# Implementing Interprovider Layer 3 VPN Option C

Published: 2012-02-27

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



YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. 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

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

# Table of Contents

# List of Figures

# List of Tables



# About the Documentation



## Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

## Supported Platforms

For the features described in this document, the following platforms are supported:

- T Series
- M Series
- MX Series

## Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

## Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

## Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

   [edit]
   user@host# **edit system scripts**
   [edit system scripts]

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

   [edit system scripts]
   user@host# **load merge relative /var/tmp/ex-script-snippet.conf**
   load complete

For more information about the **load** command, see the *Junos OS CLI User Guide*.

## Documentation Conventions

Table 1 on page xi defines notice icons used in this guide.

### Table 1: Notice Icons

| Icon | Meaning | Description |
|---|---|---|
| | Informational note | Indicates important features or instructions. |
| | Caution | Indicates a situation that might result in loss of data or hardware damage. |
| | Warning | Alerts you to the risk of personal injury or death. |
| | Laser warning | Alerts you to the risk of personal injury from a laser. |

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

### Table 2: Text and Syntax Conventions

| Convention | Description | Examples |
|---|---|---|
| **Bold text like this** | Represents text that you type. | To enter configuration mode, type the **configure** command:<br><br>user@host> **configure** |
| `Fixed-width text like this` | Represents output that appears on the terminal screen. | `user@host>` **show chassis alarms**<br><br>`No alarms currently active` |

## Table 2: Text and Syntax Conventions *(continued)*

<table>
<tr><th>Convention</th><th>Description</th><th>Examples</th></tr>
<tr><td>Italic text like this</td><td>• Introduces important new terms.<br>• Identifies book names.<br>• Identifies RFC and Internet draft titles.</td><td>• A policy term is a named structure that defines match conditions and actions.<br>• Junos OS System Basics Configuration Guide<br>• RFC 1997, BGP Communities Attribute</td></tr>
<tr><td>Italic text like this</td><td>Represents variables (options for which you substitute a value) in commands or configuration statements.</td><td>Configure the machine's domain name:<br><br>[edit]<br>root@# set system domain-name<br>  domain-name</td></tr>
<tr><td>Text like this</td><td>Represents names of configuration statements, commands, files, and directories; interface names; configuration hierarchy levels; or labels on routing platform components.</td><td>• To configure a stub area, include the stub statement at the [edit protocols ospf area area-id] hierarchy level.<br>• The console port is labeled CONSOLE.</td></tr>
<tr><td>< > (angle brackets)</td><td>Enclose optional keywords or variables.</td><td><b>stub</b> <<b>default-metric <i>metric</i></b>>;</td></tr>
<tr><td>| (pipe symbol)</td><td>Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.</td><td>broadcast | multicast<br><br>(string1 | string2 | string3)</td></tr>
<tr><td># (pound sign)</td><td>Indicates a comment specified on the same line as the configuration statement to which it applies.</td><td>rsvp { # Required for dynamic MPLS only</td></tr>
<tr><td>[ ] (square brackets)</td><td>Enclose a variable for which you can substitute one or more values.</td><td>community name members [ community-ids ]</td></tr>
<tr><td>Indention and braces ( { } )</td><td>Identify a level in the configuration hierarchy.</td><td rowspan="2">[edit]<br>routing-options {<br>  static {<br>    route default {<br>      nexthop address;<br>      retain;<br>    }<br>  }<br>}</td></tr>
<tr><td>; (semicolon)</td><td>Identifies a leaf statement at a configuration hierarchy level.</td></tr>
<tr><td colspan="3">J-Web GUI Conventions</td></tr>
<tr><td>Bold text like this</td><td>Represents J-Web graphical user interface (GUI) items you click or select.</td><td>• In the Logical Interfaces box, select All Interfaces.<br>• To cancel the configuration, click Cancel.</td></tr>
<tr><td>> (bold right angle bracket)</td><td>Separates levels in a hierarchy of J-Web selections.</td><td>In the configuration editor hierarchy, select Protocols>Ospf.</td></tr>
</table>

## Documentation Feedback

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

- Document or topic name
- URL or page number
- Software release version (if applicable)

## Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf .
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/ .
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

### Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/

- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

## Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/ .
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html .

PART 1

# Overview

CHAPTER 1

# Introduction to Layer 3 VPN Option C



## Interprovider Layer 3 VPN Option C 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.

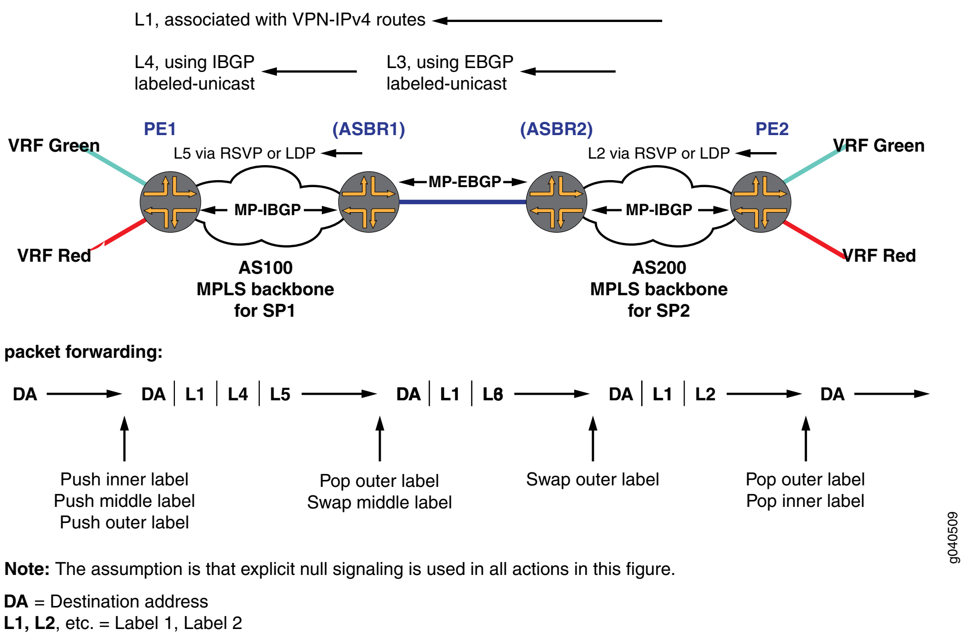
## Implementation

This document describes implementing interprovider layer 3 VPN option C, 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 all of their AS cannot be serviced by the same service provider.

In this method, only routes internal to the service provider networks are announced between ASBRs. This is achieved by using the **family inet labeled-unicast** statements in the IBGP and EBGP configuration on the PE routers. Labeled IPv4 (not VPN-IPv4). routes are exchanged by the ASBRs to support MPLS. An MP-EBGP session between the end PEs is used for the announcement of VPN-IPv4 routes. In this manner, VPN connectivity is maintained while keeping VPN-IPv4 routes out of the network core.

The logical topology of the network is shown in Figure 1 on page 4

### Figure 1: Logical Topology of Interprovider Layer 3 VPN Option C




**Related Documentation**

- Example: Configuring Interprovider Layer 3 VPN Option C on page 7

PART 2

# Configuration

CHAPTER 2

# Layer 3 VPN Option C Configuration Example



## Example: Configuring Interprovider Layer 3 VPN Option C

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



### Requirements

This example requires the following hardware and software components:

- Junos OS Release 9.5 or later.
- Eight Juniper Networks M Series Multiservice Edge Routers, T Series Core Routers, TX Matrix Routers, or MX Series 3D Universal Edge Routers.

## Configuration Overview and Topology

Interprovider layer 3 VPN option C is a very 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 SP.

*RFC 4364* section 10, refers to this method as 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.

This solution is similar to the solution described in *Implementing Interprovider Layer 3 VPN Option B*, except internal IPv4 unicast routes are advertised instead of external VPN-IPv4-unicast routes, using EBGP. Internal routes are internal to leaf SPs (SP1 and SP2 in this example), and external routes are those learned from the end customer requesting VPN services.

In this configuration:

- After the loopback address of Router PE2 is learned by Router PE1 and the loopback address of Router PE1 is learned by Router PE2, the end PE routers establish an MP-EBGP session for exchanging VPN-IPv4 routes.
- Since VPN-IPv4 routes are exchanged among end PE routers, any other router on the path from Router PE1 and Router PE2 does not need to keep or install VPN-IPv4 routes in their routing information base (RIB) or forwarding information base (FIB) tables.
- An MPLS path needs to be established between Router PE1 and Router PE2.

*RFC 4364* describes only one solution that uses a BGP labeled-unicast approach. In this approach, the ASBR routers advertise the loopback addresses of the PE routers and associate each prefix with a label according to *RFC 3107*. Service providers may use RSVP or LDP to establish an LSP between ASBR routers and PE routers in their internal network.

In this network, ASBR2 receives label information associated with the loopback IP address of Router PE1 and advertises another label to Router ASBR1 using MP-EBGP labeled-unicast. Meanwhile, the ASBRs build their own MPLS forwarding table according to the received and advertised routes and labels. Router ASBR1 uses its own IP address as the next-hop information.

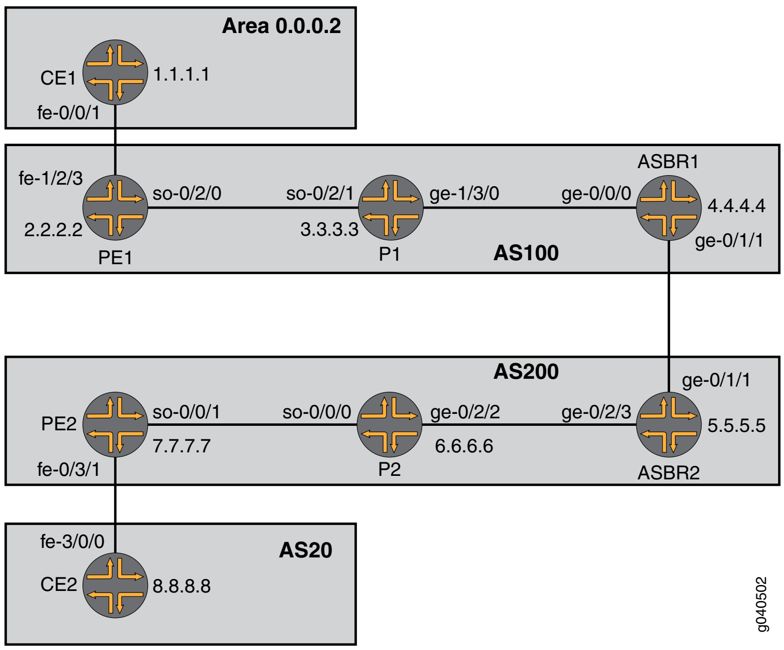
Router ASBR2 receives this prefix associated with a label, assigns another label, changes the next-hop address to its own address, and advertises it to Router PE1. Router PE1 now has an update with the label information and next-hop to Router ASBR1. Also, Router PE1 already has a label associated with the IP address of Router ASBR1. If Router PE1 sends an IP packet to Router PE2, it pushes two labels: one for the IP address of Router PE2 (obtained using MP-IBGP labeled-unicast advertisement) and one for the IP address of Router ASBR1 (obtained using LDP or RSVP).

Router ASBR1 then pops the outer label and swaps the inner label with the label learned from a neighbor ASBR for its neighboring PE router. Router ASBR2 performs a similar function and swaps the incoming label (only one) and pushes another label that is associated with the address of Router PE2. Router PE2 pops both labels and passes the remaining IP packet to its own CPU. After the end-to-end connection among the PE routers is created, the PE routers establish an MP-EBGP session to exchange VPN-IPv4 routes.

In this solution, PE routers push three labels onto the IP packet coming from the VPN end user. The inner-most label, obtained using MP-EBGP, determines the correct VPN routing and forwarding (VRF) routing instance at the remote PE. The middle label is associated with the IP address of the remote PE and is obtained from an ASBR using MP-IBGP labeled-unicast. The outer label is associated with the IP addresses of the ASBRs and is obtained using LDP or RSVP.

The physical topology of the network is shown in Figure 2 on page 9.

### Figure 2: Physical Topology of Interprovider Layer 3 VPN Option C




## 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 C, perform the following tasks:


- Configuring Router CE1 on page 10
- Configuring Router PE1 on page 10
- Configuring Router P1 on page 13
- Configuring Router ASBR1 on page 14
- Configuring Router ASBR2 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 logical 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 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;
        }
    }
}
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 LSP. Configure the LSP to Router ASBR1 and specify the IP address of the logical loopback interface on Router ASBR1. 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;
}
ospf {
```

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

4. On Router PE1, configure the **To_ASBR1** peer 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** address family. For a PE router to install a route in the VRF, the next hop must resolve to a route stored within the **inet.3** table. The **labeled-unicast resolve-vpn** statements allow labeled routes to be placed in the **inet.3** routing table for route resolution, which are then resolved for PE router connections where the remote PE is located across another AS.

```
[edit protocols]
bgp {
    group To_ASBR1 {
        type internal;
        local-address 2.2.2.2;
        neighbor 4.4.4.4 {
            family inet {
                labeled-unicast {
                    resolve-vpn;
                }
            }
        }
    }
}
```

5. On Router PE1, configure multihop EBGP toward PE2. Specify the **inet-vpn** family.

```
[edit protocols]
bgp {
    group To_PE2 {
        multihop {
            ttl 20;
        }
        local-address 2.2.2.2;
        family inet-VPN {
            unicast;
        }
        neighbor 7.7.7.7 {
            peer-as 200;
        }
    }
}
```

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

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

7. 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;
    }
  }
```

8. 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;
    }
  }
```

9. 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;
    }
  }
```

10. 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 19.19.19.2/30;
      }
      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 RSVP and MPLS protocols to support the LSP. Specify the Gigabit Ethernet interfaces and the logical loopback interface. Include the **traffic-engineering bgp-igp-both-ribs** statement at the **[edit protocols mpls]** hierarchy level.

   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 {
   traffic-engineering bgp-igp-both-ribs;
   label-switched-path To_PE1 {
      to 2.2.2.2;
   }
   interface lo0.0;
   interface ge-0/0/0.0;
   interface ge-0/1/1.0;
}
ospf {
   traffic-engineering;
   area 0.0.0.0 {
      interface ge-0/0/0.0;
      interface lo0.0;
```

```
    }
  }
```

3. 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 Gigabit Ethernet 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 {
        labeled-unicast;
      }
    }
  }
```

4. On Router ASBR1, create the **To-ASBR2** external BGP peer group. Enable the router to use BGP to advertise network layer reachability information (NLRI) for unicast routes. Specify the neighbor IP peer address as the Gigabit Ethernet interface address on Router ASBR2.

```
[edit protocols]
group To-ASBR2 {
  type external;
  family inet {
    labeled-unicast;
  }
  export To-ASBR2;
  neighbor 21.21.21.2 {
    peer-as 200;
  }
}
```

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

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

6. On Router PE 1, configure a policy to import routes from BGP that have the **test_comm** community attached.

```
[edit policy-options]
policy-statement To-ASBR2 {
  term 1 {
    route-filter 2.2.2.2/32 exact;
    then accept;
  }
  term 2 {
    then reject;
  }
}
```

## 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 RSVP and MPLS protocols to support the LSP. Specify the Gigabit Ethernet interfaces. Include the **traffic-engineering bgp-igp-both-ribs** statement at the **[edit protocols mpls]** hierarchy level.

   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 {
   traffic-engineering bgp-igp-both-ribs;
   label-switched-path To_PE2 {
      to 7.7.7.7;
   }
   interface lo0.0
   interface ge-0/2/3.0;
   interface ge-0/1/1.0;
}
ospf {
```

```
    traffic-engineering;
    area 0.0.0.0 {
      interface ge-0/2/3.0;
      interface lo0.0;
    }
  }
```

3. 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 {
        labeled-unicast;
      }
    }
  }
}
```

4. On Router ASBR2, create the **To-ASBR1** external BGP peer group. Enable the router to use BGP to advertise NLRI for unicast routes. Specify the neighbor IP peer address as the Gigabit Ethernet interface address on Router ASBR1.

```
[edit protocols]
bgp {
  group To-ASBR1 {
    type external;
    family inet {
      labeled-unicast;
    }
    export To-ASBR1;
    neighbor 21.21.21.1 {
      peer-as 100;
    }
  }
}
```

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

```
[edit]
lab@ASBR2# show routing-options
autonomous-system 200;
```

6. On Router ASBR2, configure a policy to import routes from BGP that match the **4.4.4.4/32** route.

```
[edit policy-options]
policy-statement To-ASBR2 {
  term 1 {
    route-filter 4.4.4.4/32 exact;;
    then accept;
  }
  term 2 {
```

```
    then reject;
  }
}
```

## 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** addresses families. Configure the IP addresses 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 interface.

```
[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;
    }
  }
}
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 CE1 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 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;
}
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-0/0/1.0;
    interface lo0.0;
  }
}
```

4. On Router PE2, configure the **To_ASBR2** 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.

```
[edit protocols]
bgp {
  group To_ASBR2 {
    type internal;
    local-address 7.7.7.7;
    neighbor 5.5.5.5 {
      family inet {
        labeled-unicast {
          resolve-vpn;
        }
      }
    }
  }
}
```

5. On Router PE2, configure multihop EBGP towards Router PE1 Specify the **inet-vpn** address family.

```
[edit protocols]
bgp {
  group To_PE1 {
    multihop {
```

```
            ttl 20;
        }
        family inet-vpn {
            unicast;
        }
        neighbor 2.2.2.2 {
            peer-as 100;
        }
    }
}
```

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

```
[edit routing-options]
autonomous-system 200;
```

7. 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;
    }
}
```

8. 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;
    }
}
```

9. 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 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;
    }
  }
}
```

3. 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 the BGP neighbor IP address as the logical loopback interface of Router PE1.

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

## Verifying the VPN Operation

**Step-by-Step Procedure**

1. Commit the configuration on each router.

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

2. 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
```

3. On Router PE1, use the **show route advertising-protocol** command to verify that Router PE1 advertises the **1.1.1.1** route to Router PE2 using MP-BGP with the VPN MPLS label.

```
user@PE1> show route advertising-protocol bgp 7.7.7.7 extensive

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
* 1:100:1.1.1.1/32 (1 entry, 1 announced)
 BGP group To_PE2 type External
     Route Distinguisher: 1:100
     VPN Label: 300016
     Nexthop: Self
     Flags: Nexthop Change
     MED: 1
     AS path: [100] I
     Communities: target:1:100 rte-type:0.0.0.2:1:0
```

4. On Router ASBR1, use the **show route advertising-protocol** command to verify that Router ASBR1 advertises the **2.2.2.2** route to Router ASBR2.

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

inet.0: 14 destinations, 16 routes (14 active, 0 holddown, 0 hidden)
* 2.2.2.2/32 (2 entries, 1 announced)
 BGP group To-PE2 type External
     Route Label: 300172
     Nexthop: Self
     Flags: Nexthop Change
     MED: 2
     AS path: [100] I
```

5. On Router ASBR2, use the **show route receive-protocol** command to verify that the router receives and accepts the **2.2.2.2** route and places it in the **To_ASBR2.inet.0** routing table.

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

inet.0: 10 destinations, 11 routes (10 active, 0 holddown, 0 hidden)
* 2.2.2.2/32 (1 entry, 1 announced)
     Accepted
     Route Label: 300172
     Nexthop: 21.21.21.1
     MED: 2
     AS path: 100 I
```

6. On Router ASBR2, use the **show route advertising-protocol** command to verify that Router ASBR2 advertises the **2.2.2.2** route to Router PE2 in the **To-PE2** routing instance.

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

inet.0: 10 destinations, 11 routes (10 active, 0 holddown, 0 hidden)
* 2.2.2.2/32 (1 entry, 1 announced)
```

```
 BGP group To-PE2 type Internal
     Route Label: 300192
     Nexthop: Self
     Flags: Nexthop Change
     MED: 2
     Localpref: 100
     AS path: [200] 100 I
```

7. On Router PE2, use the **show route receive-protocol** command to verify that Router PE2 receives the route and puts it in the **inet.0.** routing table. Verify that Router PE2 also receives the update from Router PE1 and accepts the route.

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

inet.0: 13 destinations, 14 routes (13 active, 0 holddown, 0 hidden)
* 2.2.2.2/32 (1 entry, 1 announced)
     Accepted
     Route Label: 300192
     Nexthop: 5.5.5.5
     MED: 2
     Localpref: 100
     AS path: 100 I
     AS path: Recorded

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

* 2.2.2.2/32 (1 entry, 1 announced)
     Accepted
     Route Label: 300192
     Nexthop: 5.5.5.5
     MED: 2
     Localpref: 100
     AS path: 100 I
     AS path: Recorded
```

8. On Router PE2, use the **show route receive-protocol** command to verify that Router PE2 puts the route in the routing table of the **To_CE2** routing instance and advertises the route to Router CE2 using EBGP.

```
user@PE2> show route receive-protocol bgp 2.2.2.2 detail

inet.0: 17 destinations, 18 routes (17 active, 0 holddown, 0 hidden)

inet.3: 6 destinations, 6 routes (6 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: 4 destinations, 5 routes (4 active, 0 holddown, 0 hidden)
* 1.1.1.1/32 (1 entry, 1 announced)
     Accepted
     Route Distinguisher: 1:100
     VPN Label: 300016
     Nexthop: 2.2.2.2
     MED: 1
     AS path: 100 I
     AS path: Recorded
     Communities: target:1:100 rte-type:0.0.0.2:1:0
```

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

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

bgp.l3vpn.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

* 1:100:1.1.1.1/32 (1 entry, 0 announced)
     Accepted
     Route Distinguisher: 1:100
     VPN Label: 300016
     Nexthop: 2.2.2.2
     MED: 1
     AS path: 100 I
     AS path: Recorded
     Communities: target:1:100 rte-type:0.0.0.2:1:0

__juniper_private1__.inet6.0: 4 destinations, 4 routes (4 active, 0
holddown, 0 hidden)
```

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.inet.0: 4 destinations, 5 routes (4 active, 0 holddown, 0 hidden)
* 1.1.1.1/32 (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=4.786 ms
64 bytes from 1.1.1.1: icmp_seq=1 ttl=58 time=10.210 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=58 time=10.588 ms
```

12. On Router PE2, use the **show route** command to verify that the traffic is sent with an inner label of **300016**, a middle label of **300192**, and a top label of **299776**.

```
user@PE2> show route 1.1.1.1 detail

To_CE2.inet.0: 4 destinations, 5 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: 653
                Next hop: via so-0/0/1.0 weight 0x1, selected
                Label-switched-path To-PE2
             Label operation: Push 300016, Push 300192, Push 299776(top)
                Protocol next hop: 2.2.2.2
                Push 300016
                Indirect next hop: 8c61138 262142
                State: <Secondary Active Ext>
                Local AS:   200 Peer AS:   100
                Age: 17:33      Metric: 1       Metric2: 2
                Task: BGP_100.2.2.2.2+62319
                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: 300016
                Localpref: 100
                Router ID: 2.2.2.2
                Primary Routing Table bgp.l3vpn.0
```

13. On Router ASBR2, use the **show route table** command to verify that Router ASBR2 receives the traffic after the top label is popped by Router P2. Verify that label **300192** is a swapped with label **300176** and the traffic is sent towards Router ASBR1 using interface ge-0/1/1.0. At this point, the bottom label **300016** is preserved.

```
lab@ASBR2# show route table mpls.0 detail

300192 (1 entry, 1 announced)
        *VPN    Preference: 170
                Next hop type: Router, Next hop index: 660
                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: Swap 300176
                State: <Active Int Ext>
                Local AS:   200
                Age: 24:01
                Task: BGP RT Background
                Announcement bits (1): 0-KRT
                AS path: 100 I
                Ref Cnt: 1
```

14. On Router ASBR1, use the **show route table** command to verify that when Router ASBR1 receives traffic with label **300176**, it swaps the label with **299824** to reach Router PE1.

```
user@ASBR1> show route table mpls.0 detail

300176 (1 entry, 1 announced)
        *VPN    Preference: 170
                Next hop type: Router, Next hop index: 651
                Next-hop reference count: 2
                Next hop: 20.20.20.1 via ge-0/0/0.0 weight 0x1, selected
                Label operation: Swap 299824
                State: <Active Int Ext>
                Local AS:   100
```

```
                Age: 25:53
                Task: BGP RT Background
                Announcement bits (1): 0-KRT
                AS path: I
                Ref Cnt: 1
```

15. On Router PE1, use the **show route table** command to verify that Router PE1 receives the traffic after the top label is popped by Router P1. Verify that label **300016** is popped and the traffic is sent towards Router CE1 using interface **fe-1/2/3.0**.

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

300016 (1 entry, 1 announced)
        *VPN    Preference: 170
                Next hop type: Router, Next hop index: 643
                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: 27:37
                Task: BGP RT Background
                Announcement bits (1): 0-KRT
                AS path: I
                Ref Cnt: 1
                Communities: rte-type:0.0.0.2:1:0
```

**Related Documentation**

- Interprovider Layer 3 VPN Option C Overview on page 3

PART 3

# Index

# Index