



Multiprotocol Label Switching (MPLS) Configuration Guide, Cisco IOS XE Gibraltar 16.12.x (Catalyst 3650 Switches)

First Published: 2019-07-31

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Configuring Multiprotocol Label Switching (MPLS) 1

- Multiprotocol Label Switching 1
- Finding Feature Information 1
- Restrictions for Multiprotocol Label Switching 1
- Information about Multiprotocol Label Switching 2
 - Functional Description of Multiprotocol Label Switching 2
 - Label Switching Functions 2
 - Distribution of Label Bindings 2
 - MPLS Layer 3 VPN 3
 - Classifying and Marking MPLS QoS EXP 3
- How to Configure Multiprotocol Label Switching 4
 - Configuring a Switch for MPLS Switching 4
 - Configuring a Switch for MPLS Forwarding 5
- Verifying Multiprotocol Label Switching Configuration 6
 - Verifying Configuration of MPLS Switching 6
 - Verifying Configuration of MPLS Forwarding 6
- Additional References for Multiprotocol Label Switching 8
- Feature Information for Multiprotocol Label Switching 9

CHAPTER 2

Configuring eBGP and iBGP Multipath 11

- BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 11
 - Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 11
 - Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 12
- Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN 12
 - Multipath Load Sharing Between eBGP and iBGP 12
 - eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network 13

| | |
|---|----|
| Benefits of Multipath Load Sharing for Both eBGP and iBGP | 14 |
| How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN | 14 |
| Configuring Multipath Load Sharing for Both eBGP and iBGP | 14 |
| Verifying Multipath Load Sharing for Both eBGP and iBGP | 15 |
| Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature | 16 |
| eBGP and iBGP Multipath Load Sharing Configuration Example | 16 |
| Additional References | 17 |
| Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN | 18 |

CHAPTER 3

Configuring EIGRP MPLS VPN PE-CE Site of Origin 19

| | |
|---|----|
| EIGRP MPLS VPN PE-CE Site of Origin | 19 |
| Prerequisites for EIGRP MPLS VPN PE-CE Site of Origin | 19 |
| Restrictions for EIGRP MPLS VPN PE-CE Site of Origin | 19 |
| Information About EIGRP MPLS VPN PE-CE Site of Origin | 20 |
| EIGRP MPLS VPN PE-CE Site of Origin Support Overview | 20 |
| Site of Origin Support for Backdoor Links | 20 |
| Router Interoperation with the Site of Origin Extended Community | 21 |
| Redistribution of BGP VPN Routes That Carry the Site of Origin into EIGRP | 21 |
| Benefits of the EIGRP MPLS VPN PE-CE Site of Origin Support Feature | 21 |
| How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support | 22 |
| Configuring the Site of Origin Extended Community | 22 |
| Verifying the Configuration of the SoO Extended Community | 24 |
| Configuration Examples for EIGRP MPLS VPN PE-CE SoO | 25 |
| Example Configuring the Site of Origin Extended Community | 25 |
| Example Verifying the Site of Origin Extended Community | 25 |
| Additional References | 26 |
| Feature Information for EIGRP MPLS VPN PE-CE Site of Origin | 27 |

CHAPTER 4

Configuring Ethernet-over-MPLS and Pseudowire Redundancy 29

| | |
|-----------------------------|----|
| Finding Feature Information | 29 |
| Configuring EoMPLS | 29 |
| Information About EoMPLS | 29 |
| Prerequisites for EoMPLS | 29 |

| | |
|--|----|
| Restrictions for EoMPLS | 30 |
| Configuring Port-Mode EoMPLS | 30 |
| Xconnect Mode | 31 |
| Protocol CLI Method | 32 |
| Configuration Examples for EoMPLS | 35 |
| Configuring Pseudowire Redundancy | 39 |
| Information About Pseudowire Redundancy | 39 |
| Prerequisites for Pseudowire Redundancy | 39 |
| Restrictions for Pseudowire Redundancy | 39 |
| Configuring Pseudowire Redundancy | 40 |
| Xconnect Mode | 40 |
| Protocol CLI Method | 41 |
| Configuration Examples for Pseudowire Redundancy | 44 |

CHAPTER 5 **Configuring IPv6 Provider Edge over MPLS (6PE)** **47**

| | |
|--------------------------------|----|
| Finding Feature Information | 47 |
| Configuring 6PE | 47 |
| Information About 6PE | 47 |
| Prerequisites for 6PE | 48 |
| Restrictions for 6PE | 48 |
| Configuring 6PE | 48 |
| Configuration Examples for 6PE | 51 |

CHAPTER 6 **Configuring IPv6 VPN Provider Edge over MPLS (6VPE)** **55**

| | |
|---------------------------------|----|
| Configuring 6VPE | 55 |
| Information About 6VPE | 55 |
| Restrictions for 6VPE | 55 |
| Information About 6VPE | 56 |
| Configuration Examples for 6VPE | 56 |
| Finding Feature Information | 60 |

CHAPTER 7 **Configuring IP-aware Netflow for VRF Ingress** **61**

| | |
|--|----|
| Restrictions for IP-aware Netflow for VRF Ingress | 61 |
| Information About IP-aware Netflow for VRF Ingress | 61 |

| | |
|--|----|
| How to Configure IP-aware Netflow for VRF Ingress | 62 |
| Creating a Flow Record | 62 |
| Creating a Flow Exporter | 64 |
| Creating a Flow Monitor | 65 |
| Applying Flow Monitor to an Interface | 66 |
| Configuration Examples for IP-aware Netflow for VRF Ingress | 67 |
| Feature History and Information for IP-aware Netflow for VRF Ingress | 69 |

CHAPTER 8**Configuring Virtual Private LAN Service (VPLS) and VPLS BGP-Based Autodiscovery 71**

| | |
|--|----|
| Finding Feature Information | 71 |
| Configuring VPLS | 71 |
| Information About VPLS | 71 |
| Configuration Examples for VPLS | 73 |
| Restrictions for VPLS | 76 |
| Configuring PE Layer 2 Interfaces to CEs | 76 |
| Configuring 802.1Q Trunks for Tagged Traffic from a CE | 76 |
| Configuring 802.1Q Access Ports for Untagged Traffic from a CE | 78 |
| Configuring Layer 2 VLAN Instances on a PE | 79 |
| Configuring MPLS in the PE | 80 |
| Configuring VFI in the PE | 81 |
| Associating the Attachment Circuit with the VFI at the PE | 82 |
| Configuration Examples for VPLS | 84 |
| Configuring VPLS BGP-based Autodiscovery | 86 |
| Information About VPLS BGP-Based Autodiscovery | 86 |
| Enabling VPLS BGP-based Autodiscovery | 87 |
| Configuring BGP to Enable VPLS Autodiscovery | 88 |
| Configuration Examples for VPLS BGP-AD | 91 |

CHAPTER 9**Configuring MPLS VPN Route Target Rewrite 93**

| | |
|---|----|
| Finding Feature Information | 93 |
| Prerequisites for MPLS VPN Route Target Rewrite | 93 |
| Restrictions for MPLS VPN Route Target Rewrite | 93 |
| Information About MPLS VPN Route Target Rewrite | 94 |
| Route Target Replacement Policy | 94 |

| | |
|---|-----|
| Route Maps and Route Target Replacement | 94 |
| How to Configure MPLS VPN Route Target Rewrite | 95 |
| Configuring a Route Target Replacement Policy | 95 |
| Applying the Route Target Replacement Policy | 98 |
| Associating Route Maps with Specific BGP Neighbors | 98 |
| Verifying the Route Target Replacement Policy | 100 |
| Configuration Examples for MPLS VPN Route Target Rewrite | 101 |
| Examples: Configuring Route Target Replacement Policies | 101 |
| Examples: Applying Route Target Replacement Policies | 102 |
| Examples: Associating Route Maps with Specific BGP Neighbor | 102 |

CHAPTER 10**Configuring Multicast Virtual Private Network 103**

| | |
|--|-----|
| Configuring Multicast VPN | 103 |
| Finding Feature Information | 103 |
| Prerequisites for Configuring Multicast VPN | 103 |
| Restrictions for Configuring Multicast VPN | 103 |
| Information About Configuring Multicast VPN | 104 |
| Multicast VPN Operation | 104 |
| Benefits of Multicast VPN | 104 |
| Multicast VPN Routing and Forwarding and Multicast Domains | 104 |
| Multicast Distribution Trees | 104 |
| Multicast Tunnel Interface | 105 |
| MDT Address Family in BGP for Multicast VPN | 106 |
| How to Configure Multicast VPN | 106 |
| Configuring the Data Multicast Group | 106 |
| Configuring a Default MDT Group for a VRF | 108 |
| Configuring the MDT Address Family in BGP for Multicast VPN | 110 |
| Verifying Information for the MDT Default Group | 112 |
| Configuration Examples for Multicast VPN | 113 |
| Example: Configuring MVPN and SSM | 113 |
| Example: Enabling a VPN for Multicast Routing | 114 |
| Example: Configuring the Multicast Group Address Range for Data MDT Groups | 114 |
| Example: Limiting the Number of Multicast Routes | 114 |
| Additional References for Configuring Multicast VPN | 114 |

Feature Information for Configuring Multicast VPN 115



CHAPTER 1

Configuring Multiprotocol Label Switching (MPLS)

- [Multiprotocol Label Switching, on page 1](#)
- [Finding Feature Information, on page 1](#)
- [Restrictions for Multiprotocol Label Switching, on page 1](#)
- [Information about Multiprotocol Label Switching, on page 2](#)
- [How to Configure Multiprotocol Label Switching, on page 4](#)
- [Verifying Multiprotocol Label Switching Configuration, on page 6](#)
- [Additional References for Multiprotocol Label Switching, on page 8](#)
- [Feature Information for Multiprotocol Label Switching, on page 9](#)

Multiprotocol Label Switching

This module describes Multiprotocol Label Switching and how to configure it on Cisco switches.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Restrictions for Multiprotocol Label Switching

- Multiprotocol Label Switching (MPLS) fragmentation is not supported.
- MPLS maximum transmission unit (MTU) is not supported.
- **ip unnumbered** command is not supported in MPLS configuration.

Information about Multiprotocol Label Switching

Multiprotocol label switching (MPLS) combines the performance and capabilities of Layer 2 (data link layer) switching with the proven scalability of Layer 3 (network layer) routing. MPLS enables you to meet the challenges of explosive growth in network utilization while providing the opportunity to differentiate services without sacrificing the existing network infrastructure. The MPLS architecture is flexible and can be employed in any combination of Layer 2 technologies. MPLS support is offered for all Layer 3 protocols, and scaling is possible well beyond that typically offered in today's networks.

Functional Description of Multiprotocol Label Switching

Label switching is a high-performance packet forwarding technology that integrates the performance and traffic management capabilities of data link layer (Layer 2) switching with the scalability, flexibility, and performance of network layer (Layer 3) routing.

Label Switching Functions

In conventional Layer 3 forwarding mechanisms, as a packet traverses the network, each switch extracts all the information relevant to forwarding the packet from the Layer 3 header. This information is then used as an index for a routing table lookup to determine the next hop for the packet.

In the most common case, the only relevant field in the header is the destination address field, but in some cases, other header fields might also be relevant. As a result, the header analysis must be done independently at each switch through which the packet passes. In addition, a complicated table lookup must also be done at each switch.

In label switching, the analysis of the Layer 3 header is done only once. The Layer 3 header is then mapped into a fixed length, unstructured value called a *label*.

Many different headers can map to the same label, as long as those headers always result in the same choice of next hop. In effect, a label represents a *forwarding equivalence class*—that is, a set of packets which, however different they may be, are indistinguishable by the forwarding function.

The initial choice of a label need not be based exclusively on the contents of the Layer 3 packet header; for example, forwarding decisions at subsequent hops can also be based on routing policy.

Once a label is assigned, a short label header is added at the front of the Layer 3 packet. This header is carried across the network as part of the packet. At subsequent hops through each MPLS switch in the network, labels are swapped and forwarding decisions are made by means of MPLS forwarding table lookup for the label carried in the packet header. Hence, the packet header does not need to be reevaluated during packet transit through the network. Because the label is of fixed length and unstructured, the MPLS forwarding table lookup process is both straightforward and fast.

Distribution of Label Bindings

Each label switching router (LSR) in the network makes an independent, local decision as to which label value to use to represent a forwarding equivalence class. This association is known as a label binding. Each LSR informs its neighbors of the label bindings it has made. This awareness of label bindings by neighboring switches is facilitated by the following protocols:

- Label Distribution Protocol (LDP)--enables peer LSRs in an MPLS network to exchange label binding information for supporting hop-by-hop forwarding in an MPLS network
- Border Gateway Protocol (BGP)--Used to support MPLS virtual private networks (VPNs)

When a labeled packet is being sent from LSR A to the neighboring LSR B, the label value carried by the IP packet is the label value that LSR B assigned to represent the forwarding equivalence class of the packet. Thus, the label value changes as the IP packet traverses the network.

For more information about LDP configuration, see the see MPLS: LDP Configuration Guide at http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mpls/config_library/xe-3s/mp-xe-3s-library.html



Note As the scale of label entries is limited in, especially with ECMP, it is recommended to enable LDP label filtering. LDP labels shall be allocated only for well known prefixes like loopback interfaces of routers and any prefix that needs to be reachable in the global routing table.

MPLS Layer 3 VPN

A Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) consists of a set of sites that are interconnected by means of an MPLS provider core network. At each customer site, one or more customer edge (CE) routers attach to one or more provider edge (PE) routers.

Before configuring MPLS Layer 3 VPNs, you should have MPLS, Label Distribution Protocol (LDP), and Cisco Express Forwarding (CEF) installed in your network. All routers in the core, including the PE routers, must be able to support CEF and MPLS forwarding.

Classifying and Marking MPLS QoS EXP

The QoS EXP Matching feature allows you to classify and mark network traffic by modifying the Multiprotocol Label Switching (MPLS) experimental bits (EXP) field in IP packets.

The QoS EXP Matching feature allows you to organize network traffic by setting values for the MPLS EXP field in MPLS packets. By choosing different values for the MPLS EXP field, you can mark packets so that packets have the priority that they require during periods of congestion. Setting the MPLS EXP value allows you to:

- **Classify traffic:** The classification process selects the traffic to be marked. Classification accomplishes this by partitioning traffic into multiple priority levels, or classes of service. Traffic classification is the primary component of class-based QoS provisioning.
- **Police and mark traffic:** Policing causes traffic that exceeds the configured rate to be discarded or marked to a different drop level. Marking traffic is a way to identify packet flows to differentiate them. Packet marking allows you to partition your network into multiple priority levels or classes of service.

Restrictions

Following is the list of restrictions for classifying and marking MPLS QoS EXP:

- Only Uniform mode and Pipe mode are supported; Short-pipe mode is not supported.
- Support range of QoS-group values range between 0 and 30. (Total 31 QoS-groups).

- EXP marking using QoS policy is supported only on the outer label; inner EXP marking is not supported.

How to Configure Multiprotocol Label Switching

This section explains how to perform the basic configuration required to prepare a switch for MPLS switching and forwarding.

Configuring a Switch for MPLS Switching

MPLS switching on Cisco switches requires that Cisco Express Forwarding be enabled.



Note `ip unnumbered` command is not supported in MPLS configuration.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ip cef distributed`
4. `mpls label range minimum-value maximum-value`
5. `mpls label protocol ldp`

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | enable Example: Device> <code>enable</code> | Enables privileged EXEC mode. Enter your password, if prompted. |
| Step 2 | configure terminal Example: Device# <code>configure terminal</code> | Enters global configuration mode. |
| Step 3 | ip cef distributed Example: Device(config)# <code>ip cef distributed</code> | Enables Cisco Express Forwarding on the switch. |
| Step 4 | mpls label range <i>minimum-value maximum-value</i> Example: Device(config)# <code>mpls label range 16 4096</code> | Configure the range of local labels available for use with MPLS applications on packet interfaces. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 5 | mpls label protocol ldp Example: Device(config)# mpls label protocol ldp | Specifies the label distribution protocol for the platform. |

Configuring a Switch for MPLS Forwarding

MPLS forwarding on Cisco switches requires that forwarding of IPv4 packets be enabled.



Note **ip unnumbered** command is not supported in MPLS configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/subslot lport*
4. **mpls ip**
5. **mpls label protocol ldp**
6. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. Enter your password, if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>type slot/subslot lport</i> Example: Device(config)# interface gigabitethernet 1/0/0 | Specifies the Gigabit Ethernet interface and enters interface configuration mode. For Switch Virtual Interface (SVI), the example is Device(config)# interface vlan 1000 |
| Step 4 | mpls ip Example: Device(config-if)# mpls ip | Enables MPLS forwarding of IPv4 packets along routed physical interfaces (Gigabit Ethernet), Switch Virtual Interface (SVI), or port channels. |

| | Command or Action | Purpose |
|---------------|---|--|
| Step 5 | mpls label protocol ldp Example: Device(config-if)# mpls label protocol ldp | Specifies the label distribution protocol for an interface. Note MPLS LDP cannot be enabled on a Virtual Routing and Forwarding (VRF) interface. |
| Step 6 | end Example: Device(config-if)# end | Exits interface configuration mode and returns to privileged EXEC mode. |

Verifying Multiprotocol Label Switching Configuration

This section explains how to verify successful configuration of MPLS switching and forwarding.

Verifying Configuration of MPLS Switching

To verify that Cisco Express Forwarding has been configured properly, issue the **show ip cef summary** command, which generates output similar to that shown below:

SUMMARY STEPS

1. **show ip cef summary**

DETAILED STEPS

show ip cef summary

Example:

```
Switch# show ip cef summary

IPv4 CEF is enabled for distributed and running
VRF Default
 150 prefixes (149/1 fwd/non-fwd)
  Table id 0x0
  Database epoch:      4 (150 entries at this epoch)
Switch#
```

Verifying Configuration of MPLS Forwarding

To verify that MPLS forwarding has been configured properly, issue the **show mpls interfaces detail** command, which generates output similar to that shown below:



Note The MPLS MTU value is equivalent to the IP MTU value of the port or switch by default. MTU configuration for MPLS is not supported.

SUMMARY STEPS

1. **show mpls interfaces detail**
2. **show running-config interface**
3. **show mpls forwarding**

DETAILED STEPS

Step 1 show mpls interfaces detail

Example:

For physical (Gigabit Ethernet) interface:

```
Switch# show mpls interfaces detail interface GigabitEthernet 1/0/0
```

```
Type Unknown
IP labeling enabled
LSP Tunnel labeling not enabled
IP FRR labeling not enabled
BGP labeling not enabled
MPLS not operational
MTU = 1500
```

For Switch Virtual Interface (SVI):

```
Switch# show mpls interfaces detail interface Vlan1000
```

```
Type Unknown
IP labeling enabled (ldp) :
  Interface config
LSP Tunnel labeling not enabled
IP FRR labeling not enabled
BGP labeling not enabled
MPLS operational
MTU = 1500
```

Step 2 show running-config interface

Example:

For physical (Gigabit Ethernet) interface:

```
Switch# show running-config interface interface GigabitEthernet 1/0/0
```

Building configuration...

```
Current configuration : 307 bytes
!
interface TenGigabitEthernet1/0/0
no switchport
ip address xx.xx.x.x xxx.xxx.xxx.x
mpls ip
mpls label protocol ldp
```

```
end
```

For Switch Virtual Interface (SVI):

```
Switch# show running-config interface interface Vlan1000
```

```
Building configuration...
```

```
Current configuration : 187 bytes
!
interface Vlan1000
ip address xx.xx.x.x xxx.xxx.xxx.x
mpls ip
mpls label protocol ldp
end
```

Step 3 show mpls forwarding

Example:

For physical (Gigabit Ethernet) interface:

```
Switch#show mpls forwarding-table
Local      Outgoing  Prefix          Bytes Label  Outgoing  Next Hop
Label      Label     or Tunnel Id   Switched     interface
500        No Label  12ckt(3)       0            Gi3/0/22  point2point
501        No Label  12ckt(1)       12310411816789 none       point2point
502        No Label  12ckt(2)       0            none      point2point
503        566      15.15.15.15/32 0            Po5       192.1.1.2
504        530      7.7.7.7/32     538728528   Po5       192.1.1.2
505        573      6.6.6.10/32   0            Po5       192.1.1.2
506        606      6.6.6.6/32    0            Po5       192.1.1.2
507        explicit-n 1.1.1.1/32    0            Po5       192.1.1.2
556        543      19.10.1.0/24  0            Po5       192.1.1.2
567        568      20.1.1.0/24   0            Po5       192.1.1.2
568        574      21.1.1.0/24   0            Po5       192.1.1.2
574        No Label  213.1.1.0/24[V] 0            aggregate/vpn113
575        No Label  213.1.2.0/24[V] 0            aggregate/vpn114
576        No Label  213.1.3.0/24[V] 0            aggregate/vpn115
577        No Label  213:1:1::/64   0            aggregate
594        502      103.1.1.0/24  0            Po5       192.1.1.2
595        509      31.1.1.0/24   0            Po5       192.1.1.2
596        539      15.15.1.0/24  0            Po5       192.1.1.2
597        550      14.14.1.0/24  0            Po5       192.1.1.2
633        614      2.2.2.0/24    0            Po5       192.1.1.2
634        577      90.90.90.90/32 873684     Po5       192.1.1.2
635        608      154.1.1.0/24  0            Po5       192.1.1.2
636        609      153.1.1.0/24  0            Po5       192.1.1.2
Switch#
end
```

Additional References for Multiprotocol Label Switching

Related Documents

| Related Topic | Document Title |
|--|----------------|
| For complete syntax and usage information for the commands used in this chapter. | |

Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for Multiprotocol Label Switching

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Multiprotocol Label Switching

| Release | Modification |
|---------------------------------------|------------------------------|
| Cisco IOS XE 3.3SE Cisco IOS XE 3.3SE | This feature was introduced. |



CHAPTER 2

Configuring eBGP and iBGP Multipath

- [BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, on page 11](#)
- [Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, on page 12](#)
- [How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, on page 14](#)
- [Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature, on page 16](#)
- [Additional References, on page 17](#)
- [Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN, on page 18](#)

BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The BGP Multipath Load Sharing for eBGP and iBGP feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides improved load balancing deployment and service offering capabilities and is useful for multi-homed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks.

Prerequisites for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Cisco Express Forwarding (CEF) or distributed CEF (dCEF) must be enabled on all participating devices.

Restrictions for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Address Family Support

This feature is configured on a per VPN routing and forwarding instance (VRF) basis. This feature can be configured under both IPv4 and IPv6 VRF address families.

Memory Consumption Restriction

Each BGP multipath routing table entry will use additional memory. We recommend that you do not use this feature on a device with a low amount of available memory and especially if the device carries full Internet routing tables.

Number of Paths Limitation

The number of paths supported are limited to 2 BGP multipaths. This could either be 2 iBGP multipaths or 1 iBGP multipath and 1 eBGP multipath.

Unsupported Commands

`ip unnumbered` command is not supported in MPLS configuration.

Information About BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

Multipath Load Sharing Between eBGP and iBGP

A BGP routing process will install a single path as the best path in the routing information base (RIB) by default. The `maximum-paths` command allows you to configure BGP to install multiple paths in the RIB for multipath load sharing. BGP uses the best path algorithm to select a single multipath as the best path and advertise the best path to BGP peers.



Note

The number of paths of multipaths that can be configured is documented on the `maximum-paths` command reference page.

Load balancing over the multipaths is performed by CEF. CEF load balancing is configured on a per-packet round robin or on a per session (source and destination pair) basis. For information about CEF, refer to Cisco IOS IP Switching Configuration Guide documentation: [IP Switching Cisco Express Forwarding Configuration Guide](#). The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature is enabled under the IPv4 VRF address family and IPv6 VRF address family configuration modes. When enabled, this feature can perform load balancing on eBGP and/or iBGP paths that are imported into the VRF. The number of multipaths is configured on a per VRF basis. Separate VRF multipath configurations are isolated by unique route distinguisher.

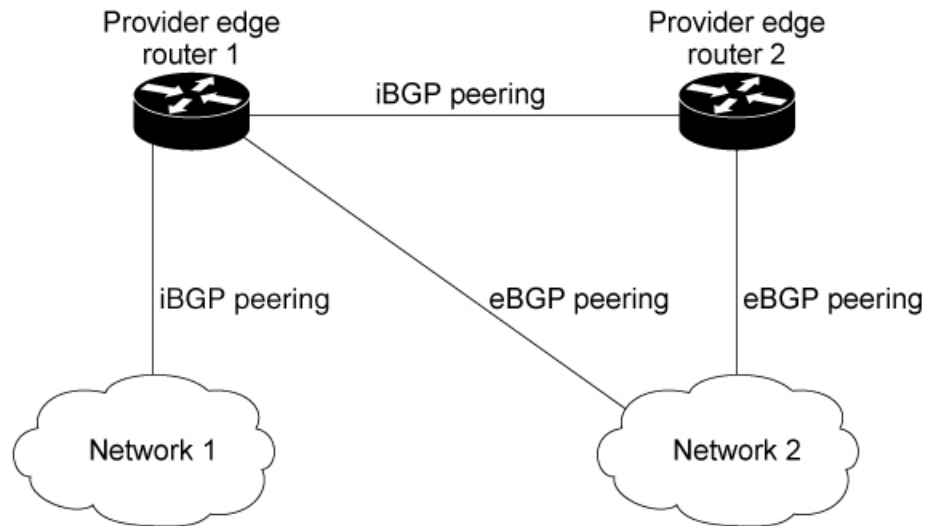


Note The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature operates within the parameters of configured outbound routing policy.

eBGP and iBGP Multipath Load Sharing in a BGP MPLS Network

The following figure shows a service provider BGP MPLS network that connects two remote networks to PE router 1 and PE router 2. PE router 1 and PE router 2 are both configured for VPNv4 unicast iBGP peering. Network 2 is a multihomed network that is connected to PE router 1 and PE router 2. Network 2 also has extranet VPN services configured with Network 1. Both Network 1 and Network 2 are configured for eBGP peering with the PE routers.

Figure 1: Service Provider BGP MPLS Network



PE router 1 can be configured with the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature so that both iBGP and eBGP paths can be selected as multipaths and imported into the VRF . The multipaths will be used by CEF to perform load balancing. IP traffic that is sent from Network 1 to Network 2, PE router 1 will Load Share with eBGP paths as IP traffic & iBGP path will be sent as MPLS traffic.



- Note**
- eBGP session between local CE & local PE is not supported .
 - eBGP session from a local PE to a remote CE is supported.
 - eiBGP Multipath is supported in per prefix label allocation mode only. It is not supported in other label allocation modes.

Benefits of Multipath Load Sharing for Both eBGP and iBGP

The BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS VPN feature allows multihomed autonomous systems and PE routers to be configured to distribute traffic across both eBGP and iBGP paths.

How to Configure BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

This section contains the following procedures:

Configuring Multipath Load Sharing for Both eBGP and iBGP

SUMMARY STEPS

1. **enable**
2. **configure** { **terminal** | **memory** | **network** }
3. **router bgp as-number**
4. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name* }
5. **address-family ipv4 vrfvrf-name**
6. **address-family ipv6 vrfvrf-name**
7. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name* } **update-source** *interface-type interface-name*
8. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name* } **activate**
9. **maximum-paths eibgp number** [*import number*]

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted. |
| Step 2 | configure { terminal memory network } Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | router bgp as-number Example: Device(config)# router bgp 40000 | Enters router configuration mode to create or configure a BGP routing process. |
| Step 4 | neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } Example: Device(config-router)# neighbor group192 | Accepts and attempts BGP connections to external peers residing on networks that are not directly connected. |
| Step 5 | address-family ipv4 vrfvrf-name | Places the router in address family configuration mode. |

| | Command or Action | Purpose |
|---------------|--|---|
| | Example: Device(config-router)# address-family ipv4 vrf RED | <ul style="list-style-type: none"> Separate VRF multipath configurations are isolated by unique route distinguisher. |
| Step 6 | address-family ipv6 vrf vrf-name Example: Device(config-router)# address-family ipv6 vrf RED | Places the router in address family configuration mode. <ul style="list-style-type: none"> Separate VRF multipath configurations are isolated by unique route distinguisher. |
| Step 7 | neighbor {ip-address ipv6-address peer-group-name } update-source interface-type interface-name Example: Device(config-router)# neighbor FE80::1234:BFF:FE0E:A471 update-source GigabitEthernet 1/0/0 | Specifies the link-local address over which the peering is to occur. |
| Step 8 | neighbor {ip-address ipv6-address peer-group-name } activate Example: Device(config-router)# neighbor group192 activate | Activates the neighbor or listen range peer group for the configured address family. |
| Step 9 | maximum-paths eibgp number [import number] Example: Device(config-router-af)# maximum-paths eibgp 2 | Configures the number of parallel iBGP and eBGP routes that can be installed into a routing table. |

Verifying Multipath Load Sharing for Both eBGP and iBGP

SUMMARY STEPS

- enable
- show ip bgp neighbors
- show ip bgp vpnv4 vrf vrf name
- show ip route vrf vrf-name

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | show ip bgp neighbors Example: Device# show ip bgp neighbors | Displays information about the TCP and BGP connections to neighbors. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 3 | show ip bgp vpnv4 vrf <i>vrf name</i> Example: Device# show ip bgp vpnv4 vrf RED | Displays VPN address information from the BGP table. This command is used to verify that the VRF has been received by BGP. |
| Step 4 | show ip route vrf <i>vrf-name</i> Example: Device# show ip route vrf RED | Displays the IP routing table associated with a VRF instance. The show ip route vrf command is used to verify that the VRF is in the routing table. |

Example

What to do next

.

Configuration Examples for the BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN Feature

The following examples show how to configure and verify this feature:

eBGP and iBGP Multipath Load Sharing Configuration Example

This following configuration example configures a router in IPv4 address-family mode to select two BGP routes (eBGP or iBGP) as multipaths:

```
Device router bgp 40000
  Deviceaddress-family ipv4 vrf RED
  Devicemaximum-paths eibgp 2
Deviceend
```

This following configuration example configures a router in IPv6 address-family mode to select two BGP routes (eBGP or iBGP) as multipaths:

```
Device router bgp 40000
  Deviceaddress-family ipv6 vrf RED
  Devicemaximum-paths eibgp 2
Deviceend
```


Additional References

Related Documents

Table 2: Related Documents

| Related Topic | Document Title |
|--|---|
| BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | • Cisco IOS IP Command Reference, Volume 2 of 4: Routing Protocols, Release 12.3T |
| BGP configuration tasks | • Cisco IOS IP Configuration Guide, Release 12.3 |
| Comprehensive BGP link bandwidth configuration examples and tasks | • BGP Link Bandwidth |
| CEF configuration tasks | • Cisco IOS Switching Services Configuration Guide |

Table 3: Standards

| Standard | Title |
|--|-------|
| No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature | -- |

Table 4: RFCs

| RFC | Title |
|----------|------------------------------------|
| RFC 1771 | A Border Gateway Protocol 4 (BGP4) |
| RFC 2547 | BGP/MPLS VPNs |
| RFC 2858 | Multiprotocol Extensions for BGP-4 |

Table 5: Technical Assistance

| Description | Link |
|---|---|
| Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, tools, and lots more. Registered Cisco.com users can log in from this page to access even more content. | http://www.cisco.com/public/support/tac/home.shtml |

Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN

| Feature Name | Releases | Feature Information |
|--|------------------|---|
| BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN | Cisco IOS 16.6.1 | The BGP Multipath Load Sharing for eBGP and iBGP feature allows you to configure multipath load balancing with both external BGP (eBGP) and internal BGP (iBGP) paths in Border Gateway Protocol (BGP) networks that are configured to use Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs). This feature provides improved load balancing deployment and service offering capabilities and is useful for multi-homed autonomous systems and Provider Edge (PE) routers that import both eBGP and iBGP paths from multihomed and stub networks. |



CHAPTER 3

Configuring EIGRP MPLS VPN PE-CE Site of Origin

- [EIGRP MPLS VPN PE-CE Site of Origin, on page 19](#)
- [Information About EIGRP MPLS VPN PE-CE Site of Origin, on page 20](#)
- [How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support, on page 22](#)
- [Configuration Examples for EIGRP MPLS VPN PE-CE SoO, on page 25](#)
- [Additional References, on page 26](#)
- [Feature Information for EIGRP MPLS VPN PE-CE Site of Origin, on page 27](#)

EIGRP MPLS VPN PE-CE Site of Origin

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies. This feature is designed to support the MPLS VPN Support for EIGRP Between Provider Edge (PE) and Customer Edge (CE) feature. Support for backdoor links is provided by this feature when installed on PE routers that support EIGRP MPLS VPNs.

Prerequisites for EIGRP MPLS VPN PE-CE Site of Origin

This document assumes that Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone). The following tasks will also need to be completed before you can configure this feature:

- This feature was introduced to support the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature and should be configured after the EIGRP MPLS VPN is created.
- All PE routers that are configured to support the EIGRP MPLS VPN must run Cisco IOS XE Release 2.1 or a later release, which provides support for the SoO extended community

Restrictions for EIGRP MPLS VPN PE-CE Site of Origin

- If a VPN site is partitioned and the SoO extended community attribute is configured on a backdoor router interface, the backdoor link cannot be used as an alternate path to reach prefixes originated in other partitions of the same site

- A unique SoO value must be configured for each individual VPN site. The same value must be configured on all provider edge and customer edge interfaces (if SoO is configured on the CE routers) that support the same VPN site.
- **ip unnumbered** command is not supported in MPLS configuration.

Information About EIGRP MPLS VPN PE-CE Site of Origin

EIGRP MPLS VPN PE-CE Site of Origin Support Overview

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces SoO support for EIGRP-to-BGP and BGP-to-EIGRP redistribution. The SoO extended community is a BGP extended community attribute that is used to identify routes that have originated from a site so that the readvertisement of that prefix back to the source site can be prevented. The SoO extended community uniquely identifies the site from which a PE router has learned a route. SoO support provides the capability to filter MPLS VPN traffic on a per-EIGRP-site basis. SoO filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent routing loops from occurring in complex and mixed network topologies, such as EIGRP VPN sites that contain both VPN and backdoor links.

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Site of Origin Support for Backdoor Links

The EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature introduces support for backdoor links. A backdoor link or a route is a connection that is configured outside of the VPN between a remote and main site; for example, a WAN leased line that connects a remote site to the corporate network. Backdoor links are typically used as back up routes between EIGRP sites if the VPN link is down or not available. A metric is set on the backdoor link so that the route through the backdoor router is not selected unless there is a VPN link failure.

The SoO extended community is defined on the interface of the backdoor router. It identifies the local site ID, which should match the value that is used on the PE routers that support the same site. When the backdoor router receives an EIGRP update (or reply) from a neighbor across the backdoor link, the router checks the update for an SoO value. If the SoO value in the EIGRP update matches the SoO value on the local backdoor interface, the route is rejected and not added to the EIGRP topology table. This scenario typically occurs when the route with the local SoO value in the received EIGRP update was learned by the other VPN site and then advertised through the backdoor link by the backdoor router in the other VPN site. SoO filtering on the backdoor link prevents transient routing loops from occurring by filtering out EIGRP updates that contain routes that carry the local site ID.

If this feature is enabled on the PE routers and the backdoor routers in the customer sites, and SoO values are defined on both the PE and backdoor routers, both the PE and backdoor routers will support convergence between the VPN sites. The other routers in the customer sites need only propagate the SoO values carried by the routes, as the routes are forwarded to neighbors. These routers do not otherwise affect or support convergence beyond normal Diffusing Update Algorithm (DUAL) computations.

Router Interoperation with the Site of Origin Extended Community

The configuration of an SoO extended community allows routers that support EIGRP MPLS VPN PE-CE Site of Origin feature to identify the site from which each route originated. When this feature is enabled, the EIGRP routing process on the PE or CE router checks each received route for the SoO extended community and filters based on the following conditions:

- A received route from BGP or a CE router contains an SoO value that matches the SoO value on the receiving interface : If a route is received with an associated SoO value that matches the SoO value that is configured on the receiving interface, the route is filtered because it was learned from another PE router or from a backdoor link. This behavior is designed to prevent routing loops.
- A received route from a CE router is configured with an SoO value that does not match: If a route is received with an associated SoO value that does not match the SoO value that is configured on the receiving interface, the route is added to the EIGRP topology table so that it can be redistributed into BGP. If the route is already installed to the EIGRP topology table but is associated with a different SoO value, the SoO value from the topology table will be used when the route is redistributed into BGP.
- A received route from a CE router does not contain an SoO value: If a route is received without a SoO value, the route is accepted into the EIGRP topology table, and the SoO value from the interface that is used to reach the next hop CE router is appended to the route before it is redistributed into BGP.

When BGP and EIGRP peers that support the SoO extended community receive these routes, they will also receive the associated SoO values and pass them to other BGP and EIGRP peers that support the SoO extended community. This filtering is designed to prevent transient routes from being relearned from the originating site, which prevents transient routing loops from occurring.

Redistribution of BGP VPN Routes That Carry the Site of Origin into EIGRP

When an EIGRP routing process on a PE router redistributes BGP VPN routes into an EIGRP topology table, EIGRP extracts the SoO value (if one is present) from the appended BGP extended community attributes and appends the SoO value to the route before adding it to the EIGRP topology table. EIGRP tests the SoO value for each route before sending updates to CE routers. Routes that are associated with SoO values that match the SoO value configured on the interface are filtered out before they are passed to the CE routers. When an EIGRP routing process receives routes that are associated with different SoO values, the SoO value is passed to the CE router and carried through the CE site.

Benefits of the EIGRP MPLS VPN PE-CE Site of Origin Support Feature

The configuration of the EIGRP MPLS VPN PE-CE Site of Origin Support feature introduces per-site VPN filtering, which improves support for complex topologies, such as MPLS VPNs with backdoor links, CE routers that are dual-homed to different PE routers, and PE routers that support CE routers from different sites within the same virtual routing and forwarding (VRF) instance.

How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support

Configuring the Site of Origin Extended Community

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Before you begin

- Confirm that the Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone).
- Configure an EIGRP MPLS VPN before configuring this feature.
- All PE routers that are configured to support the EIGRP MPLS VPN must support the SoO extended community.
- A unique SoO value must be configured for each VPN site. The same value must be used on the interface of the PE router that connects to the CE router for each VPN site.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-name* { **permit** | **deny** } [*sequence-number*]
4. **set extcommunity** *sooextended-community-value*
5. **exit**
6. **interface** *type number*
7. **no switchport**
8. **vrf forwarding** *vrf-name*
9. **ip vrf sitemap** *route-map-name*
10. **ip address** *ip-address subnet-mask*
11. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. • Enter your password if prompted. |
| Step 2 | configure terminal Example: | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|--|---|
| | Device# <code>configure terminal</code> | |
| Step 3 | route-map <i>map-name</i> { permit deny } [<i>sequence-number</i>] Example: Device(config)# <code>route-map Site-of-Origin permit 10</code> | Enters route-map configuration mode and creates a route map. <ul style="list-style-type: none"> • The route map is created in this step so that SoO extended community can be applied. |
| Step 4 | set extcommunity soo <i>extended-community-value</i> Example: Device(config-route-map)# <code>set extcommunity soo 100:1</code> | Sets BGP extended community attributes. <ul style="list-style-type: none"> • The soo keyword specifies the site of origin extended community attribute. • The extended-community-value argument specifies the value to be set. The value can be one of the following formats: <ul style="list-style-type: none"> • autonomous-system-number: network-number • ip-address: network-number The colon is used to separate the autonomous system number and network number or IP address and network number. |
| Step 5 | exit Example: Device(config-route-map)# <code>exit</code> | Exits route-map configuration mode and enters global configuration mode. |
| Step 6 | interface <i>type number</i> Example: Device(config)# <code>interface GigabitEthernet 1/0/1</code> | Enters interface configuration mode to configure the specified interface. |
| Step 7 | no switchport Example: Device(config-if)# <code>no switchport</code> | causes the interface to cease operating as a Layer 2 port and become a Cisco-routed (Layer 3) port: |
| Step 8 | vrf forwarding <i>vrf-name</i> Example: Device(config-if)# <code>ip vrf forwarding VRF1</code> | Associates the VRF with an interface or subinterface. <ul style="list-style-type: none"> • The VRF name configured in this step should match the VRF name created for the EIGRP MPLS VPN with the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature. |
| Step 9 | ip vrf sitemap <i>route-map-name</i> Example: Device(config-if)# <code>ip vrf sitemap Site-of-Origin</code> | Associates the VRF with an interface or subinterface. <ul style="list-style-type: none"> • The route map name configured in this step should match the route map name created to apply the SoO extended community in Step 3. |

| | Command or Action | Purpose |
|----------------|--|---|
| Step 10 | ip address <i>ip-address subnet-mask</i> Example: Device(config-if)# ip address 10.0.0.1 255.255.255.255 | Configures the IP address for the interface. <ul style="list-style-type: none"> • The IP address needs to be reconfigured after enabling VRF forwarding. |
| Step 11 | end Example: Device(config-if)# end | Exits interface configuration mode and enters privileged EXEC mode. |

Example

What to do next

- For mixed EIGRP MPLS VPN network topologies that contain backdoor routes, the next task is to configure the “prebest path” cost community for backdoor routes.

Verifying the Configuration of the SoO Extended Community

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** { **all** | *rdroute-distinguisher* | *vrfvrf-name* } [*ip-prefix/length*]

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | show ip bgp vpnv4 { all <i>rdroute-distinguisher</i> <i>vrfvrf-name</i> } [<i>ip-prefix/length</i>] Example: Device# ip bgp vpnv4 vrf SOO-1 20.2.1.1/32 | Displays VPN address information from the BGP table. <ul style="list-style-type: none"> • Use the show ip bgp vpnv4 command with the all keyword to verify that the specified route has been configured with the SoO extended community attribute. |

Configuration Examples for EIGRP MPLS VPN PE-CE SoO

Example Configuring the Site of Origin Extended Community

The following example, beginning in global configuration mode, configures SoO extended community on an interface:

```

route-map Site-of-Origin permit 10
 set extcommunity soo 100:1
exit
GigabitEthernet1/0/1
 ip vrf forwarding RED
 ip vrf sitemap Site-of-Origin
 ip address 10.0.0.1 255.255.255.255
end

```

Example Verifying the Site of Origin Extended Community

The following example shows VPN address information from the BGP table and verifies the configuration of the SoO extended community:

```

switch# show ip bgp vpnv4 all 10.0.0.1
 BGP routing table entry for 100:1:10.0.0.1/32, version 6
 Paths: (1 available, best #1, no table)
 Advertised to update-groups:
 1
 100 300
 192.168.0.2 from 192.168.0.2 (172.16.13.13)
 Origin incomplete, localpref 100, valid, external, best
 Extended Community: SOO:100:1

```

Show command Customer Edge Device

```

CE1#show ip eigrp topo 20.2.1.1/32
EIGRP-IPv4 Topology Entry for AS(30)/ID(30.0.0.1) for 20.2.1.1/32
 State is Passive, Query origin flag is 1, 2 Successor(s), FD is 131072
 Descriptor Blocks:
 31.1.1.2 (GigabitEthernet1/0/13), from 31.1.1.2, Send flag is 0x0
 Composite metric is (131072/130816), route is External
 Vector metric:
 Minimum bandwidth is 1000000 Kbit
 Total delay is 5020 microseconds
 Reliability is 255/255
 Load is 1/255
 Minimum MTU is 1500
 Hop count is 2
 Originating router is 30.0.0.2
 Extended Community: SoO:100:1
 External data:
 AS number of route is 0
 External protocol is Connected, external metric is 0
 Administrator tag is 0 (0x00000000)

```

Show command Provider Edge Device

```

PE2#show ip eigrp vrf SOO-1 topology 31.1.1.0/24
EIGRP-IPv4 VR(L3VPN) Topology Entry for AS(30)/ID(2.2.2.22)
      Topology(base) TID(0) VRF(SOO-1)
EIGRP-IPv4(30): Topology base(0) entry for 31.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1310720
  Descriptor Blocks:
  1.1.1.1, from VPNv4 Sourced, Send flag is 0x0
    Composite metric is (1310720/0), route is Internal (VPNv4 Sourced)
    Vector metric:
      Minimum bandwidth is 1000000 Kbit
      Total delay is 10000000 picoseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 0
      Originating router is 1.1.1.11
    Extended Community: SoO:100:1

```

Additional References

Related Documents

| Related Topic | Document Title |
|---------------------------|---|
| CEF commands | Cisco IOS IP Switching Command Reference |
| CEF configuration tasks | Cisco Express Forwarding Overview module of the Cisco IOS IP Switching Configuration Guide |
| EIGRP commands | Cisco IOS IP Routing: EIGRP Command Reference |
| EIGRP configuration tasks | Configuring EIGRP |
| MPLS VPNs | MPLS Layer 3 VPNs module of the Cisco IOS Multiprotocol Label Switching Configuration Guide |

Table 7: Standards

| Standard | Title |
|----------|-------|
| None | -- |

Table 8: MIBs

| MIB | MIBs Link |
|------|---|
| None | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://mibs.cloudapps.cisco.com/ITDIT/MIBS/servlet/index |

Table 9: RFCs

| RFC | Title |
|------|-------|
| None | -- |

Table 10: Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | https://www.cisco.com/c/en/us/support/index.html |

Feature Information for EIGRP MPLS VPN PE-CE Site of Origin

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for EIGRP MPLS VPN PE-CE Site of Origin

| Feature Name | Releases | Feature Information |
|-------------------------------------|------------------|---|
| EIGRP MPLS VPN PE-CE Site of Origin | Cisco IOS 16.6.1 | The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies. |



CHAPTER 4

Configuring Ethernet-over-MPLS and Pseudowire Redundancy

- [Finding Feature Information, on page 29](#)
- [Configuring EoMPLS, on page 29](#)
- [Configuring Pseudowire Redundancy, on page 39](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Configuring EoMPLS

Information About EoMPLS

EoMPLS is one of the AToM transport types. EoMPLS works by encapsulating Ethernet PDUs in MPLS packets and forwarding them across the MPLS network. Each PDU is transported as a single packet.

Only the following mode is supported:

- Port mode—Allows all traffic on a port to share a single VC across an MPLS network. Port mode uses VC type 5.

For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Prerequisites for EoMPLS

Before you configure EoMPLS, ensure that the network is configured as follows:

- Configure IP routing in the core so that the PE routers can reach each other through IP.
- Configure MPLS in the core so that a label switched path (LSP) exists between the PE routers.
- Configure **no switchport**, **no keepalive** and **no ip address** before configuring xconnect on the attachment circuit.
- For load-balancing, **port-channel load-balance** command is mandatory to be configured.

Restrictions for EoMPLS

- VLAN mode is not supported. Ethernet Flow Point is not supported.
- QoS : Customer DSCP Re-marking is not supported with VPWS and EoMPLS.
- VCCV Ping with explicit null is not supported.
- L2 VPN Interworking is not supported.
- L2 Protocol Tunneling CLI is not supported.
- Untagged, tagged and 802.1Q in 802.1Q are supported as incoming traffic.



Note Flow Load balance for 802.1Q in 802.1Q over EoMPLS is not supported.

- Flow Aware Transport Pseudowire Redundancy (FAT PW) is supported only in Protocol-CLI mode. Supported load balancing parameters are Source IP, Source MAC address, Destination IP and Destination MAC address.
- Enabling or disabling Control word is supported.
- MPLS QoS is supported in Pipe and Uniform Mode. Default mode is Pipe Mode.
- Both – the legacy xconnect and Protocol-CLI (interface pseudowire configuration) modes are supported.
- Xconnect and MACSec cannot be configured on the same interface.
- MACSec should be configured on CE devices and Xconnect should be configured on PE devices.
- A MACSec session should be between CE devices.

By default, EoMPLS PW tunnels all protocols like CDP, STP. EoMPLS PW cannot perform selective protocol tunneling as part of L2 Protocol Tunneling CLI.

Configuring Port-Mode EoMPLS

Port-Mode EoMPLS can be configured in two modes :

- Xconnect Mode
- Protocol CLI Method

Xconnect Mode

To configure port-mode EoMPLS in xconnect mode, perform the following task :

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no switchport**
5. **no ip address**
6. **no keepalive**
7. **xconnect** *peer-device-id* *vc-id* **encapsulation mpls**
8. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>interface-id</i> Example: Device (config) # interface TenGigabitEthernet1/0/36 | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Step 4 | no switchport Example: Device (config-if) # no switchport | For physical ports only, enters Layer 3 mode.. |
| Step 5 | no ip address Example: Device (config-if) # no ip address | Ensures that there is no IP address assigned to the physical port. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 6 | no keepalive Example: Device(config-if)# no keepalive | Ensures that the device does not send keepalive messages. |
| Step 7 | xconnect <i>peer-device-id</i> <i>vc-id</i> encapsulation mpls Example: Device(config-if)# xconnect 1.1.1.1 962 encapsulation mpls | Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports. |
| Step 8 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Protocol CLI Method

To configure port-mode EoMPLS in protocol-CLI mode, perform the following task :

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **port-channel load-balance *dst-ip***
4. **interface *interface-id***
5. **no switchport**
6. **no ip address**
7. **no keepalive**
8. **exit**
9. **interface pseudowire *number***
10. **encapsulation mpls**
11. **neighbor *peer-device-id* *vc-id***
12. **load-balance flow ip *dst-ip***
13. **load-balance flow-label both**
14. **l2vpn xconnect context *context-name***
15. **member *interface-id***
16. **member pseudowire *number***
17. **end**

DETAILED STEPS

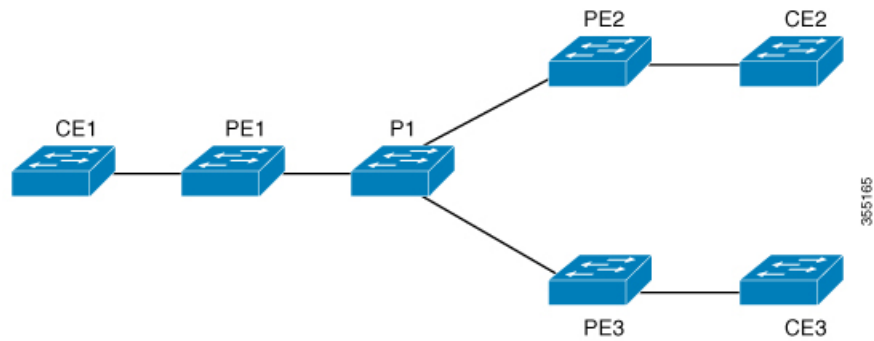
| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | port-channel load-balance dst-ip Example: Device(config)# port-channel load-balance 192.168.2.25 | Sets the load-distribution method to the destination IP address. <ul style="list-style-type: none"> • <i>dst-ip</i>— Destination IP address |
| Step 4 | interface interface-id Example: Device(config)# interface TenGigabitEthernet1/0/21 | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Step 5 | no switchport Example: Device(config-if)# no switchport | For physical ports only, enters Layer 3 mode.. |
| Step 6 | no ip address Example: Device(config-if)# no ip address | Ensures that there is no IP address assigned to the physical port. |
| Step 7 | no keepalive Example: Device(config-if)# no keepalive | Ensures that the device does not send keepalive messages. |

| | Command or Action | Purpose |
|----------------|---|--|
| Step 8 | exit Example: Device(config-if) # exit | Exits interface configuration mode. |
| Step 9 | interface pseudowire <i>number</i> Example: Device(config-if) # interface pseudowire 17 | Establishes an interface pseudowire with a value that you specify and enters pseudowire configuration mode. <ul style="list-style-type: none"> • <i>number</i> — Specifies the number of the pseudowire to be configured. |
| Step 10 | encapsulation mpls Example: Device(config-if) # encapsulation mpls | Specifies the tunneling encapsulation. |
| Step 11 | neighbor <i>peer-device-id vc-id</i> Example: Device(config-if) # neighbor 4.4.4.4 17 | Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire. |
| Step 12 | load-balance flow ip <i>dst-ip</i> Example: Device(config-if) # load-balance flow ip 192.168.2.25 | Enables edge load balancing of traffic across multiple core facing interfaces using equal cost multipaths (ECMP). <ul style="list-style-type: none"> • <i>dst-ip</i>— Destination IP address |
| Step 13 | load-balance flow-label both Example: Device(config-if) # load-balance flow-label both | Enables core load balancing based on flow-labels. |
| Step 14 | l2vpn xconnect context <i>context-name</i> Example: Device(config) # l2vpn xconnect context vpws17 | Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect context configuration mode. |

| | Command or Action | Purpose |
|---------|---|--|
| Step 15 | member <i>interface-id</i> Example: Device(config-if)# member TenGigabitEthernet1/0/21 | Specifies interface that forms a Layer 2 VPN (L2VPN) cross connect. |
| Step 16 | member pseudowire <i>number</i> Example: Device(config-if)# member pseudowire 17 | Specifies pseudowire interface that forms a Layer 2 VPN (L2VPN) cross connect. |
| Step 17 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuration Examples for EoMPLS

Figure 2: EoMPLS Topology



| PE Configuration | CE Configuration |
|--|--|
| <pre> mpls ip mpls label protocol ldp mpls ldp graceful-restart mpls ldp router-id loopback 1 force interface Loopback1 ip address 1.1.1.1 255.255.255.255 ip ospf 100 area 0 router ospf 100 router-id 1.1.1.1 nsf system mtu 9198 port-channel load-balance dst-ip ! interface GigabitEthernet2/0/39 no switchport no ip address no keepalive ! interface pseudowire101 encapsulation mpls neighbor 4.4.4.4 101 load-balance flow ip dst-ip load-balance flow-label both l2vpn xconnect context pw101 member pseudowire101 member GigabitEthernet2/0/39 ! interface TenGigabitEthernet3/0/10 switchport trunk allowed vlan 142 switchport mode trunk channel-group 42 mode active ! interface Port-channel42 switchport trunk allowed vlan 142 switchport mode trunk ! interface Vlan142 ip address 142.1.1.1 255.255.255.0 ip ospf 100 area 0 mpls ip mpls label protocol ldp ! </pre> | <pre> interface GigabitEthernet1/0/33 switchport trunk allowed vlan 912 switchport mode trunk spanning-tree portfast trunk ! interface Vlan912 ip address 10.91.2.3 255.255.255.0 ! </pre> |

The following is a sample output of **show mpls l2 vc vcid vc-id detail** command :

```

Local interface: Gil/0/1 up, line protocol up, Ethernet up
  Destination address: 1.1.1.1, VC ID: 101, VC status: up
Output interface: Vl182, imposed label stack {17 16}
Preferred path: not configured
Default path: active
Next hop: 182.1.1.1
Load Balance: ECMP
flow classification: ip dst-ip
Create time: 06:22:11, last status change time: 05:58:42

```

```

Last label FSM state change time: 05:58:42 Signaling protocol:
LDP, peer 1.1.1.1:0 up
Targeted Hello: 4.4.4.4(LDP Id) -> 1.1.1.1, LDP is UP
Graceful restart: not configured and not enabled
Non stop routing: not configured and not enabled
Status TLV support (local/remote) : enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru
Last local dataplane status rcvd: No fault
Last BFD dataplane status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV status sent: No fault
Last remote LDP TLV status rcvd: No fault
Last remote LDP ADJ status rcvd: No fault
MPLS VC labels: local 512, remote 16
Group ID: local n/a, remote 0
MTU: local 9198, remote 9198
Remote interface description: Sequencing: receive disabled, send disabled

Control Word: On (configured: autosense)
SSO Descriptor: 1.1.1.1/101, local label: 512
Dataplane:
SSM segment/switch IDs: 4096/4096 (used), PWID: 1
VC statistics: transit packet totals: receive 172116845, send 172105364

transit byte totals: receive 176837217071, send 172103349728
transit packet drops: receive 0, seq error 0, send 0

```

The following is a sample output of **show l2vpn atom vc vcid vc-id detail** command :

```

pseudowire101 is up, VC status is up PW type: Ethernet
Create time: 06:30:41, last status change time: 06:07:12
Last label FSM state change time: 06:07:12
Destination address: 1.1.1.1 VC ID: 101
Output interface: Vl182, imposed label stack {17 16}
Preferred path: not configured
Default path: active Next hop: 182.1.1.1
Load Balance: ECMP Flow classification: ip dst-ip
Member of xconnect service pw101
Associated member Gi1/0/1 is up, status is up
Interworking type is Like2Like Service id: 0xe5000001
Signaling protocol: LDP, peer 1.1.1.1:0 up
Targeted Hello: 4.4.4.4(LDP Id) -> 1.1.1.1, LDP is UP
Graceful restart: not configured and not enabled
Non stop routing: not configured and not enabled
PWid FEC (128), VC ID: 101 Status TLV support (local/remote)
: enabled/supported
LDP route watch : enabled
Label/status state machine : established, LruRru

```

```

Local dataplane status received      : No fault
BFD dataplane status received       : Not sent
BFD peer monitor status received    : No fault
Status received from access circuit : No fault
Status sent to access circuit       : No fault
Status received from pseudowire i/f : No fault
Status sent to network peer         : No fault
Status received from network peer   : No fault
Adjacency status of remote peer     : No fault
Sequencing: receive disabled, send disabled Bindings
  Parameter      Local                               Remote
-----
Label            512                                16
Group ID        n/a                                0
Interface

MTU              9198                                9198
Control word on (configured: autosense) on
PW type          Ethernet                            Ethernet
VCCV CV type    0x02                                0x02
                LSPV [2]                            LSPV [2]

VCCV CC type    0x06                                0x06
                RA [2], TTL [3]                        RA [2], TTL [3]
Status TLV      enabled                            supported
Flow Label      T=1, R=1                            T=1, R=1
SSO Descriptor: 1.1.1.1/101, local label: 512
Dataplane:
SSM segment/switch IDs: 4096/4096 (used), PWID: 1
Rx Counters     176196691 input transit packets, 181028952597 bytes
                 0 drops, 0 seq err
Tx Counters     176184928 output transit packets, 176182865992 bytes
                 0 drops

```

The following is a sample output of **show mpls forwarding-table** command:

| Local | Outgoing | Prefix | Bytes | Label | Outgoing | Next Hop |
|-------|----------|--------------|----------|-------|-----------|-----------|
| Label | Label | or Tunnel Id | Switched | | interface | |
| 57 | 18 | 1.1.1.1/32 | 0 | | Po45 | 145.1.1.1 |
| | No Label | 1.1.1.1/32 | 0 | | Te1/0/2 | 147.1.1.1 |
| | No Label | 1.1.1.1/32 | 0 | | Te1/0/11 | 149.1.1.1 |
| | No Label | 1.1.1.1/32 | 0 | | Te1/0/40 | 155.1.1.1 |

Configuring Pseudowire Redundancy

Information About Pseudowire Redundancy

The L2VPN Pseudowire Redundancy feature enables you to configure your network to detect a failure in the network and reroute the Layer 2 (L2) service to another endpoint that can continue to provide service. This feature provides the ability to recover from a failure either of the remote provider edge (PE) router or of the link between the PE and customer edge (CE) routers.

Pseudowire Redundancy (PWR) can be configured using both – the xconnect and the protocol-CLI method.

For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Prerequisites for Pseudowire Redundancy

- Configure **no switchport**, **no keepalive** and **no ip address** before configuring xconnect mode to connect the attachment circuit.
- For load-balancing, **port-channel load-balance** command is mandatory to be configured.

Restrictions for Pseudowire Redundancy

- VLAN mode, EFP (Ethernet Flow Point) and IGMP Snooping is not supported.
- PWR is supported with port mode EoMPLS only.
- Untagged, tagged and 802.1Q in 802.1Q are supported as incoming traffic.



Note Load balance for 802.1Q in 802.1Q with Pseudowire Redundancy is not supported.

- Flow Label for ECMP Load balancing in core network based on customer's source IP, destination IP, source MAC and destination MAC.
- Enabling or disabling Control word is supported.
- MPLS QoS is supported in Pipe and Uniform Mode. Default mode is Pipe Mode.
- Port-channel as attachment circuit is not supported.
- QoS : Customer DSCP Re-marking is not supported with VPWS and EoMPLS.
- VCCV Ping with explicit null is not supported.
- L2 VPN Interworking is not supported.
- **ip unnumbered** command is not supported in MPLS configuration.
- Not more than one backup pseudowire supported.
- PW redundancy group switchover is not supported

Configuring Pseudowire Redundancy

Pseudowire Redundancy can be configured in two modes :

- Xconnect Mode
- Protocol CLI Method

Xconnect Mode

To configure pseudowire redundancy in xconnect mode, perform the following task :



Note To enable load balance, use the corresponding load-balance commands from [Xconnect Mode, on page 31](#) section of Configuring Port-Mode EoMPLS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface *interface-id***
4. **no switchport**
5. **no ip address**
6. **no keepalive**
7. **xconnect *peer-device-id* *vc-id*encapsulation mpls**
8. **backup peer *peer-router-ip-addr* *vcid* *vc-id* [**priority value**]**
9. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>interface-id</i> Example: Device(config)# interface GigabitEthernet1/0/44 | Defines the interface to be configured as a trunk, and enters interface configuration mode. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 4 | no switchport Example: Device(config-if) # no switchport | For physical ports only, enters Layer 3 mode.. |
| Step 5 | no ip address Example: Device(config-if) # no ip address | Ensures that there is no IP address assigned to the physical port. |
| Step 6 | no keepalive Example: Device(config-if) # no keepalive | Ensures that the device does not send keepalive messages. |
| Step 7 | xconnect peer-device-id vc-id encapsulation mpls Example: Device(config-if) # xconnect 1.1.1.1 117 encapsulation mpls | Binds the attachment circuit to a pseudowire VC. The syntax for this command is the same as for all other Layer 2 transports. |
| Step 8 | backup peer peer-router-ip-addr vcid vc-id [priority value] Example: Device(config-if) # backup peer 6.6.6.6 118 priority 9 | Specifies a redundant peer for a pseudowire virtual circuit (VC). |
| Step 9 | end Example: Device(config) # end | Returns to privileged EXEC mode. |

Protocol CLI Method

To configure pseudowire redundancy in protocol-CLI mode, perform the following task :

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no switchport**
5. **no ip address**
6. **no keepalive**
7. **exit**
8. **interface pseudowire** *number*
9. **encapsulation mpls**
10. **neighbor** *peer-device-id vc-id*
11. **exit**
12. **interface pseudowire** *number*
13. **encapsulation mpls**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>interface-id</i> Example: Device(config)# interface GigabitEthernet2/0/39 | Defines the interface to be configured as a trunk, and enters interface configuration mode. |
| Step 4 | no switchport Example: Device(config-if)# no switchport | For physical ports only, enters Layer 3 mode.. |
| Step 5 | no ip address Example: Device(config-if)# no ip address | Ensures that there is no IP address assigned to the physical port. |

| | Command or Action | Purpose |
|----------------|--|---|
| Step 6 | no keepalive Example: Device(config-if)# no keepalive | Ensures that the device does not send keepalive messages. |
| Step 7 | exit Example: Device(config-if)# exit | Exits interface configuration mode. |
| Step 8 | interface pseudowire <i>number</i> Example: Device(config)# interface pseudowire 101 | Establishes an interface pseudowire with a value that you specify and enters pseudowire configuration mode. |
| Step 9 | encapsulation mpls Example: Device(config-if)# encapsulation mpls | Specifies the tunneling encapsulation. |
| Step 10 | neighbor <i>peer-device-id vc-id</i> Example: Device(config-if)# neighbor 4.4.4.4 101 | Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire. |
| Step 11 | exit Example: Device(config-if)# exit | Exits interface configuration mode. |
| Step 12 | interface pseudowire <i>number</i> Example: Device(config)# interface pseudowire 102 | Establishes an interface pseudowire with a value that you specify and enters pseudowire configuration mode. |

| | Command or Action | Purpose |
|---------|--------------------------------|---------|
| Step 13 | encapsulation mpls Example: | |

Configuration Examples for Pseudowire Redundancy

| PE Configuration | CE Configuration |
|--|--|
| <pre> mpls ip mpls label protocol ldp mpls ldp graceful-restart mpls ldp router-id loopback 1 force ! interface Loopback1 ip address 1.1.1.1 255.255.255.255 ip ospf 100 area 0 router ospf 100 router-id 1.1.1.1 nsf ! interface GigabitEthernet2/0/39 no switchport no ip address no keepalive ! interface pseudowire101 encapsulation mpls neighbor 4.4.4.4 101 ! interface pseudowire102 encapsulation mpls neighbor 3.3.3.3 101 l2vpn xconnect context pw101 member pseudowire101 group pwgrp1 priority 1 member pseudowire102 group pwgrp1 priority 15 member GigabitEthernet2/0/39 ! interface TenGigabitEthernet3/0/10 switchport trunk allowed vlan 142 switchport mode trunk channel-group 42 mode active ! interface Port-channel42 switchport trunk allowed vlan 142 switchport mode trunk ! interface Vlan142 ip address 142.1.1.1 255.255.255.0 ip ospf 100 area 0 mpls ip mpls label protocol ldp ! </pre> | <pre> interface GigabitEthernet1/0/33 switchport trunk allowed vlan 912 switchport mode trunk spanning-tree portfast trunk ! interface Vlan912 ip address 10.91.2.3 255.255.255.0 ! </pre> |

The following is sample output of the **show mpls l2transport vc vc-id** command :

```
Device# show mpls l2transport vc 101
Local intf   Local circuit   Dest address   VC ID   Status
-----
Gi2/0/39    Ethernet        4.4.4.4        101     UP

Device# show mpls l2transport vc 102
Local intf   Local circuit   Dest address   VC ID   Status
-----
Gi2/0/39    Ethernet        3.3.3.3        102     STANDBY
```




CHAPTER 5

Configuring IPv6 Provider Edge over MPLS (6PE)

- [Finding Feature Information, on page 47](#)
- [Configuring 6PE, on page 47](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Configuring 6PE

Information About 6PE

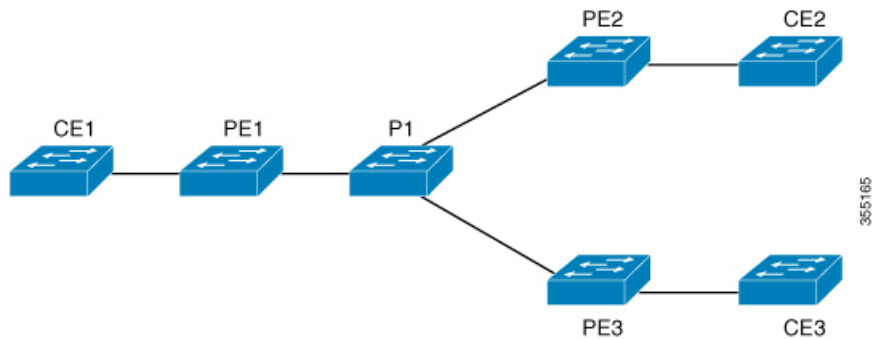
6PE is a technique that provides global IPv6 reachability over IPv4 MPLS. It allows one shared routing table for all other devices. 6PE allows IPv6 domains to communicate with one another over the IPv4 without an explicit tunnel setup, requiring only one IPv4 address per IPv6 domain.

While implementing 6PE, the provider edge routers are upgraded to support 6PE, while the rest of the core network is not touched (IPv6 unaware). This implementation requires no reconfiguration of core routers because forwarding is based on labels rather than on the IP header itself. This provides a cost-effective strategy for deploying IPv6. The IPv6 reachability information is exchanged by PE routers using multiprotocol Border Gateway Protocol (mp-iBGP) extensions.

6PE relies on mp-iBGP extensions in the IPv4 network configuration on the PE router to exchange IPv6 reachability information in addition to an MPLS label for each IPv6 address prefix to be advertised. PE routers are configured as dual stacks, running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange. The next hop advertised by the PE router for 6PE and 6VPE prefixes is still the IPv4 address that is used for IPv4 L3 VPN routes. A value of `::FFFF:` is prepended to the IPv4 next hop, which is an IPv4-mapped IPv6 address.

The following figure illustrates the 6PE topology.

Figure 3: 6PE Topology



For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Prerequisites for 6PE

Redistribute PE-CE IGP IPv6 routes into core BGP and vice-versa

Restrictions for 6PE

eBGP as CE-PE is not supported. Static Routes, OSPFv3, ISIS, RIPv2 are supported as CE-PE.

Configuring 6PE

Ensure that you configure 6PE on PE routers participating in both the IPv4 cloud and IPv6 clouds.

BGP running on a PE router should establish (IPv4) neighborhood with BGP running on other PEs. Subsequently, it should advertise the IPv6 prefixes learnt from the IPv6 table to the neighbors. The IPv6 prefixes advertised by BGP would automatically have IPv4-encoded-IPv6 addresses as the next-hop-address in the advertisement.

To configure 6PE, complete the following steps:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **router bgp *as-number***
5. **bgp router-id interface *interface-id***
6. **bgp log-neighbor-changes**
7. **bgp graceful-restart**
8. **neighbor { *ip-address* | *ipv6-address* | *peer-group-name* } remote-as *as-number***
9. **neighbor { *ip-address* | *ipv6-address* | *peer-group-name* } update-source *interface-type interface-number***
10. **address-family ipv6**
11. **redistribute *protocol as-number* match { internal | external 1 | external 2**
12. **neighbor { *ip-address* | *ipv6-address* | *peer-group-name* } activate**

13. **neighbor** { *ip-address* | *ipv6-address* | *peer-group-name* } **send-label**
14. **exit-address-family**
15. **end**

DETAILED STEPS

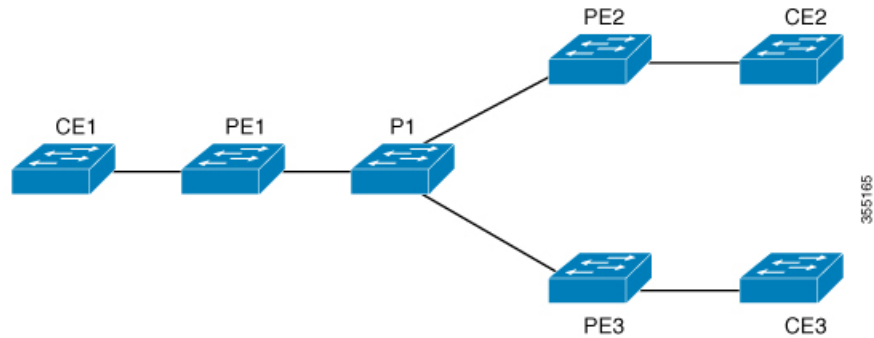
| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |
| Step 3 | ipv6 unicast-routing Example: <pre>Device(config)# ipv6 unicast-routing</pre> | Enables the forwarding of IPv6 unicast datagrams. |
| Step 4 | router bgp <i>as-number</i> Example: <pre>Device(config)# router bgp 65001</pre> | Enters the number that identifies the autonomous system (AS) in which the router resides. <i>as-number</i> —Autonomous system number. Range for 2-byte numbers is 1 to 65535. Range for 4-byte numbers is 1.0 to 65535.65535. |
| Step 5 | bgp router-id interface <i>interface-id</i> Example: <pre>Device(config-router)# bgp router-id interface Loopback1</pre> | Configures a fixed router ID for the local Border Gateway Protocol (BGP) routing process. |
| Step 6 | bgp log-neighbor-changes Example: <pre>Device(config-router)# bgp log-neighbor-changes</pre> | Enables logging of BGP neighbor resets. |
| Step 7 | bgp graceful-restart Example: <pre>Device(config-router)# bgp graceful-restart</pre> | Enables the Border Gateway Protocol (BGP) graceful restart capability globally for all BGP neighbors. |
| Step 8 | neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> | Adds an entry to the BGP or multiprotocol BGP neighbor table. |

| | Command or Action | Purpose |
|----------------|---|---|
| | <p>Example:</p> <pre>Device(config-router)# neighbor 33.33.33.33 remote-as 65001</pre> | <ul style="list-style-type: none"> • <i>ip-address</i>—IP address of a peer router with which routing information will be exchanged. • <i>ipv6-address</i>—IPv6 address of a peer router with which routing information will be exchanged. • <i>peer-group-name</i>—Name of the BGP peer group. • <i>remote-as</i>—Specifies a remote autonomous system. • <i>as-number</i>—Number of an autonomous system to which the neighbor belongs, ranging from 1 to 65535. |
| Step 9 | <p>neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 33.33.33.33 update-source Loopback1</pre> | Configures BGP sessions to use any operational interface for TCP connections. |
| Step 10 | <p>address-family ipv6</p> <p>Example:</p> <pre>Device(config-router)# address-family ipv6</pre> | Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard IPv6 address prefixes. |
| Step 11 | <p>redistribute <i>protocol as-number</i> match { internal external 1 external 2 }</p> <p>Example:</p> <pre>Device(config-router-af)# redistribute ospf 11 match internal external 1</pre> | Redistributes routes from one routing domain into another routing domain. |
| Step 12 | <p>neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 33.33.33.33 activate</pre> | Enables the exchange of information with a BGP neighbor. |
| Step 13 | <p>neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 33.33.33.33 send-label</pre> | Sends MPLS labels with BGP routes to a neighboring BGP router. |
| Step 14 | <p>exit-address-family</p> <p>Example:</p> | Exits BGP address-family submode. |

| | Command or Action | Purpose |
|----------------|---|----------------------------------|
| | Device(config-router-af)# exit-address-family | |
| Step 15 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuration Examples for 6PE

Figure 4: 6PE Topology



| PE Configuration | CE Configuration |
|---|--|
| <pre> router ospfv3 11 ip routing ipv6 unicast-routing address-family ipv6 unicast redistribute bgp 65001 exit-address-family ! router bgp 65001 bgp router-id interface Loopback1 bgp log-neighbor-changes bgp graceful-restart neighbor 33.33.33.33 remote-as 65001 neighbor 33.33.33.33 update-source Loopback1 ! address-family ipv4 neighbor 33.33.33.33 activate ! address-family ipv6 redistribute ospf 11 match internal external 1 external 2 include-connected neighbor 33.33.33.33 activate neighbor 33.33.33.33 send-label neighbor 33.33.33.33 send-community extended ! </pre> | <pre> ipv6 unicast-routing ! interface vlan4 no ip address ipv6 address 10:1:1:2::2/64 ipv6 enable ospfv3 11 ipv6 area 0 ! router ospfv3 11 address-family ipv6 unicast exit-address-family ! </pre> |

The following is a sample output of **show bgp ipv6 unicast summary** :

```

BGP router identifier 1.1.1.1, local AS number 100
BGP table version is 34, main routing table version 34
4 network entries using 1088 bytes of memory
4 path entries using 608 bytes of memory
4/4 BGP path/bestpath attribute entries using 1120 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2816 total bytes of memory
BGP activity 6/2 prefixes, 16/12 paths, scan interval 60 secs

```

```

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down
State/PfxRcd
2.2.2.2        4      100     21     21     34    0    0 00:04:57
                2

```

```

sh ipv route
IPv6 Routing Table - default - 7 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP
external
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid la
- LISP away
C   10:1:1:2::/64 [0/0]
   via Vlan4, directly connected
L   10:1:1:2::1/128 [0/0]
   via Vlan4, receive
LC  11:11:11:11::11/128 [0/0]
   via Loopback1, receive
B   30:1:1:2::/64 [200/0]
   via 33.33.33.33%default, indirectly connected
B   40:1:1:2::/64 [200/0]
   via 44.44.44.44%default, indirectly connected

```

The following is a sample output of **show bgp ipv6 unicast** command :

```

BGP table version is 112, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
           r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
           x best-external, a additional-path, c RIB-compressed,
           t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network      Next Hop      Metric LocPrf Weight Path

```

```
*> 10:1:1:2::/64      ::                0          32768 ?
*>i 30:1:1:2::/64      ::FFFF:33.33.33.33 0          100      0 ?
*>i 40:1:1:2::/64      ::FFFF:44.44.44.44 0          100      0 ?
*>i 173:1:1:2::/64     ::FFFF:33.33.33.33 2          100      0 ?
```

The following is a sample output of **show ipv6 cef 40:1:1:2::0/64 detail** command :

```
40:1:1:2::/64, epoch 6, flags [rib defined all labels]
  recursive via 44.44.44.44 label 67
  nexthop 1.20.4.2 Port-channel103 label 99-(local:147)
```




CHAPTER 6

Configuring IPv6 VPN Provider Edge over MPLS (6VPE)

- [Configuring 6VPE, on page 55](#)
- [Finding Feature Information, on page 60](#)

Configuring 6VPE

Information About 6VPE

6VPE is a mechanism to use the IPv4 backbone to provide VPN IPv6 services. It takes advantage of operational IPv4 MPLS backbones, eliminating the need for dual-stacking within the MPLS core. This translates to savings in operational costs and addresses the security limitations of the 6PE approach. 6VPE is more like a regular IPv4 MPLS-VPN provider edge, with an addition of IPv6 support within VRF. It provides logically separate routing table entries for VPN member devices.

Components of MPLS-based 6VPE Network

- VPN route target communities – A list of all other members of a VPN community.
- Multiprotocol BGP (MP-BGP) peering of VPN community PE routers – Propagates VRF reachability information to all members of a VPN community.
- MPLS forwarding – Transports all traffic between all VPN community members across a VPN service-provider network.

In the MPLS-VPN model a VPN is defined as a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, where the service provider associates each interface with a VPN routing table—known as the VRF table.

For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Restrictions for 6VPE

- Inter-AS and carrier supporting carrier (CSC) is not supported.
- VRF Route-Leaking is not supported.

- eBGP as CE-PE is not supported.
- EIGRP, OSPFv3, RIP, ISIS, Static Routes are supported as CE-PE.
- MPLS Label Allocation modes supported are Per-VRF and Per-Prefix. Per-Prefix is the default mode.

Information About 6VPE

6VPE is a mechanism to use the IPv4 backbone to provide VPN IPv6 services. It takes advantage of operational IPv4 MPLS backbones, eliminating the need for dual-stacking within the MPLS core. This translates to savings in operational costs and addresses the security limitations of the 6PE approach. 6VPE is more like a regular IPv4 MPLS-VPN provider edge, with an addition of IPv6 support within VRF. It provides logically separate routing table entries for VPN member devices.

Components of MPLS-based 6VPE Network

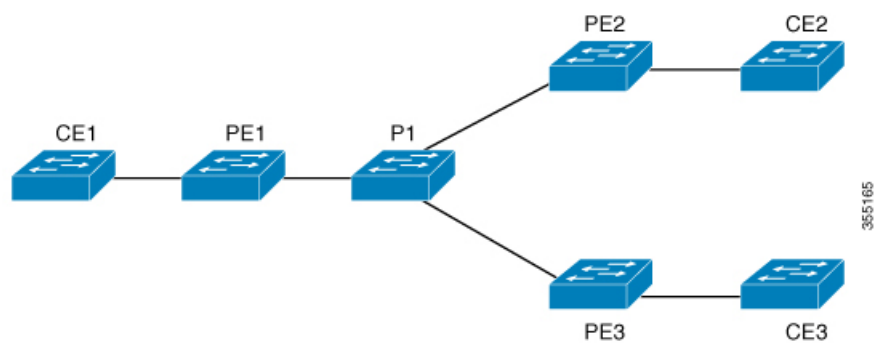
- VPN route target communities – A list of all other members of a VPN community.
- Multiprotocol BGP (MP-BGP) peering of VPN community PE routers – Propagates VRF reachability information to all members of a VPN community.
- MPLS forwarding – Transports all traffic between all VPN community members across a VPN service-provider network.

In the MPLS-VPN model a VPN is defined as a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, where the service provider associates each interface with a VPN routing table—known as the VRF table.

For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Configuration Examples for 6VPE

Figure 5: 6VPE Topology



| PE Configuration | CE Configuration |
|------------------|--|
| | <pre>interface TenGigabitEthernet1/0/38 no switchport ip address 10.3.1.2 255.255.255.0 ip ospf 2 area 0 ipv6 address 10:111:111:111::2/64 ipv6 enable ipv6 ospf 1 area 0 ! router ospfv3 1 nsr graceful-restart address-family ipv6 unicast !</pre> |

| PE Configuration | CE Configuration |
|--|------------------|
| <pre> vrf definition 6VPE-1 rd 65001:11 route-target export 1:1 route-target import 1:1 ! address-family ipv4 exit-address-family ! address-family ipv6 exit-address-family ! interface TenGigabitEthernet1/0/38 no switchport vrf forwarding 6VPE-1 ip address 10.3.1.1 255.255.255.0 ip ospf 2 area 0 ipv6 address 10:111:111:111::1/64 ipv6 enable ospfv3 1 ipv6 area 0 ! router ospf 2 vrf 6VPE-1 router-id 1.1.11.11 redistribute bgp 65001 subnets ! router ospfv3 1 nsr graceful-restart ! address-family ipv6 unicast vrf 6VPE-1 redistribute bgp 65001 exit-address-family ! router bgp 65001 bgp router-id interface Loopback1 bgp log-neighbor-changes bgp graceful-restart neighbor 33.33.33.33 remote-as 65001 neighbor 33.33.33.33 update-source Loopback1 ! address-family ipv4 vrf 6VPE-1 redistribute ospf 2 match internal external 1 external 2 exit-address-family address-family ipv6 vrf 6VPE-1 redistribute ospf 1 match internal external 1 external 2 include-connected exit-address-family ! address-family vpnv4 neighbor 33.33.33.33 activate neighbor 33.33.33.33 send-community both neighbor 44.44.44.44 activate neighbor 44.44.44.44 send-community both neighbor 55.55.55.55 activate neighbor 55.55.55.55 send-community both exit-address-family ! address-family vpnv6 neighbor 33.33.33.33 activate neighbor 33.33.33.33 send-community both neighbor 44.44.44.44 activate </pre> | |

| PE Configuration | CE Configuration |
|--|------------------|
| <pre>neighbor 44.44.44.44 send-community both neighbor 55.55.55.55 activate neighbor 55.55.55.55 send-community both exit-address-family !</pre> | |

The following is a sample output of **show mpls forwarding-table vrf** :

```
Local Outgoing Prefix Bytes Label Outgoing Next Hop
Label Label or Tunnel Id Switched interface
29 No Label A:A:A:565::/64[V] \ 0 aggregate/VRF601
32 No Label A:B5:1:5::/64[V] 2474160 V1601 FE80::200:7BFF:FE62:2636
33 No Label A:B5:1:4::/64[V] 2477978 V1601 FE80::200:7BFF:FE62:2636
35 No Label A:B5:1:3::/64[V] 2477442 V1601 FE80::200:7BFF:FE62:2636
36 No Label A:B5:1:2::/64[V] 2476906 V1601 FE80::200:7BFF:FE62:2636
37 No Label A:B5:1:1::/64[V] 2476370 V1601 FE80::200:7BFF:FE62:2636
```

The following is a sample output of **show vrf counter** command :

```
Maximum number of VRFs supported: 256
Maximum number of IPv4 VRFs supported: 256
Maximum number of IPv6 VRFs supported: 256
Maximum number of platform iVRFs supported: 10
Current number of VRFs: 127
Current number of IPv4 VRFs: 6
Current number of IPv6 VRFs: 127
Current number of VRFs in delete state: 0
Current number of platform iVRFs: 1
```

The following is a sample output of **show ipv6 route vrf** command :

```
IPv6 Routing Table - VRF1 - 8 entries Codes: C - Connected, L - Local, S
- Static, U - Per-user Static route B - BGP, R - RIP, I1 - ISIS L1, I2
- ISIS L2 IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP
external ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr -
Redirect RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1 OE2
- OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2 la - LISP
alt, lr - LISP site-registrations, ld - LISP dyn-eid la - LISP away

B 1:1:1:1::1/128 [200/1] via 1.1.1.11%default, indirectly connected
O 2:2:2:2::2/128 [110/1] via FE80::A2E0:AFFF:FE30:3E40,
TenGigabitEthernet1/0/7
B 3:3:3:3::3/128 [200/1] via 3.3.3.33%default, indirectly connected
B 10:1:1:1::/64 [200/0] via 1.1.1.11%default, indirectly connected
C 10:2:2:2::/64 [0/0] via TenGigabitEthernet1/0/7, directly connected
L 10:2:2:2:1/128 [0/0] via TenGigabitEthernet1/0/7, receive
B 10:3:3:3::/64 [200/0] via 3.3.3.33%default, indirectly connected
L FF00::/8 [0/0] via Null0, receive
```

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



CHAPTER 7

Configuring IP-aware Netflow for VRF Ingress

- [Restrictions for IP-aware Netflow for VRF Ingress, on page 61](#)
- [Information About IP-aware Netflow for VRF Ingress, on page 61](#)
- [How to Configure IP-aware Netflow for VRF Ingress, on page 62](#)
- [Configuration Examples for IP-aware Netflow for VRF Ingress, on page 67](#)
- [Feature History and Information for IP-aware Netflow for VRF Ingress, on page 69](#)

Restrictions for IP-aware Netflow for VRF Ingress

- IP-aware VRF ingress Netflow is supported with IPv4, IPv6 and MVPNv4 as CE facing interface
- Supported only on layer 3 interface
- Supported only for ingress traffic on the VRF interface
- Supported only for MPLS L3 VPN VRF interface
- IP aware VRF ingress Netflow on MVPNv6 as CE facing interface is not supported
- Not supported on portchannel, SVI as CE facing interface
- Not supported for egress traffic on the VRF interface
- Not supported on MPLS L2VPN Attachment circuit interface

Information About IP-aware Netflow for VRF Ingress

This feature enables collecting the virtual routing and forwarding (VRF) ID from incoming packets on a router by applying an input flow monitor having a flow record that collects the VRF ID as a key or a non-key field.

Table 12: Scale Numbers

| Platform | SDM Template | Max IPv4 Flows | Max IPv6 Flows |
|----------|--------------|----------------|----------------|
| 3650 | Advanced | 8k | 4k |
| 3850 | Advanced | 8k | 4k |

| Platform | SDM Template | Max IPv4 Flows | Max IPv6 Flows |
|----------|--------------|----------------|----------------|
| 9300 | Access | 16k | 8k |
| 9400 | Distribution | 32k | 16k |
| 9500 | Access | 32k | 16k |

How to Configure IP-aware Netflow for VRF Ingress

Creating a Flow Record

Perform the following task to create a flow record.

Step 1

SUMMARY STEPS

1. **configure terminal**
2. **flow record** *flow_record_name*
3. **description** *description*
4. **match ipv4 version**
5. **match ipv4** {source | destination} *address*
6. **match ipv4 protocol**
7. **match transport** {source-port | destination-port}
8. **match ipv4 tos**
9. **match ipv4 ttl**
10. **match flow direction**
11. **collect counter packets long**
12. **collect counter bytes long**
13. **end**
14. **show flow record**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | flow record <i>flow_record_name</i> Example: Device(config)# flow record flow-record-1 | Enters flow record configuration mode. |

| | Command or Action | Purpose |
|---------|---|--|
| Step 3 | description <i>description</i> Example: Device (config-flow-record) # description flow-record-1 | (Optional) Creates a description for the flow record. |
| Step 4 | match ipv4 version Example: Device (config-flow-record) # match ipv4 version | Specifies a match to the IP version from the IPv4 header. |
| Step 5 | match ipv4 {source destination} address | Specifies a match to the IPv4 source and destination address. |
| Step 6 | match ipv4 protocol Example: Device (config-flow-record) # match ipv4 protocol | Specifies a match to the IPv4 protocol. |
| Step 7 | match transport {source-port destination-port} | Configures source-port or destination port as a key field for the flow record. |
| Step 8 | match ipv4 tos Example: Device (config-flow-record) # match ipv4 tos | Configures IPv4 ToS as a key field for the flow record. |
| Step 9 | match ipv4 ttl Example: Device (config-flow-record) # match ipv4 ttl | Configures IPv4 TTL as a key field for the flow record. |
| Step 10 | match flow direction Example: Device (config-flow-record) # match flow direction | Specifies a match to the flow identifying fields. |
| Step 11 | collect counter packets long Example: Device (config-flow-record) # collect flow direction | Configures the number of packets seen in a flow as a non-key field and enables collecting the total number of packets from the flow. |
| Step 12 | collect counter bytes long Example: Device (config-flow-record) # collect counter bytes long | Configures the number of bytes seen in a flow as a non-key field and enables collecting the total number of bytes from the flow. |
| Step 13 | end Example: Device (config) # end | Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode. |

| | Command or Action | Purpose |
|---------|--|--|
| Step 14 | show flow record Example: Device # show flow record | Displays information about all the flow records. |

Creating a Flow Exporter

You can create a flow exporter to define the export parameters for a flow.

SUMMARY STEPS

1. **configure terminal**
2. **flow exporter** *flow_exporter_name*
3. **description** *description*
4. **destination** { *hostname* | *ipv4-address* | *ipv6-address* }
5. **source** *interface-type interface-name*
6. **end**
7. **show flow exporter**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | flow exporter <i>flow_exporter_name</i> Example: Device(config)# flow exporter flow-exporter-1 | Enters flow exporter configuration mode. |
| Step 3 | description <i>description</i> Example: Device(config-flow-exporter)# description flow-exporter-1 | (Optional) Creates a description for the flow exporter. |
| Step 4 | destination { <i>hostname</i> <i>ipv4-address</i> <i>ipv6-address</i> } Example: Device (config-flow-exporter)# destination 10.10.1.1 | Specifies the hostname, IPv4 or IPv6 address of the system to which the exporter sends data. |
| Step 5 | source <i>interface-type interface-name</i> Example: Device (config-flow-exporter)# destination 10.10.1.1 | Specifies the local interface from which the exporter will use the IP address as the source IP address for exported datagrams. |

| | Command or Action | Purpose |
|--------|--|---|
| Step 6 | end Example: Device(config)# end | Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode. |
| Step 7 | show flow exporter Example: Device # show flow exporter | Displays information about all the flow exporters. |

Creating a Flow Monitor

You can create a flow monitor and associate it with a flow record.

SUMMARY STEPS

1. **configure terminal**
2. **flow monitor** *monitor-name*
3. **description** *description*
4. **record** *record-name*
5. **exporter** *exporter-name*
6. **cache type normal** {*timeout* | *active* | *inactive*} | **type normal**
7. **end**
8. **show flow monitor**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | flow monitor <i>monitor-name</i> Example: Device (config)# flow monitor flow-monitor-1 | Creates a flow monitor and enters flow monitor configuration mode. |
| Step 3 | description <i>description</i> Example: Device (config-flow-monitor)# description flow-monitor-1 | (Optional) Creates a description for the flow monitor. |
| Step 4 | record <i>record-name</i> Example: Device (config-flow-monitor)# record flow-record-1 | Specifies the name of a record that was created previously. |

| | Command or Action | Purpose |
|---------------|--|---|
| Step 5 | exporter <i>exporter-name</i> Example: Device (config-flow-monitor) # exporter flow-exporter-1 | Specifies the name of an exporter that was created previously. |
| Step 6 | cache type normal { timeout active inactive } type normal | (Optional) Specifies to configure flow cache parameters. |
| Step 7 | end Example: Device(config) # end | Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode. |
| Step 8 | show flow monitor Example: Device # show flow monitor | Displays information about all the flow monitors. |

Applying Flow Monitor to an Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type interface-name*
3. **no switchport**
4. **vrf forwarding** *vrf-name*
5. **{ip | ipv6} flow-monitor** *monitor-name input*
6. **end**
7. **show flow interface**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|---|
| Step 1 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 2 | interface <i>interface-type interface-name</i> | Specifies an interface and enters interface configuration mode. |
| Step 3 | no switchport Example: Device (config-if) # description no switchport | For physical ports only, enters Layer 3 mode. |
| Step 4 | vrf forwarding <i>vrf-name</i> | Associates the VRF with the Layer 3 interface. |
| Step 5 | {ip ipv6} flow-monitor <i>monitor-name input</i> | Associates a flow monitor to the interface for input packets. |

| | Command or Action | Purpose |
|--------|--|---|
| Step 6 | end Example: Device(config)# end | Returns to privileged EXEC mode. Alternatively, you can also press Ctrl-Z to exit global configuration mode. |
| Step 7 | show flow interface Example: Device # show flow interface | Displays the status of NetFlow (enabled or disabled) on the specified interface. |

Configuration Examples for IP-aware Netflow for VRF Ingress

The **show flow interface** command displays information about Netflow on the specified interface. :

```
Interface TenGigabitEthernet1/0/36
FNF: monitor: v4vrfingress
direction: Input
traffic(ip): on
FNF: monitor: v6vrfingress
direction: Input
traffic(ipv6): on
```

The **show flow monitor *flow-monitor-name* cache** command displays the contents of the cache for the flow monitor.

```
Cache type: Normal (Platform cache)
Cache size: 10000
Current entries: 100

Flows added: 100
Flows aged: 0

IPV4 SOURCE ADDRESS: 108.3.20.100
IPV4 DESTINATION ADDRESS: 108.2.20.100
TRNS SOURCE PORT: 0
TRNS DESTINATION PORT: 0
FLOW DIRECTION: Input
IP VERSION: 4
IP TOS: 0x20
IP PROTOCOL: 255
IP TTL: 64
counter bytes long: 2956000
counter packets long: 2000
```

The **show flow exporter** command displays information about all the flow exporters. :

```
Flow Exporter v4vrfingress:
Description: User defined
Export protocol: NetFlow Version 9
Transport Configuration:
Destination type: IP
```

```

Destination IP address: 15.15.15.16
Source IP address:      15.15.15.15
Source Interface:       TenGigabitEthernet1/0/1
Transport Protocol:     UDP
Destination Port:       9995
Source Port:            52319
DSCP:                   0x0
TTL:                    255
Output Features:        Used
Flow Exporter v6vrfingress:
Description:             User defined
Export protocol:         NetFlow Version 9
Transport Configuration:
  Destination type:      IP
  Destination IP address: 15.15.15.16
  Source IP address:     15.15.15.15
  Source Interface:       TenGigabitEthernet1/0/1
  Transport Protocol:     UDP
  Destination Port:       9995
  Source Port:            50881
  DSCP:                   0x0
  TTL:                    255
  Output Features:        Used

```

The **show platform software fed switch active fnf monitors-dump** displays Netflow monitors dump.

```

FNF Monitors
=====
Monitor (0x7f4afc031748):
  profile_id(c461d4fe) ref_ct(1) wdavc_monitor(0)
  wdavc_monitor_create_requested(False)
  wdavc_remote_monitoring_remote_caching(0) flags(0x0000) is_wireless(No)
  is_etta_over_fnf No ettaOrBaseProfile(00000000) etta_refcnt(0)
  field(113) size(16) param(0) flags(1) offset(0)
  field(114) size(16) param(0) flags(1) offset(16)
  field(118) size(2) param(0) flags(1) offset(32)
  field(119) size(2) param(0) flags(1) offset(34)
  field(156) size(1) param(0) flags(1) offset(36)
  field(181) size(8) param(0) flags(0) offset(37)
  field(42) size(1) param(0) flags(1) offset(45)
  field(46) size(1) param(0) flags(1) offset(46)
  field(43) size(1) param(0) flags(1) offset(47)
  field(47) size(1) param(0) flags(1) offset(48)
Monitor (0x7f4afc029338):
  profile_id(74c02ab0) ref_ct(1) wdavc_monitor(0)
  wdavc_monitor_create_requested(False)
  wdavc_remote_monitoring_remote_caching(0) flags(0x0000) is_wireless(No)
  is_etta_over_fnf No ettaOrBaseProfile(00000000) etta_refcnt(0)
  field(93) size(4) param(0) flags(1) offset(0)
  field(94) size(4) param(0) flags(1) offset(4)
  field(118) size(2) param(0) flags(1) offset(8)
  field(119) size(2) param(0) flags(1) offset(10)

```

```
field(156) size(1) param(0) flags(1) offset(12)
field(177) size(8) param(0) flags(0) offset(13)
field(181) size(8) param(0) flags(0) offset(21)
field(42) size(1) param(0) flags(1) offset(29)
field(43) size(1) param(0) flags(1) offset(30)
field(46) size(1) param(0) flags(1) offset(31)
field(47) size(1) param(0) flags(1) offset(32)
```

Feature History and Information for IP-aware Netflow for VRF Ingress

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

| Release | Modification |
|---------------------------|------------------------------|
| Cisco IOS XE Fuji 16.8.1a | This feature was introduced. |



CHAPTER 8

Configuring Virtual Private LAN Service (VPLS) and VPLS BGP-Based Autodiscovery

- [Finding Feature Information, on page 71](#)
- [Configuring VPLS, on page 71](#)
- [Configuring VPLS BGP-based Autodiscovery, on page 86](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Configuring VPLS

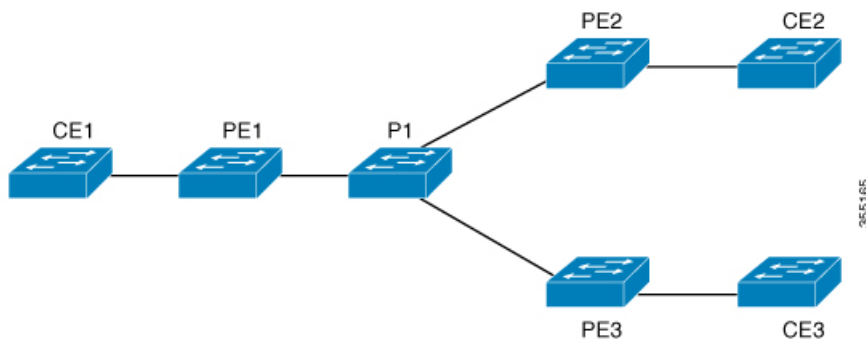
Information About VPLS

VPLS Overview

VPLS (Virtual Private LAN Service) enables enterprises to link together their Ethernet-based LANs from multiple sites via the infrastructure provided by their service provider. From the enterprise perspective, the service provider's public network looks like one giant Ethernet LAN. For the service provider, VPLS provides an opportunity to deploy another revenue-generating service on top of their existing network without major capital expenditures. Operators can extend the operational life of equipment in their network.

Virtual Private LAN Services (VPLS) uses the provider core to join multiple attachment circuits together to simulate a virtual bridge that connects the multiple attachment circuits together. From a customer point of view, there is no topology for VPLS. All of the CE devices appear to connect to a logical bridge emulated by the provider core.

Figure 6: VPLS Topology



Full-Mesh Configuration

The full-mesh configuration requires a full mesh of tunnel label switched paths (LSPs) between all the PEs that participate in the VPLS. With full-mesh, signaling overhead and packet replication requirements for each provisioned VC on a PE can be high.

You set up a VPLS by first creating a virtual forwarding instance (VFI) on each participating PE router. The VFI specifies the VPN ID of a VPLS domain, the addresses of other PE devices in the domain, and the type of tunnel signaling and encapsulation mechanism for each peer PE router.

The set of VFIs formed by the interconnection of the emulated VCs is called a VPLS instance; it is the VPLS instance that forms the logic bridge over a packet switched network. The VPLS instance is assigned a unique VPN ID.

The PE devices use the VFI to establish a full-mesh LSP of emulated VCs to all the other PE devices in the VPLS instance. PE devices obtain the membership of a VPLS instance through static configuration using the Cisco IOS CLI.

The full-mesh configuration allows the PE router to maintain a single broadcast domain. Thus, when the PE router receives a broadcast, multicast, or unknown unicast packet on an attachment circuit, it sends the packet out on all other attachment circuits and emulated circuits to all other CE devices participating in that VPLS instance. The CE devices see the VPLS instance as an emulated LAN.

To avoid the problem of a packet looping in the provider core, the PE devices enforce a "split-horizon" principle for the emulated VCs. That means if a packet is received on an emulated VC, it is not forwarded on any other emulated VC.

After the VFI has been defined, it needs to be bound to an attachment circuit to the CE device.

The packet forwarding decision is made by looking up the Layer 2 virtual forwarding instance (VFI) of a particular VPLS domain.

A VPLS instance on a particular PE router receives Ethernet frames that enter on specific physical or logical ports and populates a MAC table similarly to how an Ethernet switch works. The PE router can use the MAC address to switch those frames into the appropriate LSP for delivery to the another PE router at a remote site.

If the MAC address is not in the MAC address table, the PE router replicates the Ethernet frame and floods it to all logical ports associated with that VPLS instance, except the ingress port where it just entered. The PE router updates the MAC table as it receives packets on specific ports and removes addresses not used for specific periods.

VPLS BGP Based Autodiscovery

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

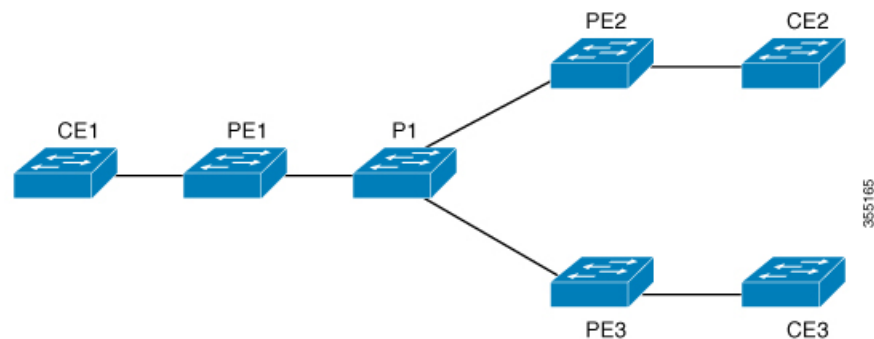
BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network.

For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Configuration Examples for VPLS

Figure 7: VPLS Topology



| PE1 Configuration | PE2 Configuration |
|---|---|
| <pre>pseudowire-class vpls2129 encapsulation mpls ! l2 vfi 2129 manual vpn id 2129 neighbor 44.254.44.44 pw-class vpls2129 ! neighbor 188.98.89.98 pw-class vpls2129 ! interface TenGigabitEthernet1/0/24 switchport trunk allowed vlan 2129 switchport mode trunk ! interface Vlan2129 no ip address xconnect vfi 2129 !</pre> | <pre>pseudowire-class vpls2129 encapsulation mpls no control-word ! l2 vfi 2129 manual vpn id 2129 neighbor 1.1.1.72 pw-class vpls2129 neighbor 188.98.89.98 pw-class vpls2129 ! interface TenGigabitEthernet1/0/47 switchport trunk allowed vlan 2129 switchport mode trunk end ! interface Vlan2129 no ip address xconnect vfi 2129 !</pre> |

The **show mpls 12transport vc detail** command provides information the virtual circuits.

```
Local interface: VFI 2129 vfi up
 Interworking type is Ethernet
 Destination address: 44.254.44.44, VC ID: 2129, VC status: up
 Output interface: Gil/0/9, imposed label stack {18 17}
 Preferred path: not configured
 Default path: active
 Next hop: 177.77.177.2
 Create time: 19:09:33, last status change time: 09:24:14
 Last label FSM state change time: 09:24:14
 Signaling protocol: LDP, peer 44.254.44.44:0 up
 Targeted Hello: 1.1.1.72(LDP Id) -> 44.254.44.44, LDP is UP
 Graceful restart: configured and enabled
 Non stop routing: not configured and not enabled
 Status TLV support (local/remote) : enabled/supported
 LDP route watch : enabled
 Label/status state machine : established, LruRru
 Last local dataplane status rcvd: No fault
 Last BFD dataplane status rcvd: Not sent
 Last BFD peer monitor status rcvd: No fault
 Last local AC circuit status rcvd: No fault
 Last local AC circuit status sent: No fault
 Last local PW i/f circ status rcvd: No fault
 Last local LDP TLV status sent: No fault
 Last remote LDP TLV status rcvd: No fault
 Last remote LDP ADJ status rcvd: No fault
 MPLS VC labels: local 512, remote 17
 Group ID: local n/a, remote 0
 MTU: local 1500, remote 1500
 Remote interface description:
 Sequencing: receive disabled, send disabled
 Control Word: Off
```

```
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
  SSM segment/switch IDs: 20498/20492 (used), PWID: 2
VC statistics:
  transit packet totals: receive 0, send 0
  transit byte totals:   receive 0, send 0
  transit packet drops: receive 0, seq error 0, send 0
```

The **show l2vpn atom vc** shows that ATM over MPLS is configured on a VC.

```
pseudowire100005 is up, VC status is up PW type: Ethernet
  Create time: 19:25:56, last status change time: 09:40:37
  Last label FSM state change time: 09:40:37
  Destination address: 44.254.44.44 VC ID: 2129
  Output interface: Gi1/0/9, imposed label stack {18 17}
  Preferred path: not configured
  Default path: active
  Next hop: 177.77.177.2
Member of vfi service 2129
  Bridge-Domain id: 2129
  Service id: 0x32000003
Signaling protocol: LDP, peer 44.254.44.44:0 up
  Targeted Hello: 1.1.1.72 (LDP Id) -> 44.254.44.44, LDP is UP
  Graceful restart: configured and enabled
  Non stop routing: not configured and not enabled
  PWid FEC (128), VC ID: 2129
  Status TLV support (local/remote)           : enabled/supported
  LDP route watch                             : enabled
  Label/status state machine                  : established, LruRru
  Local dataplane status received              : No fault
  BFD dataplane status received               : Not sent
  BFD peer monitor status received            : No fault
  Status received from access circuit         : No fault
  Status sent to access circuit               : No fault
  Status received from pseudowire i/f        : No fault
  Status sent to network peer                 : No fault
  Status received from network peer           : No fault
  Adjacency status of remote peer             : No fault
  Sequencing: receive disabled, send disabled
  Bindings
  Parameter      Local                               Remote
  -----
  Label          512                               17
  Group ID       n/a                               0
  Interface
  MTU            1500                               1500
  Control word   off                               off
  PW type        Ethernet                           Ethernet
  VCCV CV type   0x02                               0x02
```

```

LSPV [2]                                LSPV [2]
VCCV CC type 0x06                        0x06
RA [2], TTL [3]                          RA [2], TTL [3]
Status TLV enabled                       supported
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
  SSM segment/switch IDs: 20498/20492 (used), PWID: 2
Rx Counters
  0 input transit packets, 0 bytes
  0 drops, 0 seq err
Tx Counters
  0 output transit packets, 0 bytes
  0 drops

```

Restrictions for VPLS

- Protocol-based CLI Method (interface pseudowire configuration) is not supported. Only VFI and Xconnect mode are supported.
- Flow-Aware Transport Pseudowire (FAT PW) is not supported.
- IGMP Snooping is not Supported. Multicast traffic floods with IGMP Snooping disabled.
- L2 Protocol Tunneling is not supported.
- Integrated Routing and Bridging (IRB) not supported.
- Virtual Circuit Connectivity Verification (VCCV) ping with explicit null is not supported.
- The switch is supported only as spoke in H-VPLS but not as hub.
- L2 VPN Interworking is not supported.
- **ip unnumbered** command is not supported in MPLS configuration.
- VC statistics are not displayed for flood traffic in the output of show mpls l2 vc vcid detail command.
- dot1q tunnel is not supported in the attachment circuit.

Configuring PE Layer 2 Interfaces to CEs

Configuring 802.1Q Trunks for Tagged Traffic from a CE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no ip address** *ip_address mask* [**secondary**]
5. **switchport**
6. **switchport trunk encapsulation dot1q**

7. `switchport trunk allow vlan vlan_ID`
8. `switchport mode trunk`
9. `end`

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|---|
| Step 1 | <p>enable</p> <p>Example:</p> <pre>Device> enable</pre> | <p>Enables privileged EXEC mode.</p> <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | <p>configure terminal</p> <p>Example:</p> <pre>Device# configure terminal</pre> | <p>Enters global configuration mode.</p> |
| Step 3 | <p>interface <i>interface-id</i></p> <p>Example:</p> <pre>Device(config)# interface TenGigabitEthernet1/0/24</pre> | <p>Defines the interface to be configured as a trunk, and enters interface configuration mode.</p> |
| Step 4 | <p>no ip address <i>ip_address mask</i> [secondary]</p> <p>Example:</p> <pre>Device(config-if)# no ip address</pre> | <p>Disables IP processing and enters interface configuration mode.</p> |
| Step 5 | <p>switchport</p> <p>Example:</p> <pre>Device(config-if)# switchport</pre> | <p>Modifies the switching characteristics of the Layer 2-switched interface.</p> |
| Step 6 | <p>switchport trunk encapsulation dot1q</p> <p>Example:</p> <pre>Device(config-if)# switchport trunk encapsulation dot1q</pre> | <p>Sets the switch port encapsulation format to 802.1Q.</p> |
| Step 7 | <p>switchport trunk allow vlan <i>vlan_ID</i></p> <p>Example:</p> <pre>Device(config-if)# switchport trunk allow vlan 2129</pre> | <p>Sets the list of allowed VLANs.</p> |

| | Command or Action | Purpose |
|---------------|--|--|
| Step 8 | switchport mode trunk Example: Device(config-if)# switchport mode trunk | Sets the interface to a trunking VLAN Layer 2 interface. |
| Step 9 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuring 802.1Q Access Ports for Untagged Traffic from a CE

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface *interface-id***
4. **no ip address *ip_address mask* [secondary]**
5. **switchport**
6. **switchport mode access**
7. **switchport access vlan *vlan_ID***
8. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|--|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface <i>interface-id</i> Example: Device(config)# interface TenGigabitEthernet1/0/24 | Defines the interface to be configured as a trunk, and enters interface configuration mode. |

| | Command or Action | Purpose |
|--------|--|--|
| Step 4 | no ip address <i>ip_address mask</i> [secondary] Example: Device(config-if)# no ip address | Disables IP processing and enters interface configuration mode. |
| Step 5 | switchport Example: Device(config-if)# switchport | Modifies the switching characteristics of the Layer 2-switched interface. |
| Step 6 | switchport mode access Example: Device(config-if)# switchport mode access | Sets the interface type to nontrunking, nontagged single VLAN Layer 2 interface. |
| Step 7 | switchport access vlan <i>vlan_ID</i> Example: Device(config-if)# switchport access vlan 2129 | Sets the VLAN when the interface is in access mode. |
| Step 8 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuring Layer 2 VLAN Instances on a PE

Configuring the Layer 2 VLAN interface on the PE enables the Layer 2 VLAN instance on the PE router to the VLAN database to set up the mapping between the VPLS and VLANs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **interface vlan** *vlan-id*
5. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|----------------------------------|--|
| Step 1 | enable Example: | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |

| | Command or Action | Purpose |
|---------------|--|---|
| | Device> enable | |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | vlan <i>vlan-id</i> Example: Device(config)# vlan 2129 | Configures a specific virtual LAN (VLAN). |
| Step 4 | interface vlan <i>vlan-id</i> Example: Device(config-vlan)# interface vlan 2129 | Configures an interface on the VLAN. |
| Step 5 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuring MPLS in the PE

To configure MPLS in the PE, you must provide the required MPLS parameters.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls ip**
4. **mpls label protocol ldp**
5. **end**
6. **mpls ldp logging neighbor-changes**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |

| | Command or Action | Purpose |
|--------|---|---|
| Step 2 | configure terminal Example: Device# <code>configure terminal</code> | Enters global configuration mode. |
| Step 3 | mpls ip Example: Device(config)# <code>mpls ip</code> | Configures MPLS hop-by-hop forwarding. |
| Step 4 | mpls label protocol ldp Example: Device(config-vlan)# <code>mpls label protocol ldp</code> | Specifies the default Label Distribution Protocol for a platform. |
| Step 5 | end Example: Device(config)# <code>end</code> | Returns to privileged EXEC mode. |
| Step 6 | mpls ldp logging neighbor-changes Example: Device(config)# <code>mpls ldp logging neighbor-changes</code> | (Optional) Determines logging neighbor changes. |

Configuring VFI in the PE

The virtual switch instance (VFI) specifies the VPN ID of a VPLS domain, the addresses of other PE devices in this domain, and the type of tunnel signaling and encapsulation mechanism for each peer (This is where you create the VFI and associated VCs.). Configure a VFI as follows:

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `l2 vfi vfi-name manual`
4. `vpn id vpn-id`
5. `neighbor router-id {encapsulation mpls}`
6. `end`

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---------------------|-------------------------------|
| Step 1 | <code>enable</code> | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---------------|--|---|
| | Example: Device> enable | <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | l2 vfi vfi-name manual Example: Device(config)# l2 vfi 2129 manual | Enables the Layer 2 VFI manual configuration mode. |
| Step 4 | vpn id vpn-id Example: Device(config-vfi)# vpn id 2129 | Configures a VPN ID for a VPLS domain. The emulated VCs bound to this Layer 2 VRF use this VPN ID for signaling. Note <i>vpn-id</i> is the same as <i>vlan-id</i> . |
| Step 5 | neighbor router-id {encapsulation mpls} Example: Device(config-vfi)# neighbor remote-router-id encapsulation mpls | Specifies the remote peering router ID and the tunnel encapsulation type or the pseudo-wire property to be used to set up the emulated VC. |
| Step 6 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Associating the Attachment Circuit with the VFI at the PE

After defining the VFI, you must bind it to one or more attachment circuits.

SUMMARY STEPS

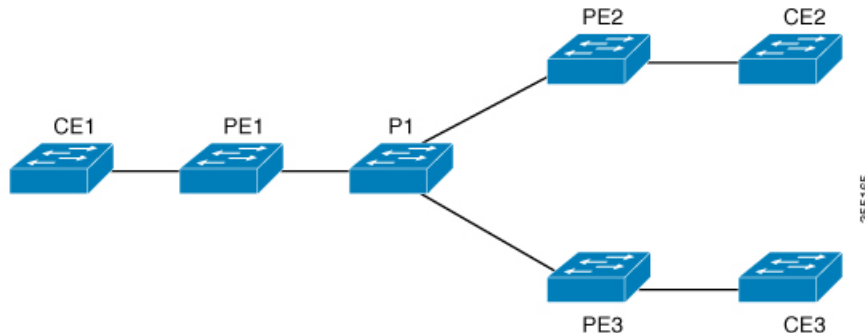
1. **enable**
2. **configure terminal**
3. **interface vlan vlan-id**
4. **no ip address**
5. **xconnect vfi vfi-name**
6. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | interface vlan <i>vlan-id</i> Example: Device(config)# interface vlan 2129 | Creates or accesses a dynamic switched virtual interface (SVI). Note <i>vlan-id</i> is the same as <i>vpn-id</i> . |
| Step 4 | no ip address Example: Device(config-if)# no ip address | Disables IP processing. (You configure a Layer 3 interface for the VLAN if you configure an IP address.) |
| Step 5 | xconnect vfi <i>vfi-name</i> Example: Device(config-if)# xconnect vfi 2129 | Specifies the Layer 2 VFI that you are binding to the VLAN port. |
| Step 6 | end Example: Device(config)# end | Returns to privileged EXEC mode. |

Configuration Examples for VPLS

Figure 8: VPLS Topology



| PE1 Configuration | PE2 Configuration |
|--|---|
| <pre> pseudowire-class vpls2129 encapsulation mpls ! l2 vfi 2129 manual vpn id 2129 neighbor 44.254.44.44 pw-class vpls2129 neighbor 188.98.89.98 pw-class vpls2129 ! interface TenGigabitEthernet1/0/24 switchport trunk allowed vlan 2129 switchport mode trunk ! interface Vlan2129 no ip address xconnect vfi 2129 ! </pre> | <pre> pseudowire-class vpls2129 encapsulation mpls no control-word ! l2 vfi 2129 manual vpn id 2129 neighbor 1.1.1.72 pw-class vpls2129 neighbor 188.98.89.98 pw-class vpls2129 ! interface TenGigabitEthernet1/0/47 switchport trunk allowed vlan 2129 switchport mode trunk end ! interface Vlan2129 no ip address xconnect vfi 2129 ! </pre> |

The **show mpls 12transport vc detail** command provides information the virtual circuits.

```

Local interface: VFI 2129 vfi up
  Interworking type is Ethernet
  Destination address: 44.254.44.44, VC ID: 2129, VC status: up
  Output interface: Gi1/0/9, imposed label stack {18 17}
  Preferred path: not configured
  Default path: active
  Next hop: 177.77.177.2
  Create time: 19:09:33, last status change time: 09:24:14
  Last label FSM state change time: 09:24:14
  Signaling protocol: LDP, peer 44.254.44.44:0 up
  Targeted Hello: 1.1.1.72(LDP Id) -> 44.254.44.44, LDP is UP
  Graceful restart: configured and enabled
  Non stop routing: not configured and not enabled
  Status TLV support (local/remote)   : enabled/supported
  LDP route watch                       : enabled

```

```

Label/status state machine      : established, LruRru
Last local dataplane  status rcvd: No fault
Last BFD dataplane    status rcvd: Not sent
Last BFD peer monitor status rcvd: No fault
Last local AC circuit status rcvd: No fault
Last local AC circuit status sent: No fault
Last local PW i/f circ status rcvd: No fault
Last local LDP TLV    status sent: No fault
Last remote LDP TLV   status rcvd: No fault
Last remote LDP ADJ   status rcvd: No fault
MPLS VC labels: local 512, remote 17
  Group ID: local n/a, remote 0
  MTU: local 1500, remote 1500
  Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: Off
  SSO Descriptor: 44.254.44.44/2129, local label: 512
  Dataplane:
    SSM segment/switch IDs: 20498/20492 (used), PWID: 2
  VC statistics:
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0

```

The **show l2vpn atom vc** shows that ATM over MPLS is configured on a VC.

```

pseudowire100005 is up, VC status is up PW type: Ethernet
  Create time: 19:25:56, last status change time: 09:40:37
  Last label FSM state change time: 09:40:37
  Destination address: 44.254.44.44 VC ID: 2129
  Output interface: Gi1/0/9, imposed label stack {18 17}
  Preferred path: not configured
  Default path: active
  Next hop: 177.77.177.2
  Member of vfi service 2129
  Bridge-Domain id: 2129
  Service id: 0x32000003
  Signaling protocol: LDP, peer 44.254.44.44:0 up
  Targeted Hello: 1.1.1.72 (LDP Id) -> 44.254.44.44, LDP is UP
  Graceful restart: configured and enabled
  Non stop routing: not configured and not enabled
  PWid FEC (128), VC ID: 2129
  Status TLV support (local/remote)      : enabled/supported
  LDP route watch                        : enabled
  Label/status state machine             : established, LruRru
  Local dataplane status received        : No fault
  BFD dataplane status received          : Not sent
  BFD peer monitor status received       : No fault
  Status received from access circuit    : No fault
  Status sent to access circuit          : No fault
  Status received from pseudowire i/f    : No fault

```

```

Status sent to network peer          : No fault
  Status received from network peer   : No fault
  Adjacency status of remote peer     : No fault
Sequencing: receive disabled, send disabled
Bindings
  Parameter      Local              Remote
-----
Label            512                17
Group ID         n/a                0
Interface

MTU              1500                1500
Control word    off                  off
PW type         Ethernet             Ethernet
VCCV CV type    0x02                0x02
                LSPV [2]            LSPV [2]

VCCV CC type    0x06                0x06
                RA [2], TTL [3]      RA [2], TTL [3]
Status TLV      enabled              supported
SSO Descriptor: 44.254.44.44/2129, local label: 512
Dataplane:
  SSM segment/switch IDs: 20498/20492 (used), PWID: 2
Rx Counters
  0 input transit packets, 0 bytes
  0 drops, 0 seq err
Tx Counters
  0 output transit packets, 0 bytes
  0 drops

```

Configuring VPLS BGP-based Autodiscovery

Information About VPLS BGP-Based Autodiscovery

VPLS BGP Based Autodiscovery

VPLS Autodiscovery enables each Virtual Private LAN Service (VPLS) provider edge (PE) device to discover other PE devices that are part of the same VPLS domain. VPLS Autodiscovery also tracks PE devices when they are added to or removed from a VPLS domain. As a result, with VPLS Autodiscovery enabled, you no longer need to manually configure a VPLS domain and maintain the configuration when a PE device is added or deleted. VPLS Autodiscovery uses the Border Gateway Protocol (BGP) to discover VPLS members and set up and tear down pseudowires in a VPLS domain.

BGP uses the Layer 2 VPN (L2VPN) Routing Information Base (RIB) to store endpoint provisioning information, which is updated each time any Layer 2 virtual forwarding instance (VFI) is configured. The prefix and path information is stored in the L2VPN database, which allows BGP to make decisions about the best path. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, this endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

The BGP autodiscovery mechanism facilitates the configuration of L2VPN services, which are an integral part of the VPLS feature. VPLS enables flexibility in deploying services by connecting geographically dispersed sites as a large LAN over high-speed Ethernet in a robust and scalable IP Multiprotocol Label Switching (MPLS) network.

For scale information related to this feature, see [Cisco Catalyst 3650 Series Switches Data Sheet](#).

Enabling VPLS BGP-based Autodiscovery

Perform this task to enable Virtual Private LAN Service (VPLS) PE devices to discover other PE devices that are part of the same VPLS domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi *vfi-name* autodiscovery**
4. **vpn id *vpn-id***
5. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | l2 vfi <i>vfi-name</i> autodiscovery Example: Device(config)# l2 vfi 2128 autodiscovery | Enables VPLS Autodiscovery on a PE device and enters L2 VFI configuration mode. |
| Step 4 | vpn id <i>vpn-id</i> Example: Device(config-vfi)# vpn id 2128 | Configures a VPN ID for the VPLS domain. |
| Step 5 | end Example: | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|--|------------------------------|---------|
| | Device (config) # end | |

Configuring BGP to Enable VPLS Autodiscovery

The Border Gateway Protocol (BGP) Layer 2 VPN (L2VPN) address family supports a separate L2VPN Routing Information Base (RIB) that contains endpoint provisioning information for Virtual Private LAN Service (VPLS) Autodiscovery. BGP learns the endpoint provisioning information from the L2VPN database, which is updated each time a Layer 2 virtual forwarding instance (VFI) is configured. When BGP distributes the endpoint provisioning information in an update message to all its BGP neighbors, the endpoint information is used to configure a pseudowire mesh to support L2VPN-based services.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **no bgp default ipv4-unicast**
5. **bgp log-neighbor-changes**
6. **neighbor remote-as** { *ip-address* | *peer-group-name* } **remote-as** *autonomous-system-number*
7. **neighbor** { *ip-address* | *peer-group-name* } **update-source** *interface-type interface-number*
8. Repeat Steps 6 and 7 to configure other BGP neighbors.
9. **address-family l2vpn** [*vpls*]
10. **neighbor** { *ip-address* | *peer-group-name* } **activate**
11. **neighbor** { *ip-address* | *peer-group-name* } **send-community** { **both** | **standard** | **extended** }
12. Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family.
13. **exit-address-family**
14. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|---|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |

| | Command or Action | Purpose |
|---------------|---|---|
| Step 3 | router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 1000 | Enters router configuration mode for the specified routing process. |
| Step 4 | no bgp default ipv4-unicast Example: Device(config-router)# no bgp default ipv4-unicast | Disables the IPv4 unicast address family for the BGP routing process. Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured using the neighbor remote-as router configuration command unless you configure the no bgp default ipv4-unicast router configuration command before configuring the neighbor remote-as command. Existing neighbor configurations are not affected. |
| Step 5 | bgp log-neighbor-changes Example: Device(config-router)# bgp log-neighbor-changes | Enables logging of BGP neighbor resets. |
| Step 6 | neighbor remote-as { <i>ip-address</i> <i>peer-group-name</i> } remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 44.254.44.44 remote-as 1000 | Adds the IP address or peer group name of the neighbor in the specified autonomous system to the IPv4 multiprotocol BGP neighbor table of the local device. <ul style="list-style-type: none"> • If the autonomous-system-number argument matches the autonomous system number specified in the router bgp command, the neighbor is an internal neighbor. • If the autonomous-system-number argument does not match the autonomous system number specified in the router bgp command, the neighbor is an external neighbor. |
| Step 7 | neighbor { <i>ip-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Device(config-router)# neighbor 44.254.44.44 update-source Loopback300 | (Optional) Configures a device to select a specific source or interface to receive routing table updates. |
| Step 8 | Repeat Steps 6 and 7 to configure other BGP neighbors. | Exits interface configuration mode. |
| Step 9 | address-family l2vpn [vpls] Example: | Specifies the L2VPN address family and enters address family configuration mode. |

| | Command or Action | Purpose |
|----------------|---|---|
| | Device(config-router)# address-family l2vpn vpls | The optional vpls keyword specifies that the VPLS endpoint provisioning information is to be distributed to BGP peers. |
| Step 10 | neighbor { ip-address peer-group-name } activate Example: Device(config-router-af)# neighbor 44.254.44.44 activate | Enables the exchange of information with a BGP neighbor. |
| Step 11 | neighbor { ip-address peer-group-name } send-community { both standard extended } Example: Device(config-router-af)# neighbor 44.254.44.44 send-community both | Specifies that a communities attribute should be sent to a BGP neighbor. |
| Step 12 | Repeat Steps 10 and 11 to activate other BGP neighbors under an L2VPN address family. | |
| Step 13 | exit-address-family Example: Device(config-router-af)# exit-address-family | Exits address family configuration mode and returns to router configuration mode. |
| Step 14 | end Example: Device(config-router-af)# end | Exits router configuration mode and returns to privileged EXEC mode. |

Configuration Examples for VPLS BGP-AD

PE Configuration

```

router bgp 1000
  bgp log-neighbor-changes
  bgp graceful-restart
  neighbor 44.254.44.44 remote-as 1000
  neighbor 44.254.44.44 update-source Loopback300
!
  address-family l2vpn vpls
    neighbor 44.254.44.44 activate
    neighbor 44.254.44.44 send-community both
  exit-address-family
!
l2 vfi 2128 autodiscovery
  vpn id 2128
interface Vlan2128
  no ip address
  xconnect vfi 2128
!

```

The following is a sample output of **show platform software fed sw 1 matm macTable vlan 2000** command :

| VLAN | MAC | Type | Seq# | macHandle | siHandle |
|------|----------------|-----------|---------|------------------|--------------|
| | diHandle | *a_time | *e_time | ports | |
| 2000 | 2852.6134.05c8 | 0X8002 | 0 | 0xffbba312c8 | 0xffbb9ef938 |
| | 0x5154 | 0 | 0 | Vlan2000 | |
| 2000 | 0000.0078.9012 | 0X1 | 32627 | 0xffbb665ec8 | 0xffbb60b198 |
| | 0xffbb653f98 | 300 | 278448 | Port-channel11 | |
| 2000 | 2852.6134.0000 | 0X1 | 32651 | 0xffba15e1a8 | 0xff454c2328 |
| | 0xffbb653f98 | 300 | 63 | Port-channel11 | |
| 2000 | 0000.0012.3456 | 0X2000001 | 32655 | 0xffba15c508 | 0xff44f9ec98 |
| | 0x0 | 300 | 1 | 2000:33.33.33.33 | |

Total Mac number of addresses:: 4

*a_time=aging_time(secs) *e_time=total_elapsed_time(secs)

Type:

| | | | |
|----------------------|-----------|-----------------------|-----------|
| MAT_DYNAMIC_ADDR | 0x1 | MAT_STATIC_ADDR | 0x2 |
| MAT_CPU_ADDR | 0x4 | MAT_DISCARD_ADDR | 0x8 |
| MAT_ALL_VLANS | 0x10 | MAT_NO_FORWARD | 0x20 |
| MAT_IPMULT_ADDR | 0x40 | MAT_RESYNC | 0x80 |
| MAT_DO_NOT_AGE | 0x100 | MAT_SECURE_ADDR | 0x200 |
| MAT_NO_PORT | 0x400 | MAT_DROP_ADDR | 0x800 |
| MAT_DUP_ADDR | 0x1000 | MAT_NULL_DESTINATION | 0x2000 |
| MAT_DOT1X_ADDR | 0x4000 | MAT_ROUTER_ADDR | 0x8000 |
| MAT_WIRELESS_ADDR | 0x10000 | MAT_SECURE_CFG_ADDR | 0x20000 |
| MAT_OPQ_DATA_PRESENT | 0x40000 | MAT_WIRED_TUNNEL_ADDR | 0x80000 |
| MAT_DLR_ADDR | 0x100000 | MAT_MRP_ADDR | 0x200000 |
| MAT_MSRRP_ADDR | 0x400000 | MAT_LISP_LOCAL_ADDR | 0x800000 |
| MAT_LISP_REMOTE_ADDR | 0x1000000 | MAT_VPLS_ADDR | 0x2000000 |

The following is a sample output of **show bgp l2vpn vpls all** command :

```

BGP table version is 6, local router ID is 222.5.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
  r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
  x best-external, a additional-path, c RIB-compressed,
  t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1000:2128
*> 1000:2128:1.1.1.72/96
      0.0.0.0                      32768 ?
*>i 1000:2128:44.254.44.44/96
      44.254.44.44                0    100    0 ?

```



CHAPTER 9

Configuring MPLS VPN Route Target Rewrite

- [Finding Feature Information, on page 93](#)
- [Prerequisites for MPLS VPN Route Target Rewrite, on page 93](#)
- [Restrictions for MPLS VPN Route Target Rewrite, on page 93](#)
- [Information About MPLS VPN Route Target Rewrite, on page 94](#)
- [How to Configure MPLS VPN Route Target Rewrite, on page 95](#)
- [Configuration Examples for MPLS VPN Route Target Rewrite, on page 101](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Prerequisites for MPLS VPN Route Target Rewrite

- You should know how to configure Multiprotocol Label Switching (MPLS) Virtual Private Networks (VPNs).
- You need to identify the RT replacement policy and target device for the autonomous system (AS).

Restrictions for MPLS VPN Route Target Rewrite

Route Target Rewrite can only be implemented in a single AS topology.

ip unnumbered command is not supported in MPLS configuration.

Information About MPLS VPN Route Target Rewrite

Route Target Replacement Policy

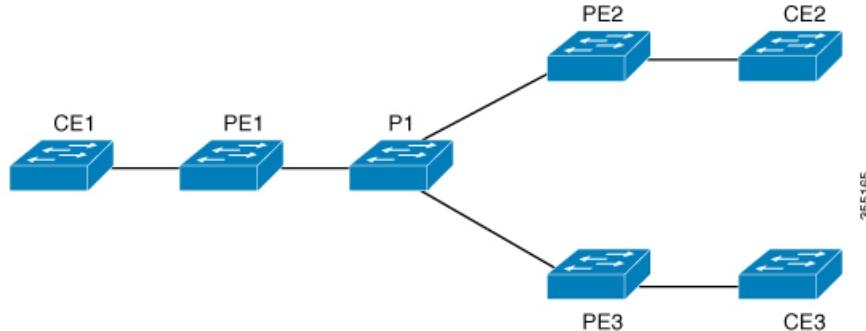
Routing policies for a peer include all configurations that may impact inbound or outbound routing table updates. The MPLS VPN Route Target Rewrite feature can influence routing table updates by allowing the replacement of route targets on inbound and outbound Border Gateway Protocol (BGP) updates. Route targets are carried as extended community attributes in BGP Virtual Private Network IP Version 4 (VPNv4) updates. Route target extended community attributes are used to identify a set of sites and VPN routing and forwarding (VRF) instances that can receive routes with a configured route target.

You can configure the MPLS VPN Route Target Rewrite feature on provider edge (PE) devices.

The figure below shows an example of route target replacement on PE devices in a Multiprotocol Label Switching (MPLS) VPN single autonomous system topology. This example includes the following configurations:

- PE1 is configured to import and export RT 65000:1 for VRF Customer A and to rewrite all inbound VPNv4 prefixes with RT 65000:1 to RT 65000:2.
- PE2 is configured to import and export RT 65000:2 for VRF Customer B and to rewrite all inbound VPNv4 prefixes with RT 65000:2 to RT 65000:1.

Figure 9: Route Target Replacement on Provide Edge(PE) devices in a single MPLS VPN Autonomous System Topology



Route Maps and Route Target Replacement

The MPLS VPN Route Target Rewrite feature extends the Border Gateway Protocol (BGP) inbound/outbound route map functionality to enable route target replacement. The `set extcomm-list delete` command entered in route-map configuration mode allows the deletion of a route target extended community attribute based on an extended community list.

How to Configure MPLS VPN Route Target Rewrite

Configuring a Route Target Replacement Policy

Perform this task to configure a route target (RT) replacement policy for your internetwork.

If you configure a provider edge (PE) device to rewrite RT x to RT y and the PE has a virtual routing and forwarding (VRF) instance that imports RT x , you need to configure the VRF to import RT y in addition to RT x .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip extcommunity-list** {*standard-list-number* | *expanded-list-number*} {**permit** | **deny**} [*regular-expression*] [**rt** | **soo** *extended-community-value*]
4. **route-map** *map-name* [**permit** | **deny**] [*sequence-number*]
5. **match extcommunity** {*standard-list-number* | *expanded-list-number*}
6. **set extcomm-list** *extended-community-list-number* **delete**
7. **set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
8. **end**
9. **show route-map** *map-name*

DETAILED STEPS

| | Command or Action | Purpose |
|--------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ip extcommunity-list { <i>standard-list-number</i> <i>expanded-list-number</i> } { permit deny } [<i>regular-expression</i>] [rt soo <i>extended-community-value</i>] Example: Device(config)# ip extcommunity-list 1 permit rt 65000:2 | Creates an extended community access list and controls access to it. <ul style="list-style-type: none"> • The <i>standard-list-number</i> argument is an integer from 1 to 99 that identifies one or more permit or deny groups of extended communities. • The <i>expanded-list-number</i> argument is an integer from 100 to 500 that identifies one or more permit or deny groups of extended communities. Regular expressions can be configured with expanded lists but not standard lists. |

| | Command or Action | Purpose |
|---------------|--|--|
| | | <ul style="list-style-type: none"> • The permit keyword permits access for a matching condition. • The deny keyword denies access for a matching condition. • The <i>regular-expression</i> argument specifies an input string pattern to match against. When you use an expanded extended community list to match route targets, include the pattern RT: in the regular expression. • The rt keyword specifies the route target extended community attribute. The rt keyword can be configured only with standard extended community lists and not expanded community lists. • The soo keyword specifies the site of origin (SOO) extended community attribute. The soo keyword can be configured only with standard extended community lists and not expanded community lists. • The <i>extended-community-value</i> argument specifies the route target or site of origin. The value can be one of the following combinations: <ul style="list-style-type: none"> • autonomous-system-number:network-number • ip-address:network-number <p>The colon is used to separate the autonomous system number and network number or IP address and network number.</p> |
| Step 4 | <p>route-map <i>map-name</i> [permit deny] [<i>sequence-number</i>]</p> <p>Example:</p> <pre>Device(config)# route-map rtrewrite permit 10</pre> | <p>Defines the conditions for redistributing routes from one routing protocol into another or enables policy routing and enables route-map configuration mode.</p> <ul style="list-style-type: none"> • The <i>map-name</i> argument defines a meaningful name for the route map. The redistribute router configuration command uses this name to reference this route map. Multiple route maps can share the same map name. • If the match criteria are met for this route map, and the permit keyword is specified, the route is redistributed as controlled by the set actions. In the case of policy routing, the packet is policy routed. <p>If the match criteria are not met, and the permit keyword is specified, the next route map with the same map tag is tested. If a route passes none of the match criteria for the set of route maps sharing the same name, it is not redistributed by that set.</p> |

| | Command or Action | Purpose |
|---------------|--|--|
| | | <p>The permit keyword is the default.</p> <ul style="list-style-type: none"> If the match criteria are met for the route map and the deny keyword is specified, the route is not redistributed. In the case of policy routing, the packet is not policy routed, and no further route maps sharing the same map tag name will be examined. If the packet is not policy routed, the normal forwarding algorithm is used. The <i>sequence-number</i> argument is a number that indicates the position a new route map will have in the list of route maps already configured with the same name. If given with the no form of this command, the position of the route map should be deleted. |
| Step 5 | <p>match extcommunity {<i>standard-list-number</i> <i>expanded-list-number</i>}</p> <p>Example:</p> <pre>Device(config-route-map)# match extcommunity 1</pre> <p>Example:</p> <pre>Device(config-route-map)# match extcommunity 101</pre> | <p>Matches the Border Gateway Protocol (BGP) extended community list attributes.</p> <ul style="list-style-type: none"> The <i>standard-list-number</i> argument is a number from 1 to 99 that identifies one or more permit or deny groups of extended community attributes. The <i>expanded-list-number</i> argument is a number from 100 to 500 that identifies one or more permit or deny groups of extended community attributes. |
| Step 6 | <p>set extcomm-list <i>extended-community-list-number</i> delete</p> <p>Example:</p> <pre>Device(config-route-map)# set extcomm-list 1 delete</pre> | <p>Removes a route target from an extended community attribute of an inbound or outbound BGP Virtual Private Network Version 4 (VPNv4) update.</p> <ul style="list-style-type: none"> The <i>extended-community-list-number</i> argument specifies the extended community list number. |
| Step 7 | <p>set extcommunity {rt <i>extended-community-value</i> [additive] soo <i>extended-community-value</i>}</p> <p>Example:</p> <pre>Device(config-route-map)# set extcommunity rt 65000:1 additive</pre> | <p>Sets BGP extended community attributes.</p> <ul style="list-style-type: none"> The rt keyword specifies the route target extended community attribute. The soo keyword specifies the site of origin extended community attribute. The <i>extended-community-value</i> argument specifies the value to be set. The value can be one of the following combinations: <ul style="list-style-type: none"> autonomous-system-number : network-number ip-address : network-number <p>The colon is used to separate the autonomous system number and network number or IP address and network number.</p> |

| | Command or Action | Purpose |
|---------------|---|---|
| | | <ul style="list-style-type: none"> The additive keyword adds a route target to the existing route target list without replacing any existing route targets. |
| Step 8 | end Example: <pre>Device(config-route-map) # end</pre> | (Optional) Returns to privileged EXEC mode. |
| Step 9 | show route-map map-name Example: <pre>Device# show route-map extmap</pre> | (Optional) Verifies that the match and set entries are correct. <ul style="list-style-type: none"> The <i>map-name</i> argument is the name of a specific route map. |

Applying the Route Target Replacement Policy

Perform the following tasks to apply the route target replacement policy to your network:

Associating Route Maps with Specific BGP Neighbors

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp as-number**
4. **neighbor {ip-address | peer-group-name} remote-as as-number**
5. **address-family vpnv4 [unicast]**
6. **neighbor {ip-address | peer-group-name} activate**
7. **neighbor {ip-address | peer-group-name} send-community [both | extended | standard]**
8. **neighbor {ip-address | peer-group-name} route-map map-name {in | out}**
9. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | enable Example: <pre>Device> enable</pre> | Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted. |
| Step 2 | configure terminal Example: <pre>Device# configure terminal</pre> | Enters global configuration mode. |

| | Command or Action | Purpose |
|--------|---|--|
| Step 3 | <p>router bgp <i>as-number</i></p> <p>Example:</p> <pre>Device(config)# router bgp 100</pre> | <p>Configures a Border Gateway Protocol (BGP) routing process and places the device in router configuration mode.</p> <ul style="list-style-type: none"> The <i>as-number</i> argument indicates the number of an autonomous system that identifies the device to other BGP devices and tags the routing information passed along. <p>The range is 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.</p> |
| Step 4 | <p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} remote-as <i>as-number</i></p> <p>Example:</p> <pre>Device(config-router)# neighbor 172.10.0.2 remote-as 200</pre> | <p>Adds an entry to the BGP or multiprotocol BGP neighbor table.</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP peer group. The <i>as-number</i> argument specifies the autonomous system to which the neighbor belongs. |
| Step 5 | <p>address-family vpnv4 [unicast]</p> <p>Example:</p> <pre>Device(config-router)# address-family vpnv4</pre> | <p>Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard Virtual Private Network Version 4 (VPNv4) address prefixes.</p> <ul style="list-style-type: none"> The optional unicast keyword specifies VPNv4 unicast address prefixes. |
| Step 6 | <p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} activate</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 172.16.0.2 activate</pre> | <p>Enables the exchange of information with a neighboring BGP device.</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP peer group. |
| Step 7 | <p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} send-community [both extended standard]</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 172.16.0.2 send-community extended</pre> | <p>Specifies that a communities attribute should be sent to a BGP neighbor.</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the BGP-speaking neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP peer group. The both keyword sends standard and extended community attributes. |

| | Command or Action | Purpose |
|---------------|---|--|
| | | <ul style="list-style-type: none"> The extended keyword sends an extended community attribute. The standard keyword sends a standard community attribute. |
| Step 8 | <p>neighbor {<i>ip-address</i> <i>peer-group-name</i>} route-map <i>map-name</i> {in out}</p> <p>Example:</p> <pre>Device(config-router-af)# neighbor 172.16.0.2 route-map extmap in</pre> | <p>Apply a route map to incoming or outgoing routes</p> <ul style="list-style-type: none"> The <i>ip-address</i> argument specifies the IP address of the neighbor. The <i>peer-group-name</i> argument specifies the name of a BGP or multiprotocol peer group. The <i>map-name</i> argument specifies the name of a route map. The in keyword applies route map to incoming routes. The out keyword applies route map to outgoing routes. |
| Step 9 | <p>end</p> <p>Example:</p> <pre>Device(config-router-af)# end</pre> | (Optional) Returns to privileged EXEC mode. |

Verifying the Route Target Replacement Policy

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 vrf** *vrf-name*
3. **exit**

DETAILED STEPS

Step 1 enable

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 show ip bgp vpnv4 vrf *vrf-name*

Verifies that Virtual Private Network Version 4 (VPNv4) prefixes with a specified route target (RT) extended community attribute are replaced with the proper RT extended community attribute to verify that the provider edge (PE) devices receive the rewritten RT extended community attributes.

Verify route target replacement on PE1:

Example:

```
Device# show ip bgp vpnv4 vrf Customer_A 192.168.1.1/32 internal
BGP routing table entry for 65000:1:192.168.1.1/32, version 6901
Paths: (1 available, best #1, table Customer_A)
  Advertised to update-groups:
    5
  Refresh Epoch 1
  650002
    3.3.3.3 (metric 3) (via default) from 3.3.3.3 (55.5.4.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:65000:1
      mpls labels in/out nolabel/3025
      rx pathid: 0, tx pathid: 0x0
      net: 0xFFB0A72E38, path: 0xFFB0E6A370, pathext: 0xFFB0E5D970
      flags: net: 0x0, path: 0x7, pathext: 0x181
```

Step 3 **exit**

Returns to user EXEC mode:

Example:

```
Device# exit
Device>
```

Configuration Examples for MPLS VPN Route Target Rewrite

Examples: Configuring Route Target Replacement Policies

This example shows the route target (RT) replacement configuration of a Provider Edge (PE) device that exchanges Virtual Private Network Version 4 (VPNv4) prefixes with another Provider Edge (PE) device. The route map `extmap` is configured to replace RTs on inbound updates. Any incoming update with RT 65000:2 is replaced with RT 65000:1.

```
!
ip extcommunity-list 1 permit rt 65000:2
!
route-map rtrewrite permit 10
match extcommunity 1
set extcomm-list 1 delete
set extcommunity rt 65000:1 additive
!
```

This example shows the use of the route-map configuration `continue` command when you need to apply more than one replacement rule on an update. In this example, an incoming update with RT 7777:22222222 is replaced with RT 65000:2. Without the `continue 20` command, route-map evaluation would stop when a match on sequence 10 is made. With the `continue 20` command, route-map evaluation continues into sequence 20 even if a match occurs in sequence 10.

```

!
ip extcommunity-list 2 permit rt 7777:222222222
ip extcommunity-list 3 permit rt 2:2
ip extcommunity-list 4 permit rt 20000:111
!
route-map extmap1 permit 10
match extcommunity 2
continue 20
set extcomm-list 2 delete
set extcommunity rt 65000:2 additive
!
route-map extmap1 permit 20
match extcommunity 3
continue 30
set extcomm-list 3 delete
!
route-map extmap1 permit 30
match extcommunity 4
set extcomm-list 4 delete
!

```



Note The route-map configuration **continue** command is not supported on outbound route maps.

Examples: Applying Route Target Replacement Policies

Examples: Associating Route Maps with Specific BGP Neighbor

This example shows the association of route map extmap with a Border Gateway Protocol (BGP) neighbor. The BGP inbound route map is configured to replace route targets (RTs) on incoming updates.

```

router bgp 1
address-family vpnv4
neighbor 2.2.2.2 route-map rtrewrite in

```

This example shows the association of the same route map with the outbound BGP neighbor. The route map is configured to replace RTs on outgoing updates.

```

router bgp 1
address-family vpnv4
neighbor 2.2.2.2 route-map rtrewrite out

```



CHAPTER 10

Configuring Multicast Virtual Private Network

- [Configuring Multicast VPN, on page 103](#)

Configuring Multicast VPN

The Multicast VPN (MVPN) feature provides the ability to support multicast over a Layer 3 VPN. As enterprises extend the reach of their multicast applications, service providers can accommodate them over their Multiprotocol Label Switching (MPLS) core network. IP multicast is used to stream video, voice, and data over an MPLS VPN network core.

Historically, point-to-point tunnels were the only way to connect through a service provider network. Although such tunneled networks tend to have scalability issues, they represented the only means of passing IP multicast traffic through a VPN.

Because Layer 3 VPNs support only unicast traffic connectivity, deploying MPLS in conjunction with a Layer 3 VPN allows service providers to offer both unicast and multicast connectivity to Layer 3 VPN customers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Prerequisites for Configuring Multicast VPN

Enable IP multicast and configure the PIM interfaces using the tasks described in the “Configuring Basic IP Multicast” module.

Restrictions for Configuring Multicast VPN

- The update source interface for the Border Gateway Protocol (BGP) peerings must be the same for all BGP peerings configured on the device in order for the default multicast distribution tree (MDT) to be

configured properly. If you use a loopback address for BGP peering, PIM sparse mode must be enabled on the loopback address.

- MVPN does not support multiple BGP peering update sources.
- Multiple BGP update sources are not supported, and configuring them can break MVPN reverse path forwarding (RPF) checking. The source IP address of the MVPN tunnels is determined by the highest IP address used for the BGP peering update source. If this IP address is not the IP address used as the BGP peering address with the remote provider edge (PE) device, MVPN will not function properly.
- **ip unnumbered** command is not supported in MPLS configuration.

Information About Configuring Multicast VPN

Multicast VPN Operation

MVPN IP allows a service provider to configure and support multicast traffic in an MPLS VPN environment. This feature supports routing and forwarding of multicast packets for each individual VRF instance, and it also provides a mechanism to transport VPN multicast packets across the service provider backbone.

A VPN is network connectivity across a shared infrastructure, such as an ISP. Its function is to provide the same policies and performance as a private network, at a reduced cost of ownership, thus creating many opportunities for cost savings through operations and infrastructure.

An MVPN allows an enterprise to transparently interconnect its private network across the network backbone of a service provider. The use of an MVPN to interconnect an enterprise network in this way does not change the way that enterprise network is administered, nor does it change general enterprise connectivity.

Benefits of Multicast VPN

- Provides a scalable method to dynamically send information to multiple locations.
- Provides high-speed information delivery.
- Provides connectivity through a shared infrastructure.

Multicast VPN Routing and Forwarding and Multicast Domains

MVPN introduces multicast routing information to the VPN routing and forwarding table. When a provider edge (PE) device receives multicast data or control packets from a customer edge (CE) router, forwarding is performed according to the information in the Multicast VPN routing and forwarding instance (MVRF). MVPN does not use label switching.

A set of MVRFs that can send multicast traffic to each other constitutes a multicast domain. For example, the multicast domain for a customer that wanted to send certain types of multicast traffic to all global employees would consist of all CE routers associated with that enterprise.

Multicast Distribution Trees

MVPN establishes a static default multicast distribution tree (MDT) for each multicast domain. The default MDT defines the path used by PE routers to send multicast data and control messages to every other PE router in the multicast domain.

If Source Specific Multicast (SSM) is used as the core multicast routing protocol, the multicast IP addresses used for the default and data MDT must be configured within the SSM range on all PE routers.

MVPN also supports the dynamic creation of MDTs for high-bandwidth transmission. Data MDTs are a feature unique to Cisco IOS software. Data MDTs are intended for high-bandwidth sources such as full-motion video inside the VPN to ensure optimal traffic forwarding in the MPLS VPN core. The threshold at which the data MDT is created can be configured on a per-router or a per-VRF basis. When the multicast transmission exceeds the defined threshold, the sending PE router creates the data MDT and sends a UDP message, which contains information about the data MDT, to all routers on the default MDT. The statistics to determine whether a multicast stream has exceeded the data MDT threshold are examined once every second. After a PE router sends the UDP message, it waits 3 more seconds before switching over; 13 seconds is the worst case switchover time, and 3 seconds is the best case.

Data MDTs are created only for (S, G) multicast route entries within the VRF multicast routing table. They are not created for (*, G) entries regardless of the value of the individual source data rate.

In the following example, a service provider has a multicast customer with offices in San Jose, New York, and Dallas. A one-way multicast presentation is occurring in San Jose. The service provider network supports all three sites associated with this customer, in addition to the Houston site of a different enterprise customer.

The default MDT for the enterprise customer consists of provider routers P1, P2, and P3 and their associated PE routers. PE4 is not part of the default MDT, because it is associated with a different customer. The figure shows that no data flows along the default MDT, because no one outside of San Jose has joined the multicast.

Figure 10: Default Multicast Distribution Tree Overview



An employee in New York joins the multicast session. The PE router associated with the New York site sends a join request that flows across the default MDT for the multicast domain of the customer. PE1, the PE router associated with the multicast session source, receives the request. The figure depicts that the PE router forwards the request to the CE router associated with the multicast source (CE1a).

Figure 11: Initializing the Data MDT



The CE router (CE1a) begins to send the multicast data to the associated PE router (PE1), which sends the multicast data along the default MDT. Immediately sending the multicast data, PE1 recognizes that the multicast data exceeds the bandwidth threshold for which a data MDT should be created. Therefore, PE1 creates a data MDT, sends a message to all routers using the default MDT, which contains information about the data MDT, and, three seconds later, begins sending the multicast data for that particular stream using the data MDT. Only PE2 has interested receivers for this source, so only PE2 will join the data MDT and receive traffic on it.

PE routers maintain a PIM relationship with other PE routers over the default MDT and a PIM relationship with directly attached PE routers.

Multicast Tunnel Interface

An MVRF, which is created per multicast domain, requires the device to create a tunnel interface from which all MVRF traffic is sourced. A multicast tunnel interface is an interface that the MVRF uses to access the multicast domain. It can be thought of as a conduit that connects an MVRF and the global MVRF. One tunnel interface is created per MVRF.

MDT Address Family in BGP for Multicast VPN

The **mdt** keyword has been added to the **address-family ipv4** command to configure an MDT address-family session. MDT address-family sessions are used to pass the source PE address and MDT group address to PIM using Border Gateway Protocol (BGP) MDT Subaddress Family Identifier (SAFI) updates.

BGP Advertisement Methods for Multicast VPN Support

In a single autonomous system, if the default MDT for an MVPN is using PIM sparse mode (PIM-SM) with a rendezvous point (RP), then PIM is able to establish adjacencies over the Multicast Tunnel Interface (MTI) because the source PE and receiver PE discover each other through the RP. In this scenario, the local PE (the source PE) sends register messages to the RP, which then builds a shortest-path tree (SPT) toward the source PE. The remote PE, which acts as a receiver for the MDT multicast group, then sends (*, G) joins toward the RP and joins the distribution tree for that group.

However, if the default MDT group is configured in a PIM Source Specific Multicast (PIM-SSM) environment rather than a PIM-SM environment, the receiver PE needs information about the source PE and the default MDT group. This information is used to send (S, G) joins toward the source PE to build a distribution tree from the source PE (without the need for an RP). The source PE address and default MDT group address are sent using BGP.

BGP Extended Community

When BGP extended communities are used, the PE loopback (source address) information is sent as a VPNv4 prefix using Route Distinguisher (RD) Type 2 (to distinguish it from unicast VPNv4 prefixes). The MDT group address is carried in a BGP extended community. Using a combination of the embedded source in the VPNv4 address and the group in the extended community, PE routers in the same MVRF instance can establish SSM trees to each other.



Note Prior to the introduction of MDT SAFI support, the BGP extended community attribute was used as an interim solution to advertise the IP address of the source PE and default MDT group before IETF standardization. A BGP extended community attribute in an MVPN environment, however, has certain limitations: it cannot be used in inter-AS scenarios (because the attribute is nontransitive), and it uses RD Type 2 (which is not a supported standard).

How to Configure Multicast VPN

Configuring the Data Multicast Group

A data MDT group can include a maximum of 256 multicast groups per VPN per VRF per PE device. Multicast groups used to create the data MDT group are dynamically chosen from a pool of configured IP addresses. Use the following procedure to configure data multicast group on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **rd** *route-distinguisher*

5. **route-target both** *ASN:nn or IP-address:nn*
6. **address family ipv4 unicast** *value*
7. **mdt default** *group-address*
8. **mdt data** *group number*
9. **mdt data threshold** *kbits*
10. **mdt log-reuse**
11. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|---------------|---|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | vrf definition <i>vrf-name</i> Example: Device(config)# vrf definition vrf1 | Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name. |
| Step 4 | rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 1:1 | Creates routing and forwarding tables for a VRF. <ul style="list-style-type: none"> • The <i>route-distinguisher</i> argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a <i>route-distinguisher</i> in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number. For example, 101:3. • 32-bit IP address: your 16-bit number. For example, 192.168.122.15:1. |
| Step 5 | route-target both <i>ASN:nn or IP-address:nn</i> Example: Device(config-vrf)# route-target both 1:1 | Creates a route-target extended community for a VRF. The both keyword specifies to import both import and export routing information to the target VPN extended community. |
| Step 6 | address family ipv4 unicast <i>value</i> Example: Device(config-vrf)# address family ipv4 unicast | Enters VRF address family configuration mode to specify an address family for a VRF. <ul style="list-style-type: none"> • The ipv4 keyword specifies an IPv4 address family for a VRF |

| | Command or Action | Purpose |
|----------------|--|--|
| Step 7 | mdt default <i>group-address</i> Example: <pre>Device(config-vrf-af)# mdt default 226.10.10.10</pre> | Configures the multicast group address range for data MDT groups for a VRF. <ul style="list-style-type: none"> • A tunnel interface is created as a result of this command. • The default MDT group address configuration must be the same on all PEs in the same VRF. |
| Step 8 | mdt data <i>group number</i> Example: <pre>Device(config-vrf-af)# mdt data 232.0.1.0 0.0.0.31</pre> | Specifies a range of addresses to be used in the data MDT pool. |
| Step 9 | mdt data threshold <i>kbps</i> Example: <pre>Device(config-vrf-af)# mdt data threshold 50</pre> | Specifies the threshold in <i>kbps</i> . The range is from 1 to 4294967. |
| Step 10 | mdt log-reuse Example: <pre>Device(config-vrf-af)# mdt log-reuse</pre> | (Optional) Enables the recording of data MDT reuse and generates a syslog message when a data MDT has been reused. |
| Step 11 | end Example: <pre>Device(config-vrf-af)# end</pre> | Returns to privileged EXEC mode. |

Configuring a Default MDT Group for a VRF

Perform this task to configure a default MDT group for a VRF.

The default MDT group must be the same group configured on all devices that belong to the same VPN. The source IP address will be the address used to source the BGP sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip multicast-routing**
4. **ip multicast-routing vrf** *vrf-name*
5. **vrf definition** *vrf-name*
6. **rd** *route-distinguisher*
7. **route-target both** *ASN:nn or IP-address:nn*
8. **address family ipv4 unicast** *value*
9. **mdt default** *group-address*
10. **end**

11. **configure terminal**
12. **ip pim vrf vrf-name rp-address value**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | ip multicast-routing Example: Device(config)# ip multicast-routing | Enables multicast routing. |
| Step 4 | ip multicast-routing vrf vrf-name Example: Device(config)# ip multicast-routing vrf vrf1 | Supports the MVPN VRF instance. |
| Step 5 | vrf definition vrf-name Example: Device(config)# vrf definition vrf1 | Enters VRF configuration mode and defines the VPN routing instance by assigning a VRF name. |
| Step 6 | rd route-distinguisher Example: Device(config-vrf)# rd 1:1 | Creates routing and forwarding tables for a VRF. <ul style="list-style-type: none"> • The <i>route-distinguisher</i> argument specifies to add an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter a <i>route-distinguisher</i> in either of these formats: • 16-bit autonomous system number (ASN): your 32-bit number. For example, 101:3. • 32-bit IP address: your 16-bit number. For example, 192.168.122.15:1. |
| Step 7 | route-target both ASN:nn or IP-address:nn Example: Device(config-vrf)# route-target both 1:1 | Creates a route-target extended community for a VRF. The both keyword specifies to import both import and export routing information to the target VPN extended community. |

| | Command or Action | Purpose |
|----------------|--|--|
| Step 8 | address family ipv4 unicast <i>value</i> Example: Device(config-vrf)# address family ipv4 unicast | Enters VRF address family configuration mode to specify an address family for a VRF. <ul style="list-style-type: none"> • The ipv4 keyword specifies an IPv4 address family for a VRF |
| Step 9 | mdt default <i>group-address</i> Example: Device(config-vrf-af)# mdt default 226.10.10.10 | Configures the multicast group address range for data MDT groups for a VRF. <ul style="list-style-type: none"> • A tunnel interface is created as a result of this command. • The default MDT group address configuration must be the same on all PEs in the same VRF. |
| Step 10 | end Example: Device(config-vrf-af)# end | Returns to privileged EXEC mode. |
| Step 11 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 12 | ip pim vrf <i>vrf-name</i> rp-address <i>value</i> Example: Device(config-vrf-af)# ip pim vrf vrf1 rp-address 1.1.1.1 | Enters the RP configuration mode. |

Configuring the MDT Address Family in BGP for Multicast VPN

Perform this task to configure an MDT address family session on PE devices to establish MDT peering sessions for MVPN.

Before you begin

Before MVPN peering can be established through an MDT address family, MPLS and Cisco Express Forwarding (CEF) must be configured in the BGP network and multiprotocol BGP on PE devices that provide VPN services to CE devices.



Note The following policy configuration parameters are not supported:

- Route-originator attribute
- Network Layer Reachability Information (NLRI) prefix filtering (prefix lists, distribute lists)
- Extended community attributes (route target and site of origin)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp *as-number***
4. **address-family ipv4 mdt**
5. **neighbor *neighbor-address* activate**
6. **neighbor *neighbor-address* send-community [both | extended | standard]**
7. **exit**
8. **address-family vpv4**
9. **neighbor *neighbor-address* activate**
10. **neighbor *neighbor-address* send-community [both | extended | standard]**
11. **end**

DETAILED STEPS

| | Command or Action | Purpose |
|--------|--|--|
| Step 1 | enable Example: Device> enable | Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted. |
| Step 2 | configure terminal Example: Device# configure terminal | Enters global configuration mode. |
| Step 3 | router bgp <i>as-number</i> Example: Device(config)# router bgp 65535 | Enters router configuration mode and creates a BGP routing process. |
| Step 4 | address-family ipv4 mdt Example: Device(config-router)# address-family ipv4 mdt | Enters address family configuration mode to create an IP MDT address family session. |
| Step 5 | neighbor <i>neighbor-address</i> activate Example: Device(config-router-af)# neighbor 192.168.1.1 activate | Enables the MDT address family for this neighbor. |
| Step 6 | neighbor <i>neighbor-address</i> send-community [both extended standard] Example: | Enables community and (or) extended community exchange with the specified neighbor. |

| | Command or Action | Purpose |
|----------------|---|---|
| | Device(config-router-af)# neighbor 192.168.1.1 send-community extended | |
| Step 7 | exit Example: Device(config-router-af)# exit | Exits address family configuration mode and returns to router configuration mode. |
| Step 8 | address-family vpnv4 Example: Device(config-router)# address-family vpnv4 | Enters address family configuration mode to create a VPNv4 address family session. |
| Step 9 | neighbor neighbor-address activate Example: Device(config-router-af)# neighbor 192.168.1.1 activate | Enables the VPNv4 address family for this neighbor. |
| Step 10 | neighbor neighbor-address send-community [both extended standard] Example: Device(config-router-af)# neighbor 192.168.1.1 send-community extended | Enables community and (or) extended community exchange with the specified neighbor. |
| Step 11 | end Example: Device(config-router-af)# end | Exits address family configuration mode and enters privileged EXEC mode. |

Verifying Information for the MDT Default Group

SUMMARY STEPS

1. enable
2. show ip pim [vrf vrf-name] mdt bgp
3. show ip pim [vrf vrf-name] mdt send
4. show ip pim vrf vrf-name mdt history interval minutes

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```


Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 `show ip pim [vrf vrf-name] mdt bgp`

Example:

```
Device# show ip pim mdt bgp

MDT-default group 232.2.1.4
rid:1.1.1.1 next_hop:1.1.1.1
```

Displays information about the BGP advertisement of the RD for the MDT default group.

Step 3 `show ip pim [vrf vrf-name] mdt send`

Example:

```
Device# show ip pim mdt send

MDT-data send list for VRF:vpn8
(source, group)                MDT-data group    ref_count
(10.100.8.10, 225.1.8.1)       232.2.8.0         1
(10.100.8.10, 225.1.8.2)       232.2.8.1         1
(10.100.8.10, 225.1.8.3)       232.2.8.2         1
(10.100.8.10, 225.1.8.4)       232.2.8.3         1
(10.100.8.10, 225.1.8.5)       232.2.8.4         1
(10.100.8.10, 225.1.8.6)       232.2.8.5         1
(10.100.8.10, 225.1.8.7)       232.2.8.6         1
(10.100.8.10, 225.1.8.8)       232.2.8.7         1
(10.100.8.10, 225.1.8.9)       232.2.8.8         1
(10.100.8.10, 225.1.8.10)      232.2.8.9         1
```

Displays detailed information about the MDT data group including MDT advertisements that the specified device has made.

Step 4 `show ip pim vrf vrf-name mdt history interval minutes`

Example:

```
Device# show ip pim vrf vrf1 mdt history interval 20

MDT-data send history for VRF - vrf1 for the past 20 minutes
MDT-data group      Number of reuse
10.9.9.8             3
10.9.9.9             2
```

Displays the data MDTs that have been reused during the past configured interval.

Configuration Examples for Multicast VPN

Example: Configuring MVPN and SSM

In the following example, PIM-SSM is configured in the backbone. Therefore, the default and data MDT groups are configured within the SSM range of IP addresses. Inside the VPN, PIM-SM is configured and only Auto-RP announcements are accepted.

```
ip vrf vrf1
```

Example: Enabling a VPN for Multicast Routing

```

rd 1:1
route-target export 1:1
route-target import 1:1
mdt default 232.0.0.1
mdt data 232.0.1.0 0.0.0.255 threshold 500 list 101
!
ip pim ssm default
ip pim vrf vrf1 accept-rp auto-rp

```

Example: Enabling a VPN for Multicast Routing

In the following example, multicast routing is enabled with a VPN routing instance named vrf1:

```
ip multicast-routing vrf1
```

Example: Configuring the Multicast Group Address Range for Data MDT Groups

In the following example, the VPN routing instance is assigned a VRF named blue. The MDT default group for a VPN VRF is 239.1.1.1, and the multicast group address range for MDT groups is 239.1.2.0 with wildcard bits of 0.0.0.3:

```

ip vrf blue
rd 55:1111
route-target both 55:1111
mdt default 239.1.1.1
mdt data 239.1.2.0 0.0.0.3
end

```

Example: Limiting the Number of Multicast Routes

In the following example, the number of multicast routes that can be added to a multicast routing table is set to 200,000 and the threshold value of the number of mroutes that will cause a warning message to occur is set to 20,000:

```

!
ip multicast-routing
ip multicast-routing vrf cisco
ip multicast cache-headers
ip multicast route-limit 200000 20000
ip multicast vrf cisco route-limit 200000 20000
no mpls traffic-eng auto-bw timers frequency 0
!

```

Additional References for Configuring Multicast VPN**Related Documents**

| Related Topic | Document Title |
|--|----------------|
| For complete syntax and usage information for the commands used in this chapter. | |

Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

Feature Information for Configuring Multicast VPN

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 13: Feature Information for Multicast VPN



INDEX

M

multicast tunnel interface [105](#)

Multicast-VPN operation [104](#)

Multicast-VPN routing and forwarding and multicast domains [104](#)

