

# Network Configuration Example

## Configuring BGP Autodiscovery for LDP VPLS

Release

12.1



Published: 2012-02-27

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 Configuring BGP Autodiscovery for LDP VPLS*

Release 12.1

Copyright © 2012, Juniper Networks, Inc.

All rights reserved.

The information in this document is current as of the date on the title page.

## 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

Virtual Private LAN Service Overview .....	1
VPLS Protocol Operation .....	3
Example: Configuring BGP Autodiscovery for LDP VPLS .....	5
Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups .....	22



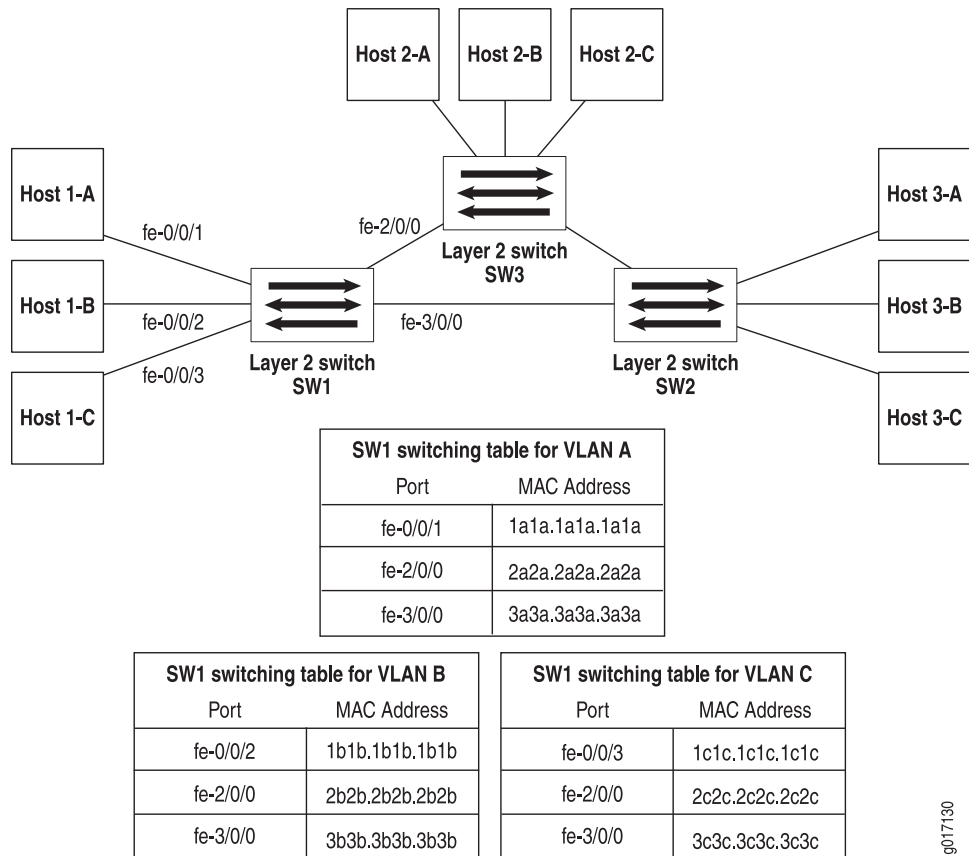
## Virtual Private LAN Service Overview

Ethernet is an increasingly important component of a service provider's slate of service offerings. Many customers are requesting the ability to connect LAN locations around the world. To fulfill customer needs, service providers have had to set up complex point-to-point Layer 2 virtual private networks (VPNs) or connect expensive Layer 2 switches to handle traffic.

Virtual private LAN service (VPLS) meets the growing Ethernet needs of service providers and their customers. VPLS is an Ethernet-based multipoint-to-multipoint Layer 2 VPN. With VPLS, multiple Ethernet LAN sites can be connected to each other across an MPLS backbone. To the customer, all sites interconnected by VPLS appear to be on the same Ethernet LAN (even though traffic travels across a service provider network).

Before VPLS, the only way you could connect Ethernet LAN sites together was to set up a non-VPLS Layer 2 VPN or install multiple Layer 2 Ethernet switches. [Figure 1 on page 1](#) shows how three switches can be connected to each other.

Figure 1: Ethernet Switching Example

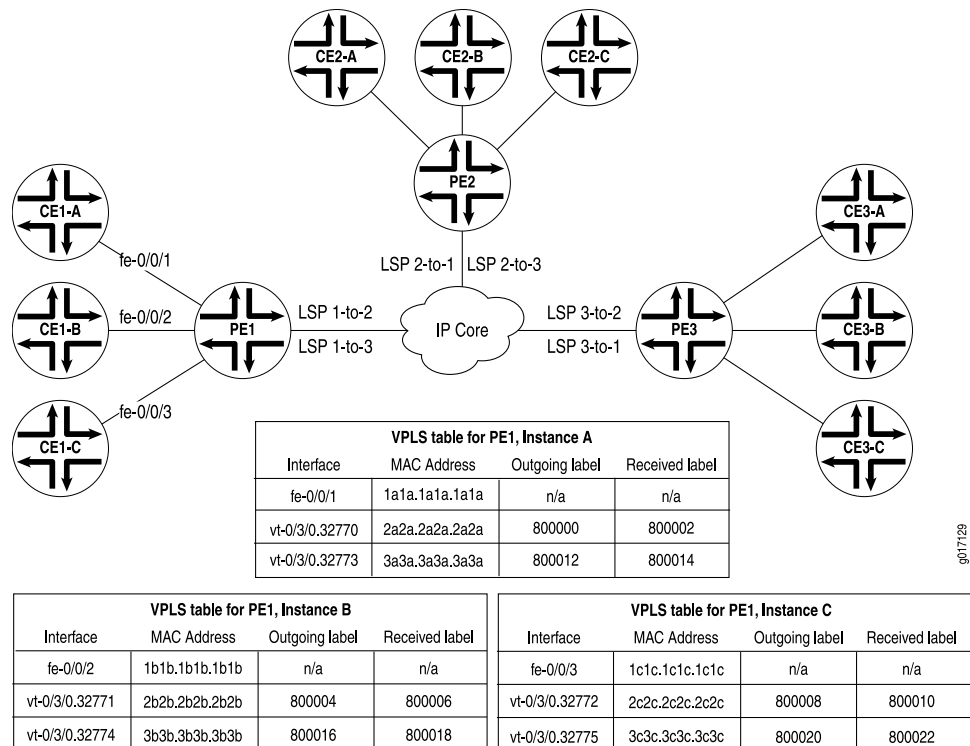


g017130

A typical switch builds its Layer 2 switching table with media access control (MAC) address and interface information learned from traffic received from other switches. If a switch does not have an exit port associated with a particular destination, it floods traffic for that destination to all ports except the port where the traffic originated. When reachability information for a destination is received, this information is added to the switching table. If the switching table has an entry for the destination, the switch sends the traffic directly to the intended recipient through the associated port listed in the switching table.

Figure 2 on page 2 shows a VPLS network comparable to the switch example and explains how VPLS functions similarly to Ethernet switches (assuming a Spanning Tree Protocol (STP) is configured).

Figure 2: VPLS Introductory Example



Notice that Layer 2 information gathered by a switch (for example, MAC addresses and interface ports) is included in the VPLS instance table. However, instead of requiring all VPLS interfaces to be physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS label-switched path (LSP) and arrive on a virtual port. The virtual port emulates a local, physical port. Traffic can be learned, forwarded, or flooded to the virtual port almost identically to the way traffic is sent to a local port.

The VPLS table learns MAC address and interface information for both physical and virtual ports. If no activity is seen for a particular MAC address, it is purged from the table over time.

---

As shown in [Figure 2 on page 2](#), the main difference between a physical port and a virtual port is that the router captures additional information from a virtual port—an outgoing MPLS label used to reach the remote site, and an incoming MPLS label for VPLS traffic received from the remote site.

When you configure VPLS on a routing platform, a virtual port is generated as a logical interface on a virtual loopback tunnel (**vt**) interface or a label-switched interface (LSI). On Juniper Networks M Series Multiservice Edge Routers and Juniper Networks T Series Core Routers, virtual ports are created dynamically on **vt** interfaces if you install a PIC that supports virtual tunnels. With VPLS, you must install at least one Tunnel Services, Link Services, or Adaptive Services PIC in each VPLS provider edge (PE) router. On Juniper Networks MX Series 3D Universal Edge Routers, virtual ports are created dynamically on **vt** interfaces if you configure tunnel services on one of the four Packet Forwarding Engines (PFEs) included in each Dense Port Concentrator (DPC). If your routing platform does not offer tunnel services through a PIC or PFE, you can configure VPLS to create virtual ports on LSI logical interfaces.

One property of flooding behavior in VPLS is that traffic received from remote PE routers is never forwarded to other PE routers. This restriction helps prevent loops in the core network. If a customer edge (CE) Ethernet switch has redundant connections to the same PE router, you must enable the STP to prevent loops.

The paths that emulate a Layer 2 point-to-point connection over a packet-switched network are called *pseudowires*. The pseudowires are signaled using either BGP or LDP.

**Related  
Documentation**

- Example: VPLS Configuration (BGP and LDP Interworking) at [Virtual Private LAN Services](#)
- [Example: Configuring BGP Autodiscovery for LDP VPLS on page 5](#)
- [Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups on page 22](#)

## VPLS Protocol Operation

---

VPLS provides a multipoint-to-multipoint Ethernet service that can span one or more metropolitan areas and provides connectivity between multiple sites as if these sites were attached to the same Ethernet LAN.

VPLS uses an IP and MPLS service provider infrastructure. From a service provider's point of view, use of IP and MPLS routing protocols and procedures instead of the Spanning Tree Protocol (STP), and MPLS labels instead of VLAN IDs, significantly improves the scalability of the VPLS service.

VPLS carries Ethernet traffic across a service provider network, so the provider network must mimic an Ethernet network in some ways. When a PE router configured with a VPLS routing instance receives a packet from a CE device, it first determines whether it has an exit port associated with the destination of the VPLS packet. If it does, it forwards the packet to the appropriate PE router or CE device. If it does not, it broadcasts the packet to all the other PE routers and CE devices that are members of that VPLS routing instance. In both cases, the CE device receiving the packet must be different from the device sending the packet.

When a PE router receives a packet from another PE router, it first determines whether it has an exit port associated with the destination of the VPLS packet. The PE router either forwards the packet or drops it, depending on whether the destination is a local or remote CE device. The PE router has three options (scenarios):

- If the destination is a local CE device, the PE router forwards the packet to it.
- If the destination is a remote CE device (connected to another PE router), it discards the packet.
- If the PE router cannot determine the destination of the VPLS packet, it floods the packet to its attached CE devices.

A VPLS can be directly connected to an Ethernet switch. Layer 2 information gathered by an Ethernet switch, such as media access control (MAC) addresses and interface ports, is included in the VPLS routing instance table. However, instead of all VPLS interfaces being physical switch ports, the router allows remote traffic for a VPLS instance to be delivered across an MPLS LSP and to arrive on a virtual port. The virtual port emulates a local, physical port. Traffic can be learned, forwarded, or flooded to the virtual port in almost the same way as traffic sent to a local port.

The VPLS routing table is populated with MAC addresses and interface information for both physical and virtual ports. One difference between a physical port and a virtual port is that on a virtual port, the router captures the outgoing MPLS label used to reach the remote site, and an incoming MPLS label for VPLS traffic received from the remote site. The virtual port is generated dynamically on a Tunnel Services PIC when you configure VPLS on a Juniper Networks M Series Multiservice Edge Router or T Series Core Router. A Tunnel Services PIC is required on each M Series or T Series VPLS router.

If your router has an Enhanced FPC installed, you can configure VPLS without a Tunnel Services PIC. To do so, use a label-switched interface (LSI) to provide VPLS functionality. An LSI MPLS label is used as the inner label for VPLS. This label maps to a VPLS routing instance. On the PE router, the LSI label is stripped and then mapped to a logical LSI interface. The Layer 2 Ethernet frame is then forwarded using the LSI interface to the correct VPLS routing instance.

One restriction on flooding behavior in VPLS is that traffic received from remote PE routers is never forwarded to other PE routers. This restriction helps prevent loops in the core network. This also means that the core network of PE routers must be fully meshed. Additionally, if a CE Ethernet switch has two or more connections to the same PE router, you must enable the STP on the CE switch to prevent loops.

Junos OS supports both forwarding equivalency class (FEC) 128 and FEC 129. FEC 128 requires manually configured pseudowires. FEC 129 uses VPLS autodiscovery to convey endpoint information. After PE routers are autodiscovered, pseudowires are created automatically.

**Related  
Documentation**

- [Example: VPLS Configuration \(BGP and LDP Interworking\)](#)
- [Example: Configuring BGP Autodiscovery for LDP VPLS on page 5](#)
- [Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups on page 22](#)



---

## Example: Configuring BGP Autodiscovery for LDP VPLS

---

This example describes how to configure BGP autodiscovery for LDP VPLS, as specified in forwarding equivalency class (FEC) 129. FEC 129 uses BGP autodiscovery to convey endpoint information, so you do not need to manually configure pseudowires.

- [Requirements on page 5](#)
- [Overview on page 5](#)
- [Configuration on page 7](#)
- [Verification on page 21](#)

### Requirements

This example uses the following hardware and software components:

- Four MX Series 3D Universal Edge Routers
- Junos OS Release 10.4R2 or later

If you are using M Series or T Series routers, the PE routers must have either virtual loopback tunnel (**vt**) interfaces or label-switched interfaces (LSIs). On M Series and T Series routers, VPLS uses tunnel-based PICs to create virtual ports on **vt** interfaces. If you do not have a tunnel-based PIC installed on your M Series or T Series router, you can still configure VPLS by using LSIs to support the virtual ports. Use of LSIs requires Ethernet-based PICs installed in an Enhanced Flexible PIC Concentrator (FPC).

You do not need to use routers for the CE devices. For example, the CE devices can be EX Series Ethernet Switches.

### Overview

All PE routers in a VPLS network operate like a large, distributed Ethernet switch to provide Layer 2 services to attached devices. This example shows a minimum configuration for PE routers and CE devices to create an autodiscovered VPLS network. The topology consists of five routers: two PE routers, two CE routers, and an optional route reflector (RR). The PE routers use BGP to autodiscover two different VPLS instances that are configured on both PE routers. Then the PE routers use LDP to automatically signal two pseudowires between the discovered end points. Finally, the PE routers bring up both VPLS instances for forwarding traffic. Each CE device is configured with two VLANs, with each VLAN belonging to different VPLS instances in the PE routers.

This example includes the following settings:

- **auto-discovery-only**—Allows the router to process only the autodiscovery network layer reachability information (NLRI) update messages for LDP-based Layer 2 VPN and VPLS update messages (BGP\_L2VPN\_AD\_NLRI) (FEC 129). Specifically, the **auto-discovery-only** statement notifies the routing process (rpd) to expect autodiscovery-related NLRI messages so that information can be deciphered and used by LDP and VPLS. You can configure this statement at the global, group, and neighbor levels for BGP. The **auto-discovery-only** statement must be configured on all PE routers in the VPLS. If you configure route reflection, the **auto-discovery-only** statement is also required on P routers that act as the route reflector in supporting FEC 129-related updates.

The **signaling** statement is not included in this example but is discussed here for completeness. The **signaling** statement allows the router to process only the BGP\_L2VPN\_NLRIs used for BGP-based Layer 2 VPNs (FEC 128).

For interoperation scenarios in which a PE router must support both types of NLRI (FEC 128 and FEC 129), you can configure both the **signaling** statement and the **auto-discovery-only** statement. For example, a single PE router might need to process a combination of BGP-signaled virtual private wire service (VPWS) and LDP-signaled VPLS assisted by BGP autodiscovery. Configuring both the **signaling** statement and the **auto-discovery-only** statement together allows both types of signaling to run independently. The **signaling** statement is supported at the same hierarchy levels as the **auto-discovery-only** statement.

- **cluster**—Configuring a route reflector is optional for FEC 129 autodiscovered PE routers. In this example, the **cluster** statement configures Router RR to be a route reflector in the IBGP group. For inbound updates, BGP autodiscovery NLRI messages are accepted if the router is configured to be a route reflector or if the **keep all** statement is configured in the IBGP group.
- **l2vpn-id**—Specifies a globally unique Layer 2 VPN community identifier for the instance. This statement is configurable for routing instances of type **vpls**.

You can configure the following formats for the community identifier:

- Autonomous system (AS) number format—**l2vpn-id:as-number:2-byte-number**. For example: **l2vpn-id:100:200**. The AS number can be in the range from 1 through 65,535.
- IPv4 format—**l2vpn-id:ip-address:2-byte-number**. For example: **l2vpn-id:10.1.1.1:2**.
- **vrf-target**—Defines the import and export route targets for the NLRI. You must either configure the **vrf-target** statement or the **vrf-import** and **vrf-export** statements to define the instance import and export policy or the import and export route targets for the NLRI. This example uses the **vrf-target** statement.
- **route-distinguisher**—Forms part of the BGP autodiscovery NLRI and distinguishes to which VPN or VPLS routing instance each route belongs. Each route distinguisher is a 6-byte value. You must configure a unique route distinguisher for each routing instance.

You can configure the following formats for the route distinguisher:

- AS number format—**as-number:2-byte-number**

- IPv4 format—*ip-address:2-byte-number*

Two notable statements are included in this example. These statements are important for interoperability with other vendors' equipment. The interoperability statements are not necessary for the topology that is used in this example, but they are included for completeness.

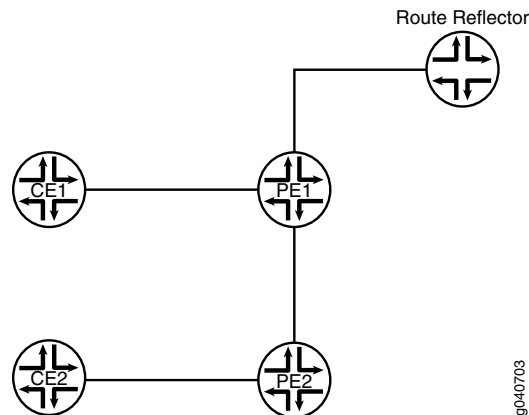
The interoperability statements are as follows:

- **input-vlan-map pop**—Removes an outer VLAN tag from the top of the VLAN tag stack.
- **output-vlan-map push**—Adds an outer VLAN tag in front of the existing VLAN tag.

### Topology Diagram

Figure 3 on page 7 shows the topology used in this example.

Figure 3: BGP Autodiscovery for LDP VPLS



### Configuration

#### CLI Quick Configuration

To quickly configure BGP autodiscovery for LDP VPLS, copy the following commands, remove any line breaks, and then paste the commands into the CLI of each device.

On Router PE1:

```

[edit]
set interfaces ge-0/1/0 vlan-tagging
set interfaces ge-0/1/0 encapsulation flexible-ethernet-services
set interfaces ge-0/1/0 unit 100 encapsulation vlan-vpls
set interfaces ge-0/1/0 unit 100 vlan-id 100
set interfaces ge-0/1/0 unit 100 input-vlan-map pop
set interfaces ge-0/1/0 unit 100 output-vlan-map push
set interfaces ge-0/1/0 unit 100 family vpls
set interfaces ge-0/1/0 unit 200 encapsulation vlan-vpls
set interfaces ge-0/1/0 unit 200 vlan-id 200
set interfaces ge-0/1/0 unit 200 family vpls
set interfaces ge-0/1/1 unit 0 description "PE1 to PE2"
set interfaces ge-0/1/1 unit 0 family inet address 8.0.40.100/24
set interfaces ge-0/1/1 unit 0 family iso
set interfaces ge-0/1/1 unit 0 family mpls
  
```

```
set interfaces ge-0/3/0 unit 0 description "PE1 to RR"
set interfaces ge-0/3/0 unit 0 family inet address 8.0.70.100/24
set interfaces ge-0/3/0 unit 0 family iso
set interfaces ge-0/3/0 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 8.0.0.100/32
set routing-options router-id 8.0.0.100
set routing-options autonomous-system 100
set protocols mpls interface lo0.0
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group int type internal
set protocols bgp group int local-address 8.0.0.100
set protocols bgp group int family l2vpn auto-discovery-only
set protocols bgp group int neighbor 8.0.0.107
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances vpls100 instance-type vpls
set routing-instances vpls100 interface ge-0/1/0.100
set routing-instances vpls100 route-distinguisher 8.0.0.100:100
set routing-instances vpls100 l2vpn-id l2vpn-id:100:100
set routing-instances vpls100 vrf-target target:100:100
set routing-instances vpls100 protocols vpls no-tunnel-services
set routing-instances vpls200 instance-type vpls
set routing-instances vpls200 interface ge-0/1/0.200
set routing-instances vpls200 route-distinguisher 8.0.0.100:200
set routing-instances vpls200 l2vpn-id l2vpn-id:100:200
set routing-instances vpls200 vrf-target target:100:208
set routing-instances vpls200 protocols vpls no-tunnel-services
```

On Device CE1:

```
[edit]
set interfaces ge-1/2/1 vlan-tagging
set interfaces ge-1/2/1 mtu 1400
set interfaces ge-1/2/1 unit 100 vlan-id 100
set interfaces ge-1/2/1 unit 100 family inet address 3.0.100.103/24
set interfaces ge-1/2/1 unit 200 vlan-id 200
set interfaces ge-1/2/1 unit 200 family inet address 3.0.200.103/24
set protocols ospf area 0.0.0.0 interface ge-1/2/1.100
set protocols ospf area 0.0.0.0 interface ge-1/2/1.200
```

On Router PE2:

```
[edit]
set interfaces ge-1/1/0 vlan-tagging
set interfaces ge-1/1/0 encapsulation flexible-ethernet-services
set interfaces ge-1/1/0 unit 100 encapsulation vlan-vpls
set interfaces ge-1/1/0 unit 100 vlan-id 100
set interfaces ge-1/1/0 unit 100 input-vlan-map pop
set interfaces ge-1/1/0 unit 100 output-vlan-map push
set interfaces ge-1/1/0 unit 100 family vpls
set interfaces ge-1/1/0 unit 200 encapsulation vlan-vpls
```

---

```
set interfaces ge-1/1/0 unit 200 vlan-id 200
set interfaces ge-1/1/0 unit 200 family vpls
set interfaces ge-1/2/1 unit 0 description "PE2 to PE1"
set interfaces ge-1/2/1 unit 0 family inet address 8.0.40.104/24
set interfaces ge-1/2/1 unit 0 family iso
set interfaces ge-1/2/1 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 8.0.0.104/32
set routing-options router-id 8.0.0.104
set routing-options autonomous-system 100
set protocols mpls interface lo0.0
set protocols mpls interface all
set protocols mpls interface fxp0.0 disable
set protocols bgp group int type internal
set protocols bgp group int local-address 8.0.0.104
set protocols bgp group int family l2vpn auto-discovery-only
set protocols bgp group int neighbor 8.0.0.107
set protocols isis level 1 disable
set protocols isis interface ge-1/2/1.0
set protocols isis interface lo0.0
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
set routing-instances vpls100 instance-type vpls
set routing-instances vpls100 interface ge-1/1/0.100
set routing-instances vpls100 route-distinguisher 8.0.0.104:100
set routing-instances vpls100 l2vpn-id l2vpn-id:100:100
set routing-instances vpls100 vrf-target target:100:100
set routing-instances vpls100 protocols vpls no-tunnel-services
set routing-instances vpls200 instance-type vpls
set routing-instances vpls200 interface ge-1/1/0.200
set routing-instances vpls200 route-distinguisher 8.0.0.104:200
set routing-instances vpls200 l2vpn-id l2vpn-id:100:200
set routing-instances vpls200 vrf-target target:100:208
set routing-instances vpls200 protocols vpls no-tunnel-services
```

On Device CE2:

```
[edit]
set interfaces ge-1/1/0 vlan-tagging
set interfaces ge-1/1/0 mtu 1400
set interfaces ge-1/1/0 unit 100 vlan-id 100
set interfaces ge-1/1/0 unit 100 family inet address 3.0.100.105/24
set interfaces ge-1/1/0 unit 200 vlan-id 200
set interfaces ge-1/1/0 unit 200 family inet address 3.0.200.105/24
set protocols ospf area 0.0.0.0 interface ge-1/1/0.100
set protocols ospf area 0.0.0.0 interface ge-1/1/0.200
```

On Router RR:

```
[edit]
set interfaces ge-1/3/2 unit 0 description "RR to PE1"
set interfaces ge-1/3/2 unit 0 family inet address 8.0.70.107/24
set interfaces ge-1/3/2 unit 0 family iso
set interfaces ge-1/3/2 unit 0 family mpls
set interfaces lo0 unit 0 family inet address 8.0.0.107/32
set routing-options router-id 8.0.0.107
set routing-options autonomous-system 100
```

```
set protocols bgp group int type internal
set protocols bgp group int local-address 8.0.0.107
set protocols bgp group int family l2vpn auto-discovery-only
set protocols bgp group int cluster 107.107.107.107
set protocols bgp group int neighbor 8.0.0.100
set protocols bgp group int neighbor 8.0.0.104
set protocols isis level 1 disable
set protocols isis interface all
set protocols isis interface fxp0.0 disable
set protocols isis interface lo0.0
set protocols ldp interface all
set protocols ldp interface fxp0.0 disable
set protocols ldp interface lo0.0
```

---

### Router PE1

#### Step-by-Step Procedure

To configure Router PE1:

1. Configure the interfaces, the interface encapsulation, and the protocol families.

```
[edit]
user@PE1# edit interfaces
[edit interfaces]
user@PE1# set ge-0/1/0 encapsulation flexible-ethernet-services
user@PE1# set ge-0/1/0 unit 100 encapsulation vlan-vpls
user@PE1# set ge-0/1/0 unit 100 family vpls
user@PE1# set ge-0/1/0 unit 200 encapsulation vlan-vpls
user@PE1# set ge-0/1/0 unit 200 family vpls
user@PE1# set ge-0/1/1 unit 0 description "PE1 to PE2"
user@PE1# set ge-0/1/1 unit 0 family inet address 8.0.40.100/24
user@PE1# set ge-0/1/1 unit 0 family iso
user@PE1# set ge-0/1/1 unit 0 family mpls
user@PE1# set ge-0/3/0 unit 0 description "PE1 to RR"
user@PE1# set ge-0/3/0 unit 0 family inet address 8.0.70.100/24
user@PE1# set ge-0/3/0 unit 0 family iso
user@PE1# set ge-0/3/0 unit 0 family mpls
user@PE1# set lo0 unit 0 family inet address 8.0.0.100/32
```

2. Configure the VLANs.

```
[edit interfaces]
user@PE1# set ge-0/1/0 vlan-tagging
user@PE1# set ge-0/1/0 unit 100 vlan-id 100
user@PE1# set ge-0/1/0 unit 100 input-vlan-map pop
user@PE1# set ge-0/1/0 unit 100 output-vlan-map push
user@PE1# set ge-0/1/0 unit 200 vlan-id 200
user@PE1# exit
```

3. Configure the protocol-independent properties.

We recommend that the router ID be the same as the local address. (See the **local-address** statement in Step 4.)

```
[edit]
user@PE1# edit routing-options
[edit routing-options]
user@PE1# set router-id 8.0.0.100
```

---

```
user@PE1# set autonomous-system 100
user@PE1# exit
```

4. Configure IBGP, including the **auto-discovery-only** statement.

```
[edit]
user@PE1# edit protocols
[edit protocols]
user@PE1# set bgp group int type internal
user@PE1# set bgp group int local-address 8.0.0.100
user@PE1# set bgp group int family l2vpn auto-discovery-only
user@PE1# set bgp group int neighbor 8.0.0.107
```

5. Configure MPLS, LDP, and an IGP.

```
[edit protocols]
user@PE1# set mpls interface lo0.0
user@PE1# set mpls interface all
user@PE1# set mpls interface fxp0.0 disable
user@PE1# set isis level 1 disable
user@PE1# set isis interface all
user@PE1# set isis interface fxp0.0 disable
user@PE1# set isis interface lo0.0
user@PE1# set ldp interface all
user@PE1# set ldp interface fxp0.0 disable
user@PE1# set ldp interface lo0.0
user@PE1# exit
```

6. Configure the routing instances.

The **no-tunnel-services** statement is required if you are using LSI interfaces for VPLS instead of **vt** interfaces.

```
[edit]
user@PE1# edit routing-instances
[edit routing-instances]
user@PE1# set vpls100 instance-type vpls
user@PE1# set vpls100 interface ge-0/1/0.100
user@PE1# set vpls100 route-distinguisher 8.0.0.100:100
user@PE1# set vpls100 l2vpn-id l2vpn-id:100:100
user@PE1# set vpls100 vrf-target target:100:100
user@PE1# set vpls100 protocols vpls no-tunnel-services
user@PE1# set vpls200 instance-type vpls
user@PE1# set vpls200 interface ge-0/1/0.200
user@PE1# set vpls200 route-distinguisher 8.0.0.100:200
user@PE1# set vpls200 l2vpn-id l2vpn-id:100:200
user@PE1# set vpls200 vrf-target target:100:208
user@PE1# set vpls200 protocols vpls no-tunnel-services
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@PE1# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show routing-options**, and **show routing-instances** commands. If the

output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show interfaces
ge-0/1/0 {
  vlan-tagging;
  encapsulation flexible-ethernet-services;
  unit 100 {
    encapsulation vlan-vpls;
    vlan-id 100;
    input-vlan-map pop;
    output-vlan-map push;
    family vpls;
  }
  unit 200 {
    encapsulation vlan-vpls;
    vlan-id 200;
    family vpls;
  }
}
ge-0/1/1 {
  unit 0 {
    description "PE1 to PE2";
    family inet {
      address 8.0.40.100/24;
    }
    family iso;
    family mpls;
  }
}
ge-0/3/0 {
  unit 0 {
    description "PE1 to RR";
    family inet {
      address 8.0.70.100/24;
    }
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 8.0.0.100/32;
    }
  }
}

user@PE1# show protocols
mpls {
  interface lo0.0;
  interface all;
  interface fxp0 disable;
}
bgp {
  group int {
```



```

    type internal;
    local-address 8.0.0.100;
    family l2vpn {
        auto-discovery-only;
    }
    neighbor 8.0.0.107;
}
}
isis {
    level 1 disable;
    interface all;
    interface lo0.0;
    interface fxp0 disable;
}
ldp {
    interface lo0.0;
    interface all;
    interface fxp0 disable;
}

user@PE1# show routing-options
router-id 8.0.0.100;
autonomous-system 100;

user@PE1# show routing-instances
vpls100 {
    instance-type vpls;
    interface ge-0/1/0.100;
    route-distinguisher 8.0.0.100:100;
    l2vpn-id l2vpn-id:100:100;
    vrf-target target:100:100;
    protocols {
        vpls {
            no-tunnel-services;
        }
    }
}
vpls200 {
    instance-type vpls;
    interface ge-0/1/0.200;
    route-distinguisher 8.0.0.100:200;
    l2vpn-id l2vpn-id:100:200;
    vrf-target target:100:208;
    protocols {
        vpls {
            no-tunnel-services;
        }
    }
}
}

```

### Device CE1

#### Step-by-Step Procedure

To configure Device CE1:

1. Configure interface addresses and the interface maximum transmission unit (MTU).  
[edit]

```
user@CE1# edit interfaces
[edit interfaces]
user@CE1# set ge-1/2/1 mtu 1400
user@CE1# set ge-1/2/1 unit 100 family inet address 3.0.100.103/24
user@CE1# set ge-1/2/1 unit 200 family inet address 3.0.200.103/24
```

2. Configure VLANs.

```
[edit interfaces]
user@CE1# set ge-1/2/1 vlan-tagging
user@CE1# set ge-1/2/1 unit 100 vlan-id 100
user@CE1# set ge-1/2/1 unit 200 vlan-id 200
user@CE1# exit
```

3. Configure an IGP.

```
user@CE1# edit protocols
[edit protocols]
user@CE1# set ospf area 0.0.0.0 interface ge-1/2/1.100
user@CE1# set ospf area 0.0.0.0 interface ge-1/2/1.200
user@CE1# exit
```

4. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE1# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE1# show interfaces
ge-1/2/1 {
  vlan-tagging;
  mtu 1400;
  unit 100 {
    vlan-id 100;
    family inet {
      address 3.0.100.103/24;
    }
  }
  unit 200 {
    vlan-id 200;
    family inet {
      address 3.0.200.103/24;
    }
  }
}

user@CE1# show protocols
ospf {
  area 0.0.0.0 {
    interface ge-1/2/1.100;
    interface ge-1/2/1.200;
  }
}
```

---

## Router PE2

---

### Step-by-Step Procedure

To configure Router PE2:

1. Configure the interfaces, the interface encapsulation, and the protocol families.

```
[edit]
user@PE2# edit interfaces
[edit interfaces]
user@PE2# set ge-1/1/0 encapsulation flexible-ethernet-services
user@PE2# set ge-1/1/0 unit 100 encapsulation vlan-vpls
user@PE2# set ge-1/1/0 unit 100 family vpls
user@PE2# set ge-1/1/0 unit 200 encapsulation vlan-vpls
user@PE2# set ge-1/1/0 unit 200 family vpls
user@PE2# set ge-1/2/1 unit 0 description "PE2 to PE1"
user@PE2# set ge-1/2/1 unit 0 family inet address 8.0.40.104/24
user@PE2# set ge-1/2/1 unit 0 family iso
user@PE2# set ge-1/2/1 unit 0 family mpls
user@PE2# set lo0 unit 0 family inet address 8.0.0.104/32
```

2. Configure the VLANs.

```
[edit interfaces]
user@PE2# set ge-1/1/0 vlan-tagging
user@PE2# set ge-1/1/0 unit 100 vlan-id 100
user@PE2# set ge-1/1/0 unit 100 input-vlan-map pop
user@PE2# set ge-1/1/0 unit 100 output-vlan-map push
user@PE2# set ge-1/1/0 unit 200 vlan-id 200
user@PE2# exit
```

3. Configure the protocols-independent properties.

We recommend that the router ID be the same as the local address. (See the **local-address** statement in Step 4.)

```
[edit]
user@PE2# edit routing-options
[edit routing-options]
user@PE2# set router-id 8.0.0.104
user@PE2# set autonomous-system 100
```

4. Configure IBGP, including the **auto-discovery-only** statement.

```
[edit]
user@PE2# edit protocols
[edit protocols]
user@PE2# set bgp group int type internal
user@PE2# set bgp group int local-address 8.0.0.104
user@PE2# set bgp group int family l2vpn auto-discovery-only
user@PE2# set bgp group int neighbor 8.0.0.107
```

5. Configure MPLS, LDP, and an IGP.

```
[edit protocols]
user@PE2# set mpls interface lo0.0
user@PE2# set mpls interface all
user@PE2# set mpls interface fxp0.0 disable
user@PE2# set isis level 1 disable
```

```
user@PE2# set isis interface ge-1/2/1.0
user@PE2# set isis interface lo0.0
user@PE2# set ldp interface all
user@PE2# set ldp interface fxp0.0 disable
user@PE2# set ldp interface lo0.0
user@PE2# exit
```

6. Configure the routing instances.

The **no-tunnel-services** statement is required if you are using LSI interfaces for VPLS instead of **vt** interfaces.

```
[edit]
user@PE2# edit routing-instances
[edit routing-instances]
user@PE2# set vpls100 instance-type vpls
user@PE2# set vpls100 interface ge-1/1/0.100
user@PE2# set vpls100 route-distinguisher 8.0.0.104:100
user@PE2# set vpls100 l2vpn-id l2vpn-id:100:100
user@PE2# set vpls100 vrf-target target:100:100
user@PE2# set vpls100 protocols vpls no-tunnel-services
user@PE2# set vpls200 instance-type vpls
user@PE2# set vpls200 interface ge-1/1/0.200
user@PE2# set vpls200 route-distinguisher 8.0.0.104:200
user@PE2# set vpls200 l2vpn-id l2vpn-id:100:200
user@PE2# set vpls200 vrf-target target:100:208
user@PE2# set vpls200 protocols vpls no-tunnel-services
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@PE2# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, **show routing-options**, and **show routing-instances** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE2# show interfaces
ge-1/1/0 {
  vlan-tagging;
  encapsulation flexible-ethernet-services;
  unit 100 {
    encapsulation vlan-vpls;
    vlan-id 100;
    input-vlan-map pop;
    output-vlan-map push;
    family vpls;
  }
  unit 200 {
    encapsulation vlan-vpls;
    vlan-id 200;
    family vpls;
  }
}
ge-1/2/1 {
```

```

    unit 0 {
        description "PE2 to PE1";
        family inet {
            address 8.0.40.104/24;
        }
        family iso;
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 8.0.0.104/32;
        }
    }
}

user@PE2# show protocols
mpls {
    interface lo0.0;
    interface all;
    interface fxp0 disable;
}
bgp {
    group int {
        type internal;
        local-address 8.0.0.104;
        family l2vpn {
            auto-discovery-only;
        }
        neighbor 8.0.0.107;
    }
}
isis {
    level 1 disable;
    interface ge-1/2/1.0;
    interface lo0.0;
}
ldp {
    interface lo0.0;
    interface all;
    interface fxp0 disable;
}

user@PE2# show routing-options
router-id 8.0.0.104;
autonomous-system 100;

user@PE2# show routing-instances
vpls100 {
    instance-type vpls;
    interface ge-1/1/0.100;
    route-distinguisher 8.0.0.104:100;
    l2vpn-id l2vpn-id:100:100;
    vrf-target target:100:100;
    protocols {
        vpls {

```

```
        no-tunnel-services;
    }
}
vpls200 {
    instance-type vpls;
    interface ge-1/1/0.200;
    route-distinguisher 8.0.0.104:200;
    l2vpn-id l2vpn-id:100:200;
    vrf-target target:100:208;
    protocols {
        vpls {
            no-tunnel-services;
        }
    }
}
```

---

### Device CE2

#### Step-by-Step Procedure

To configure Device CE2:

1. Configure VLAN interfaces.

```
[edit]
user@CE2# edit interfaces ge-1/1/0
[edit interfaces ge-1/1/0]
user@CE2# set vlan-tagging
user@CE2# set mtu 1400
user@CE2# set unit 100 vlan-id 100
user@CE2# set unit 100 family inet address 3.0.100.105/24
user@CE2# set unit 200 vlan-id 200
user@CE2# set unit 200 family inet address 3.0.200.105/24
user@CE2# exit
```

2. Configure OSPF on the interfaces.

```
[edit]
user@CE2# edit protocols ospf area 0.0.0.0
[edit protocols ospf area 0.0.0.0]
user@CE2# set interface ge-1/1/0.100
user@CE2# set interface ge-1/1/0.200
user@CE2# exit
```

3. If you are done configuring the device, commit the configuration.

```
[edit]
user@CE2# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces** and **show protocols** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@CE2# show interfaces
ge-1/1/0 {
    vlan-tagging;
    mtu 1400;
```

```

unit 100 {
    vlan-id 100;
    family inet {
        address 3.0.100.105/24;
    }
}
unit 200 {
    vlan-id 200;
    family inet {
        address 3.0.200.105/24;
    }
}
}

user@CE2# show protocols
ospf {
    area 0.0.0.0 {
        interface ge-1/1/0.100;
        interface ge-1/1/0.200;
    }
}

```

---

## Router RR

### Step-by-Step Procedure

To configure Router RR:

1. Configure interface addresses and the protocol families.

```

[edit]
user@RR# edit interfaces
[edit interfaces]
user@RR# set ge-1/3/2 unit 0 description "RR to PE1"
user@RR# set ge-1/3/2 unit 0 family inet address 8.0.70.107/24
user@RR# set ge-1/3/2 unit 0 family iso
user@RR# set ge-1/3/2 unit 0 family mpls
user@RR# set lo0 unit 0 family inet address 8.0.0.107/32
user@RR# exit

```

2. Configure the autonomous systems and the router ID.

```

[edit]
user@RR# edit routing-options
[edit routing-options]
user@RR# set autonomous-system 100
user@RR# set router-id 8.0.0.107
user@RR# exit

```

3. Configure BGP and set this router to be the route reflector. Route reflection is optional for FEC 129.

```

[edit]
user@RR# edit protocols bgp group int
[edit protocols bgp group int]
user@RR# set type internal
user@RR# set local-address 8.0.0.107
user@RR# set family l2vpn auto-discovery-only
user@RR# set cluster 107.107.107.107

```

```
user@RR# set neighbor 8.0.0.100
user@RR# set neighbor 8.0.0.104
user@RR# exit
```

4. Configure IS-IS for the IGP.

```
[edit]
user@RR# edit protocols isis
[edit protocols isis]
user@RR# set level 1 disable
user@RR# set interface all
user@RR# set interface fxp0.0 disable
user@RR# set interface lo0.0
user@RR# exit
```

5. Configure LDP for the MPLS signaling protocol.

```
[edit]
user@RR# edit protocols ldp
[edit protocols ldp]
user@RR# set interface all
user@RR# set interface fxp0.0 disable
user@RR# set interface lo0.0
user@RR# exit
```

6. If you are done configuring the device, commit the configuration.

```
[edit]
user@RR# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show interfaces**, **show protocols**, and **show routing-options** commands. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@RR# show interfaces
ge-1/3/2 {
  unit 0 {
    description "RR to PE1";
    family inet {
      address 8.0.70.107/24;
    }
    family iso;
    family mpls;
  }
}
lo0 {
  unit 0 {
    family inet {
      address 8.0.0.107/32;
    }
  }
}

user@RR# show protocols
bgp {
  group int {
    type internal;
```



---

```
local-address 8.0.0.107;
family l2vpn {
    auto-discovery-only;
}
cluster 107.107.107.107;
neighbor 8.0.0.100;
neighbor 8.0.0.104;
}
}
isis {
    level 1 disable;
    interface lo0.0;
    interface all;
    interface fxp0 disable;
}
ldp {
    interface lo0.0;
    interface all;
    interface fxp0 disable;
}

user@RR# show routing-options
router-id 8.0.0.107;
autonomous-system 100;
```

## Verification

To verify the operation, use the following commands:

- **show route extensive**
- **show route advertising-protocol *bgp neighbor***
- **show route receive-protocol *bgp neighbor***
- **show route table *bgp.l2vpn.0***
- **show route table *vpls100.l2vpn.0***
- **show route table *vpls200.l2vpn.0***
- **show vpls connections extensive**
- **show vpls mac-table detail**
- **show vpls statistics**

AD in the routing table output indicates autodiscovery NLRI.

### Related Documentation

- [Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups on page 22](#)
- [Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS](#)
- [Virtual Private LAN Service Overview on page 1](#)
- [VPLS Protocol Operation on page 3](#)

## Example: Configuring BGP Autodiscovery for LDP VPLS with User-Defined Mesh Groups

This example describes how to configure user-defined mesh groups for BGP autodiscovery for LDP VPLS, as specified in forwarding equivalency class (FEC) 129. FEC 129 uses BGP autodiscovery to convey endpoint information, so you do not need to manually configure pseudowires. You configure mesh groups on the border router to group the sets of PE routers that are automatically fully meshed and that share the same signaling protocol, either BGP or LDP. You can configure multiple mesh groups to map each fully meshed LDP-signaled or BGP-signaled VPLS domain to a mesh group.

- [Requirements on page 22](#)
- [Overview on page 22](#)
- [Configuration on page 23](#)
- [Verification on page 24](#)

### Requirements

Before you begin, configure BGP autodiscovery for LDP VPLS. See “[Example: Configuring BGP Autodiscovery for LDP VPLS](#)” on page 5.

### Overview

Configuration for a mesh group for FEC 129 is very similar to the mesh-group configuration for FEC 128.

Note the following differences for FEC 129:

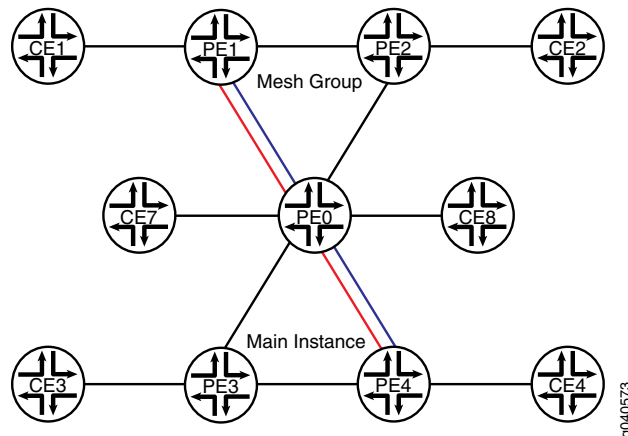
- Each user-defined mesh group must have a unique route distinguisher. Do not use the route distinguisher that is defined for the default mesh group at the **[edit routing-instances]** hierarchy level.
- Each user-defined mesh group must have its own import and export route target.
- Each user-defined mesh group can have a unique Layer 2 VPN ID. By default, all the mesh groups that are configured for a VPLS routing instance use the same Layer 2 VPN ID as the one that you configure at the **[edit routing-instances]** hierarchy level.

### Topology Diagram

---

[Figure 4 on page 23](#) shows a topology that includes a user-defined mesh group.

Figure 4: BGP Autodiscovery for LDP VPLS with a User-Defined Mesh Group



## Configuration

### CLI Quick Configuration

To quickly configure a mesh group, copy the following commands, remove any line breaks, and then paste the commands into the CLI of each device.

```
[edit]
set routing-instances red instance-type vpls
set routing-instances red route-distinguisher 10.10.10.2:3
set routing-instances red l2vpn-id l2vpn-id:100:3
set routing-instances red vrf-target target:100:3
set routing-instances red protocols vpls mesh-group regional-1 route-distinguisher
  10.10.10.2:33
set routing-instances red protocols vpls mesh-group regional-1 vrf-target target:100:33
```

### Step-by-Step Procedure

To configure a mesh group:

1. Set the routing instance type to VPLS.

```
[edit]
user@PE1# set routing-instances red instance-type vpls
```

2. Configure the route distinguisher for the routing instance.

This route distinguisher is used for the default mesh group.

```
[edit]
user@PE1# set routing-instances red route-distinguisher 10.10.10.2:3
```

3. Set the Layer 2 VPN ID for the default mesh group.

```
user@PE1# set routing-instances red l2vpn-id l2vpn-id:100:3
```

4. Set the import and export route target for the default mesh group.

```
user@PE1# set routing-instances red vrf-target target:100:3
```

5. Set the route distinguisher for the user-defined mesh group.

```
user@PE1# set routing-instances red protocols vpls mesh-group regional-1
route-distinguisher 10.10.10.2:33
```

6. Set the import and export route target for the user-defined mesh group.

```
user@PE1# set routing-instances red protocols vpls mesh-group regional-1 vrf-target
target:100:33
```

7. If you are done configuring the device, commit the configuration.

```
[edit]
user@PE1# commit
```

**Results** From configuration mode, confirm your configuration by entering the **show routing-instances** command. If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@PE1# show routing-instances
red {
  instance-type vpls;
  route-distinguisher 10.10.10.2:3;
  l2vpn-id l2vpn-id:100:3;
  vrf-target target:100:3;
  protocols {
    vpls {
      mesh-group regional-1 {
        route-distinguisher 10.10.10.2:33;
        vrf-target target:100:33;
      }
    }
  }
}
```

## Verification

To verify the operation, run the following commands:

- **show route extensive**
- **show route advertising-protocol bgp *neighbor***
- **show route receive-protocol bgp *neighbor***
- **show route table bgp.l2vpn.0**
- **show route table red.l2vpn.0**
- **show vpls connections extensive**
- **show vpls mac-table detail**
- **show vpls statistics**

AD in the routing table output indicates autodiscovery NLRI.

- Related Documentation**
- [Example: Configuring BGP Autodiscovery for LDP VPLS on page 5](#)
  - [Configuring Interoperability Between BGP Signaling and LDP Signaling in VPLS](#)
  - [Virtual Private LAN Service Overview on page 1](#)

- 
- [VPLS Protocol Operation on page 3](#)

