: via so-0/0/1.0 weight 0x1, selected
    Label-switched-path To-ASBR2
    Label operation: Push 299936, Push 299776(top)
    Protocol next hop: 5.5.5.5
    Push 299984
    Indirect next hop: 8c6109c 262143
    State: <Secondary Active Int Ext>
    Local AS: 200 Peer AS: 200
    Age: 3:37 Metric2: 2
    Task: BGP_200.5.5.5+179
    Announcement bits (3): 0-RT 1-KRT 2-BGP RT Background
    AS path: 100 I
    AS path: Recorded
    Communities: target:1:100 rte-type:0.0.0.2:1:0
    Accepted
    VPN Label: 299984
    Localpref: 100
    Router ID: 5.5.5.5
    Primary Routing Table BGP.13VPN.0
```

13. On Router ASBR2, use the **show route table** command to verify that Router ASBR2 receives the traffic.

```
1ab@ASBR2# show route table mpls.0 detail
```

```
299936 (1 entry, 1 announced)
  *VPN      Preference: 170
            Next hop type: Router, Next hop index: 649
            Next-hop reference count: 2
            Source: 21.21.21.1
            Next hop: 21.21.21.1 via ge-0/1/1.0, selected
            Label operation: Pop
            State: <Active Int Ext>
            Local AS: 200
            Age: 9:54
            Task: BGP RT Background
            Announcement bits (1): 0-KRT
            AS path: 100 I
            Ref Cnt: 1
            Communities: target:1:100 rte-type:0.0.0.2:1:0
```

14. On Router ASBR2, use the **show route table** command to verify that Router ASBR2 receives the traffic.

```
1ab@ASBR2# show route 1.1.1.1 detail
```

```
To_ASBR1.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
1.1.1.1/32 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
            Next hop type: Router, Next hop index: 576
            Next-hop reference count: 3
            Source: 21.21.21.1
            Next hop: 21.21.21.1 via ge-0/1/1.0, selected
            State: <Active Ext>
            Peer AS: 100
            Age: 13:07
            Task: BGP_100.21.21.21.1+53372
            Announcement bits (2): 0-KRT 1-BGP RT Background
            AS path: 100 I
            Communities: target:1:100 rte-type:0.0.0.2:1:0
            Accepted
            Localpref: 100
            Router ID: 21.21.21.1
```

15. On Router ASBR1, use the **show route** command to verify that ASBR1 sends traffic toward PE1 with the top label **299792** and VPN label **299856**.

```
1ab@ASBR1# show route 1.1.1.1 detail
```

```
To_ASBR2.inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
1.1.1.1/32 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
            Route Distinguisher: 1:100
            Next hop type: Indirect
            Next-hop reference count: 3
            Source: 2.2.2.2
            Next hop type: Router, Next hop index: 669
            Next hop: 20.20.20.1 via ge-0/0/0.0 weight 0x1, selected
            Label-switched-path To_PE1
            Label operation: Push 299856, Push 299792(top)
            Protocol next hop: 2.2.2.2                      Push 299856
            Indirect next hop: 8af70a0 262143
            State: <Secondary Active Int Ext>
            Local AS: 100 Peer AS: 100
```

```

Age: 12:15      Metric: 1      Metric2: 2
Task: BGP_100.2.2.2.2+58065
Announcement bits (2): 0-KRT 1-BGP RT Background
AS path: I
Communities: target:1:100 rte-type:0.0.0.2:1:0
VPN Label: 299856
Localpref: 100
Router ID: 2.2.2.2
Primary Routing Table BGP.13VPN.0

```

16. On Router PE1, use the **show route table** command to verify that Router PE1 receives the traffic with label **299856**, pops the label, and the traffic is sent toward Router CE1 through interface **fe-1/2/3.0**.

```
1ab@PE1# show route table mpls.0 detail
```

```

299856 (1 entry, 1 announced)
  *VPN      Preference: 170
    Next hop type: Router, Next hop index: 666
    Next-hop reference count: 2
    Next hop: 18.18.18.1 via fe-1/2/3.0, selected
    Label operation: Pop
    State: <Active Int Ext>
    Local AS: 100
    Age: 17:38
    Task: BGP RT Background
    Announcement bits (1): 0-KRT
    AS path: I
    Ref Cnt: 1
    Communities: rte-type:0.0.0.2:1:0

```

17. On Router PE1, use the **show route** command to verify that PE1 receives the traffic after the top label is popped by Router P and the traffic is sent toward Router CE1 through interface **fe-1/2/3.0**.

```
1ab@PE1# show route 1.1.1.1 detail
```

```

vpn2CE1.inet.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
1.1.1.1/32 (1 entry, 1 announced)
  *OSPF      Preference: 10
    Next hop type: Router, Next hop index: 634
    Next-hop reference count: 3
    Next hop: 18.18.18.1 via fe-1/2/3.0, selected
    State: <Active Int>
    Age: 18:42      Metric: 1
    Area: 0.0.0.2
    Task: VPN2alice-OSPFv2
    Announcement bits (2): 2-KRT 3-BGP RT Background
    AS path: I
    Communities: rte-type:0.0.0.2:1:0

```

Related Documentation • Overview on page 1