MPLS VPNs over IP Tunnels

Part Number OL-8694-01 (Rev C0), September 26, 2006

The MPLS VPNs over IP Tunnels feature introduces the capability to deploy Layer 3 Virtual Private Network (VPN) services, as proposed in RFC 2547, *BGP/MPLS VPNs,* over an IP core network using L2TPv3 multipoint tunneling instead of Multiprotocol Label Switching (MPLS). This feature allows L2TPv3 tunnels to be configured as multipoint tunnels to transport IP VPN services across the core IP network. Because multipoint tunnels support multiple endpoints, only one tunnel must be configured on each Provider Edge (PE) router. This feature also introduces a simple packet validation mechanism to enforce VPN integrity.

Feature History for MPLS VPNs over IP Tunnels

Release	Modification
12.0(28)S	This feature was introduced.
12.0(30)S	Support for the Cisco 12000 Series Internet Router, the Route Processor (RP), and Performance Route Processor (PRP) was integrated into Cisco IOS Release 12.0(30)S.
12.0(31)S	Support for the MPLS VPNs over IP Tunnels feature was added on the Cisco 12000 series Internet router on the following interfaces: • 2.5G ISE SPA Interface Processor (SIP): – 2-Port T3/E3 Serial shared port adaptor (SPA) – 4-Port T3/E3 Serial SPA – 2-Port Channelized T3 SPA – 4-Port Channelized T3 Serial SPA Support for the MPLS VPN—Carrier Supporting Carrier (CsC) feature on interfaces configured for MPLS VPNs over IP Tunnels was added to IP Services Engine (ISE) line cards on the Cisco 12000 series Internet router.
12.0(31)S1	Support was added for Cisco 12000 series ISE line cards that are configured for external BGP (eBGP) and internal BGP (iBGP) multipath load balancing in a BGP MPLS-VPN network (see *BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN*).

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

12.0(32)S	Support for the Cisco 10720 Internet router was added.
12.0(32)SY	Support was added for Engine 5 shared port adapters (SPAs) and SPA Interface Processors (SIPs) on the Cisco 12000 series Internet router. Support for E5 interfaces configured for eBGP and iBGP multipath load sharing in a BGP MPLS network was also added (see *BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN*). Support for the MPLS VPN Carrier Supporting Carrier over IP Tunnels feature on customer-facing interfaces on the Cisco 10720 Internet router was added.

Finding Support Information for Platforms and Cisco IOS Software Images

To use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support, go to http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions.

Contents

Prerequisites for the MPLS VPNs over IP Tunnels Feature

CEF or dCEF (for distributed platforms) must be enabled on all participating routers.

Supported Line Cards for the Cisco 12000 Series Internet Router

This section lists the Cisco 12000 series Internet router line cards that support the MPLS VPNs over IP Tunnels feature with backbone-facing interfaces (BFIs) in the network core and customer-facing interfaces (CFI) on the network edge.

Supported Backbone-Facing Interfaces

- 4-port OC-3 POS ISE
- 8-port OC-3 POS ISE
- 16-port OC-3 POS ISE
- 4-port OC-12 POS ISE
- 1-port OC-48 POS ISE

- 4-port GE ISE
- IP Services Engine (ISE/Engine 3) shared port adapters (SPAs):

  2-port channelized T3 to DS0
  4-port channelized T3 to DS0
  2-port T3/E3 Serial
  4-port T3/E3 Serial
- Engine 5 shared port adapters:

  1-port channelized STM-1c/OC-3c to DS0
  2-port channelized T3 to DS0
  4-port channelized T3 to DS0
  8-port channelized T1/E1

  1-port 10-Gigabit Ethernet
  2-port Gigabit Ethernet
  5-port Gigabit Ethernet
  10-port Gigabit Ethernet
  8-port Fast Ethernet
  8-port 10/100 Ethernet

  4-port OC-3/STM4 POS
  8-port OC-3/STM4 POS
  2-port OC-12/STM4 POS
  4-port OC-12/STM4 POS
  8-port OC-12/STM4 POS
  2-port OC-48/STM16 POS/RPR
  1-port OC-192/STM64 POS/RPR VSR
  1-port OC-192/STM64 POS/RPR SMLR
  1-port OC-192/STM64 POS/RPR XFP
- SPA Interface Processors (SIPs):

  12000-SIP-400 (2.5G ISE SPA Interface Processor)
  12000-SIP-600 (10G Engine 5 SPA Interface Processor)
  12000-SIP-401 (2.5G multiservice engine SPA Interface Processor)
  12000-SIP-501 (5G multiservice engine SPA Interface Processor)
  12000-SIP-601 (10G multiservice engine SPA Interface Processor)

Supported Customer-Facing Interfaces

- 4-port OC-3 POS ISE
- 8-port OC-3 POS ISE
- 16-port OC-3 POS ISE
- 4-port OC-12 POS ISE
- 1-port OC-48 POS ISE
- 1-port channelized OC-12 (DS1) POS ISE
- 4-port OC-12 ATM ISE
- 4-port OC-3 ATM ISE
- 4-port GE ISE
- IP Services Engine (ISE/Engine 3) shared port adapters (SPAs):

  2-port channelized T3 to DS0

4-port channelized T3 to DS0
2-port T3/E3 Serial
4-port T3/E3 Serial

- Engine 5 shared port adapters:

  1-port channelized STM-1c/OC-3c to DS0
  8-port channelized T1/E1

  1-port 10-Gigabit Ethernet
  2-port Gigabit Ethernet
  5-port Gigabit Ethernet
  10-port Gigabit Ethernet
  8-port Fast Ethernet
  8-port 10/100 Ethernet

  4-port OC-3/STM4 POS
  8-port OC-3/STM4 POS
  2-port OC-12/STM4 POS
  4-port OC-12/STM4 POS
  8-port OC-12/STM4 POS
  2-port OC-48/STM16 POS/RPR
  1-port OC-192/STM64 POS/RPR
  1-port OC-192/STM64 POS/RPR
  1-port OC-192/STM64 POS/RPR

- SPA Interface Processors (SIPs):

  12000-SIP-400 (2.5G ISE SPA Interface Processor)
  12000-SIP-600 (10G Engine 5 SPA Interface Processor
  12000-SIP-401 (2.5G multiservice engine SPA Interface Processor)
  12000-SIP-501 (5G multiservice engine SPA Interface Processor)
  12000-SIP-601 (10G multiservice engine SPA Interface Processor)

Supported Line Cards for the Cisco 10720 Internet Router

This section describes the Cisco 10720 Internet router line cards that support the MPLS VPNs over IP Tunnels feature on backbone-facing and customer-facing interfaces.

Supported Backbone-Facing Interfaces on Uplink Cards

- 24-port 10/100 Fast Ethernet
- 4-port Gigabit Ethernet 8-port 10/100BASE-TX (Revision A and Revision B versions)
- 2-port OC-48c/STM-16c POS/SRP—Allows you to change Packet-over-SONET (POS) interfaces to Dynamic Packet Transport (DPT)/Spatial Reuse Protocol (SRP).
- Dual Mode IEEE 802.17 RPR/SRP—Allows you to use the OC-48c/STM-16c interfaces in either SRP or Resilient Packet Ring (RPR)-IEEE mode.

Supported Customer-Facing Interfaces on Access Cards

- 24-port 10/100 Fast Ethernet
- 4-port Gigabit Ethernet 8-port 10/100BASE-TX (Revision A and Revision B versions)

802.1Q VLAN encapsulation is supported on these backbone-facing and customer-facing 10720 interfaces when they are configured as member interfaces of a Fast EtherChannel or Gigabit EtherChannel bundle.

For more information about supported line cards, see the *Cross-Platform Release Notes for Cisco IOS Release 12.0 S*.

Restrictions for the MPLS VPNs over IP Tunnels Feature

Configuring Static Routes

When you configure static routes in an MPLS or MPLS VPN network, some variations of the **ip route** and **ip route vrf** commands are not supported for the MPLS VPNs over IP Tunnels feature in the following trains of Cisco IOS software that support the Tag Forwarding Information Base (TFIB):

- Cisco IOS Releases 12.xT
- Cisco IOS Releases 12.xM
- Cisco IOS Release 12.0S

The TFIB cannot resolve prefixes when the recursive route over which the prefixes travel disappears and then reappears. However, certain command variations are supported in Cisco IOS releases that support the MPLS Forwarding Infrastructure (MFI), such as Cisco IOS Release 12.2(25)S and later releases. Refer to Table 1 and Table 2 for the **ip route** and **ip route vrf** commands that are supported and not supported when configuring static routes in an MPLS environment.

Table 1 describes the **ip route** and **ip route vrf** commands that are supported to configure static routes in an IP core network configured for MPLS VPNs over IP Tunnels.

Table 1 Commands Supported for Configuring Static Routes in an MPLS VPNs over IP Tunnels Network

MPLS Configuration	Commands Supported to Configure Static Routes
Standard MPLS network	**ip route** *destination-prefix mask interface next-hop-address*
MPLS network in which you configure load sharing with static nonrecursive routes on an outbound interface	**ip route** *destination-prefix mask* **interface1 next-hop1** **ip route** *destination-prefix mask* **interface2 next-hop2**
MPLS VPN in which the next hop and interface are in the same Virtual Private Network (VPN) routing and forwarding (VRF) table	**ip route vrf** *vrf-name destination-prefix mask next-hop-address* **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* **ip route vrf** *vrf-name destination-prefix mask* **interface1 next-hop1** **ip route vrf** *vrf-name destination-prefix mask* **interface2 next-hop2**
MPLS VPN in which the next hop is in the global routing table in the MPLS core network (for example, the internal gateway)	**ip route vrf** *vrf-name destination-prefix mask next-hop-address* **global** **ip route vrf** *vrf-name destination-prefix mask interface next-hop-address* (This command is supported when the next hop and interface are in the core IP network.)
MPLS VPN in which you configure load sharing with static nonrecursive routes on an outbound interface	**ip route** *destination-prefix mask* **interface1 next-hop1** **ip route** *destination-prefix mask* **interface2 next-hop2**

Table 1 Commands Supported for Configuring Static Routes in an MPLS VPNs over IP Tunnels Network

MPLS Configuration	Commands Supported to Configure Static Routes
MPLS VPN in which the next hop is in the global routing table on a CE router (for example, when the IP route prefix for the destination is the loopback address of the CE router as in EBGP multihop cases)	**ip route vrf** *vrf-name destination-prefix mask interface next-hop-address*
MPLS VPN in which the next hop is in the global routing table on a CE router and in which you configure load sharing with static nonrecursive routes on an outbound interface	**ip route** *destination-prefix mask* **interface1 nexthop1** **ip route** *destination-prefix mask* **interface2 nexthop2**

Table 2 describes the **ip route** and **ip route vrf** commands that are not supported to configure static routes in an IP core network configured for MPLS VPNs over IP Tunnels.

Table 2 Commands Not Supported for Configuring Static Routes in an MPLS VPNs over IP Tunnels Network

MPLS Configuration	Commands Not Supported to Configure Static Routes
MPLS VPN in which the next hop is in the global routing table in the MPLS core network and in which you enable load sharing so that the next hop can be reached through two paths	**ip route** *destination-prefix mask next-hop-address*
MPLS VPN in which the next hop is in the global routing table in the MPLS core network and in which you enable load sharing so that the destination can be reached through two next hops	**ip route vrf** *destination-prefix mask* **next-hop1 global** **ip route vrf** *destination-prefix mask* **next-hop2 global**
MPLS VPN that uses TFIB and in which the next hop and interface are in the same VRF table	**ip route vrf** *vrf-name destination-prefix mask* **next-hop1** **ip route vrf** *vrf-name destination-prefix mask* **next-hop2**

MPLS VPNs over IP Tunnels Feature Information

Deploying Layer 3 VPNs Over Multipoint L2TPv3 Tunnels

VPN services are traditionally deployed over IP core networks by configuring MPLS or through L2TPv3 tunnels using point-to-point links. This feature introduces the capability to deploy Layer 3 VPN services by configuring multipoint L2TPv3 tunnels over an existing IP core network. This feature is configured only on PE routers and requires no configuration on the core routers.

The L2TPv3 multipoint tunnel network allows Layer 3 VPN services to be carried through the core without the configuration of MPLS. L2TPv3 multipoint tunnelling supports multiple tunnel endpoints, which creates a full mesh topology that requires only one tunnel to be configured on each PE router. This feature provides the capability for VPN traffic to be carried from enterprise networks across cooperating service provider core networks to remote sites.

Advertising Tunnel Type and Tunnel Capabilities Between PE Routers—BGP

Border Gateway Protocol (BGP) is used to advertise the tunnel endpoints and the subaddress family identifier (SAFI) specific attributes (which contains the tunnel type, and tunnel capabilities). This feature introduces the tunnel SAFI and the BGP SAFI-Specific Attribute (SSA) attribute.

The tunnel SAFI:

- Defines the tunnel endpoint and carries the endpoint IPv4 address and next hop.
- Is identified by the SAFI number 64.

The BGP SSA:

- Carries the BGP preference and BGP flags. It also carries the tunnel cookie, tunnel cookie length, and session ID.
- Is identified by attribute number 19.

These attributes allow BGP to distribute tunnel encapsulation information between PE routers. VPNv4 traffic is routed through these tunnels. The next hop, advertised in BGP VPNv4 updates, determines which tunnel to use for routing tunnel traffic.

Configuring the PE Routers and Managing Address Space

One multipoint L2TPv3 tunnel is configured on each PE router. To create the VPN, configure a unique Virtual Routing and Forwarding (VRF) instance. The tunnel that transports the VPN traffic across the core network resides in its own address space. A special purpose VRF called a Resolve in VRF (RiV) is created to manage the tunnel address space. You also configure the address space under the RiV that is associated with the tunnel and a static route in the RiV to route outgoing traffic through the tunnel.

Packet Validation Mechanism

The MPLS VPNs over IP Tunnels feature provides a simple mechanism to validate received packets from appropriate peers. The multipoint L2TPv3 tunnel header is automatically configured with a 64-bit cookie and L2TPv3 session ID. This packet validation mechanism protects the VPN from illegitimate traffic sources, such as injecting a rogue packet into the tunnel to gain access to the VPN. The cookie and session ID are not user-configurable; however, they are visible in the packet as it is routed between the two tunnel endpoints. This packet validation mechanism does not protect the VPN from hackers who have the ability to monitor legitimate traffic between PE routers.

Configuring Quality of Service Using the Modular QoS CLI

To configure the bandwidth on the encapsulation and decapsulation interfaces, use the modular QoS CLI (MQC). This task is optional.

Use the MQC to configure the IP precedence or Differentiated Services Code Point (DSCP) value set in the IP carrier header during packet encapsulation. To set these values, enter a standalone **set** command or a **police** command using the keyword **tunnel**. In the input policy on the encapsulation interface, you can set the precedence or DSCP value in the IP payload header by using MQC commands without the keyword **tunnel**.

Note You must attach a QoS policy to the physical interface—*not* to the tunnel interface.

If Modified Deficit Round Robin (MDRR)/Weighted Random Early Detection (WRED) is configured for the encapsulation interface in the input direction, the final value of the precedence or DSCP field in the IP carrier header is used to determine the precedence class for which the MDRR/WRED policy is applied. On the decapsulation interface in the input direction, you can configure a QoS policy based on the precedence or DSCP value in the IP carrier header of the received packet. In this case, an MQC policy with a class to match on precedence or DSCP value will match the precedence or DSCP value in the received IP carrier header.

Similarly, the precedence class for which the MDRR/WRED policy is applied on the decapsulation input direction is also determined by precedence or DSCP value in the IP carrier header.

MPLS VPN Carrier Supporting Carrier over IP Tunnels

You can configure the MPLS VPN—Carrier Supporting Carrier (CsC) feature on the following interfaces that are configured for the MPLS VPNs over IP Tunnels feature:

- IP Services Engine (ISE) and Engine 5 interfaces on a Cisco 12000 series Internet router
- Fast Ethernet and Gigabit Ethernet interfaces on a Cisco 10720 Internet router

The router must be deployed as a PE router in a service-provider core network. The MPLS VPN Carrier Supporting Carrier over IP Tunnels feature is supported only on customer-facing interfaces.

Note For information about the ISE and Engine 5 shared port adaptors (SPAs) and SPA interface processors (SIPs) supported on Cisco 12000 series routers, refer to the *Cisco 12000 Series Routers SPA Hardware Installation Guide*.

The MPLS VPN Carrier Supporting Carrier over IP Tunnels feature enables one MPLS VPNs over IP Tunnel-based service provider to allow other service providers to use a segment of its backbone network.

- *Backbone carrier*—Service provider that provides the segment of the backbone network to the other provider
- *Customer carrier*—Service provider that uses the segment of the backbone network

The backbone carrier benefits in the following ways:

- The backbone carrier can accommodate either IP or MPLS VPN traffic from many customer carriers (including labelled customer traffic) and give them access to its MPLS VPNs over IP Tunnels-backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one multipoint L2TPv3 tunnel for each customer carrier simplifies the backbone carrier's VPN operations.
- The MPLS VPN Carrier Supporting Carrier over IP Tunnels feature is scalable. You can change the tunnel configuration for a customer carrier to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes to enable tens of thousands of VPNs to be set up over the same MPLS VPNs over IP Tunnels network. A service provider can offer both VPN and Internet services.

The MPLS VPN—Carrier Supporting Carrier feature focuses on a backbone carrier that offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services.

When you configure the MPLS VPN—Carrier Supporting Carrier feature on a Cisco 12000 series Internet router, the customer carrier can be either:

- An Internet service provider (ISP)
- A BGP/MPLS VPN service provider

For sample configurations of the MPLS VPN—Carrier Supporting Carrier feature for each type of customer carrier, see Configuring MPLS VPN Carrier Supporting Carrier over IP Tunnels—Examples, page 33.

When configured on a Cisco 10720 Internet router, the MPLS VPN—Carrier Supporting Carrier feature supports only customer-carrier VPNs that are configured as an MPLS VPN over IP Tunnels network.

For information about configuring a backbone carrier for the Carrier Supporting Carrier feature to allow other service providers to use a segment of its backbone network, refer to:

- *MPLS VPN—Carrier Supporting Carrier* (using the Label Distribution Protocol (LDP) to carry the MPLS labels and an Internal Gateway Protocol (IGP) to carry the routes between PE and CE routers)
- *MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution* (using the Border Gateway Protocol (BGP) to transport routes and MPLS labels between the PE routers and CE routers using multiple paths)

BGP Multipath Load Sharing for MPLS VPNs over IP Tunnels

You can configure external BGP (eBGP) and internal BGP (iBGP) multipath load balancing on the following customer-facing interfaces that are configured for the MPLS VPNs over IP Tunnels feature:

- IP Services Engine (ISE) interfaces on a Cisco 12000 series Internet router
- Engine 5 interfaces on shared port adapters (SPAs) on a Cisco 12000 series Internet router

The BGP Multipath Load Sharing for eBGP and iBGP feature:

- Allows you to configure multipath load balancing with both external BGP and internal BGP paths in BGP networks that are configured to use MPLS VPNs. When faced with multiple routes to the same destination, BGP chooses the best route for routing traffic toward the destination so that no individual router is overburdened.
- Is useful for multi-homed autonomous systems and PE routers that import both eBGP and iBGP paths from multihomed and stub networks.

For information about how to configure and use BGP multipath load sharing for both eBGP and iBGP paths, refer to *BGP Multipath Load Sharing for Both eBGP and iBGP in an MPLS-VPN*.

Configuring MPLS VPNs over IP Tunnels

To deploy Layer 3 VPN services over multipoint L2TPv3 tunnels, you create a VRF instance, create the multipoint L2TPv3 tunnel, redirect the VPN IP traffic to the tunnel, and configure the BGP VPNv4 exchange so that BGP updates are filtered through a route-map and prefixes are resolved in the VRF table. The configuration steps are described in the following sections:

- Configuring a VRF for an L2TPv3 Tunnel, page 10
- Configuring a Multipoint L2TPv3 Tunnel, page 12
- Configuring a Route Map for a Layer 3 VPN, page 14
- Defining an Address Space and Configuring BGP, page 15
- Configuring Tunnel Marking on an Encapsulation Interface, page 17
- Mapping Tunnel Marking to QoS-group and Discard-Class, page 21
- Configuring MPLS VPN Carrier Supporting Carrier over IP Tunnels, page 23
- Verifying the VRF and RiV, page 26
- Verifying the Multipoint L2TPv3 Tunnel, page 27
- Verifying the Modular QoS CLI Configuration, page 28
- Verifying the MPLS VPN—Carrier Supporting Carrier Configuration, page 29

Configuring a VRF for an L2TPv3 Tunnel

The VPN is created by configuring a unique Virtual Routing and Forwarding (VRF) instance. The tunnel that transports the VPN traffic across the core network resides in its own address space. A special purpose VRF called a Resolve in VRF (RiV) is created to manage the tunnel address space.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *as-number:network-number* | *ip-address:network number*
5. **route-target import** *as-number:network-number* | *ip-address:network number*
6. **route-target export** *as-number:network-number* | *ip-address:network number*
7. **exit**
8. **ip vrf** *vrf-name*
9. **rd** *as-number:network-number* | *ip-address:network number*
10. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`ip vrf vrf-name`
	
	Example:
	`Router(config)# ip vrf CUSTOMER_A`
		Creates a VRF routing table and specifies the VRF name (or tag).
		• The **ip vrf** command creates a VRF routing table and a CEF table, which are both named using the *vrf-name* argument. Associated with these tables is the default route-distinguisher value.

Step 4	`rd as-number:network-number	ip-address:network-number`  **Example:** `Router(config-vrf)# rd 100:110`	Creates routing and forwarding tables for the VRF instance created in Step 3. • There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).
Step 5	`route-target [export	import	both] as-number:network-number	ip-address:network-number`  **Example:** `Router(config-vrf)# route-target import 100:1000`	Imports routing information from the target VPN extended community.

	Command or Action	Purpose

Step 6	`route-target [export | import | both] as-number:network-number | ip-address:network-number`
	
	Example:
	`Router(config-vrf)# route-target export 100:1000`
		Exports routing information to the target VPN extended community.

Step 7	`exit`
	
	Example:
	`Router(config-vrf)# exit`
		Exits VRF configuration mode and enters global configuration mode.

Step 8	`ip vrf vrf-name`
	
	Example:
	`Router(config)# ip vrf MY_RIV`
		Creates the special Resolve in VRF (RiV) VRF instance and table that will be used for the tunnel and redirection of the IP address.
		• Creates a VRF routing table and specifies the VRF name (or tag). The **ip vrf** command creates a VRF routing table and a CEF table; both are named using the vrf-name argument. Associated with these tables is the default route-distinguisher value.

Step 9	`rd as-number:network-number | ip-address:network-number`
	
	Example:
	`Router(config-vrf)# rd 1:1`
		Creates routing and forwarding tables for the VRF instance created in Step 8.
		• There are two formats for configuring the route distinguisher argument. It can be configured in the as-number:network number (ASN:nn) format, as shown in the example, or it can be configured in the IP address:network number format (IP-address:nn).

Step 10	`end`
	
	Example:
	`Router(config-vrf)# end`
		Exits VRF configuration mode and enters privileged EXEC mode.

Proceed to the next task "Configuring a Multipoint L2TPv3 Tunnel."

Configuring a Multipoint L2TPv3 Tunnel

Border Gateway Protocol (BGP) is used to advertise the tunnel type, tunnel capabilities, and tunnel-specific attributes. BGP is also used to distribute VPNv4 routing information between PE routers on the edge of the network, which maintains peering relationships between the VPN service and VPN sites. The next hop advertised in BGP VPNv4 updates triggers tunnel endpoint discovery.

Prerequisites

The IP address of the interface, specified as the tunnel source, should match the IP address used by BGP as the next hop for the VPNv4 update. The BGP configuration will include the **neighbor** *ip-address* **update-source loopback 0** command.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **interface tunnel** *interface-number*
4. **ip vrf forwarding** *RiV-name*
5. **ip address** *ip-address subnet-mask*
6. **tunnel source loopback** *interface-number*
7. **tunnel mode l3vpn l2tpv3 multipoint**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel interface-number** **Example:** `Router(config)# interface tunnel 1`	Enters interface configuration mode, and creates the tunnel.
Step 4	**ip vrf forwarding RiV-name** **Example:** `Router(config-if)# ip vrf forwarding MY_RIV`	Associates the VRF with an interface or the subinterface. • The RiV name is configured for the VRF argument in this step.
Step 5	**ip address ip-address subnet-mask** **Example:** `Router(config-if)# ip-address 172.16.1.3 255.255.255.255`	Specifies the IP address for the tunnel.
Step 6	**tunnel source loopback interface-number** **Example:** `Router(config-if)# tunnel source loopback 0`	Associates the tunnel source IP address with the loopback interface.

	Command or Action	Purpose
Step 7	**tunnel mode l3vpn l2tpv3 multipoint** **Example:** Router(config-if)# **tunnel mode l3vpn l2tpv3 multipoint**	Sets the mode for the Layer 3 VPN tunnel as l2tpv3 multipoint.
Step 8	**end** **Example:** Router(config-if)# **end**	Exits interface configuration mode and enters privileged EXEC mode.

Troubleshooting

To generate and distribute a new L2TPv3 session for a Layer 3 VPN, use the **clear tunnel l3vpn l2tpv3** command. This command is issued on the PE router. The *hold-time* argument is used to configure the amount of time that the existing session remains valid, while the new session is propagated to peers. The default value for the *hold-time* argument is 15 seconds. This is enough time for most networks. However, this value can be increased if it takes longer for the new session to propagate to all other PE routers.

Proceed to the next task "Configuring a Route Map for a Layer 3 VPN."

Configuring a Route Map for a Layer 3 VPN

Configure a route map to set the next hop to be resolved within the VRF table.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **route-map** *map-name*
4. **set ip next-hop in-vrf** *RiV-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> **enable**	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** Router# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

Step 3	`route-map map-name [permit|deny] [sequence-number]`
	
	Example:
	`Router(config)# route-map SELECT_UPDATE_FOR_L3VPN permit 10`
		Enters route-map configuration mode, and creates a route-map.

Step 4	`set ip next-hop in-vrf RiV-name`
	
	Example:
	`Router(config-route-map)# set ip next-hop in-vrf MY_RIV`
		Specifies that the next hop is to be resolved in the VRF table for the specified VRF.
		• The RiV is configured for the VRF argument in this step.

Step 5	`end`
	
	Example:
	`Router(config-if)# end`
		Exits route-map configuration mode and enters privileged EXEC mode.

Proceed to the next task "Defining an Address Space and Specifying Address Resolution."

Defining an Address Space and Configuring BGP

Use the configuration task procedure in this section to set up the BGP VPNv4 exchange so that the updates are filtered through a route-map and interesting prefixes are resolved in the VRF table. The tunnel that transports the VPN traffic across the BGP core network resides in its own address space. The RiV is specified in this configuration to direct packet forwarding and next hop resolution.

SUMMARY STEPS

1. **enable**
2. **configure** {**terminal** | **memory** | **network**}
3. **ip route vrf** *riv-vrf-name 0.0.0.0 0.0.0.0* **tunnel** *interface-number*
4. **router bgp** *as-number*
5. **neighbor** *ip-address* | *peer-group-name* **remote-as** *as-number*
6. **neighbor** *ip-address* | *peer-group-name* **update-source** *interface-type*
7. **address-family vpnv4** [**unicast**]
8. **neighbor** *ip-address* | *peer-group-name* **activate**
9. **neighbor** *ip-address* | *peer-group-name* **route-map** *map-name* {**in** | **out**}
10. **exit-address-family**
11. **address-family ipv4** [**tunnel**]
12. **neighbor** *ip-address* | *peer-group-name* **activate**
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router>` **`enable`**	Enables higher privilege levels, such as privileged EXEC mode. Enter your password if prompted.
Step 2	**configure** {**terminal** | **memory** | **network**} **Example:** `Router#` **`configure terminal`**	Enters global configuration mode.
Step 3	**ip route vrf** *riv-vrf-name* **0.0.0.0 0.0.0.0 tunnel** *n* **Example:** `Router(config)#` **`ip route vrf MY_RIV 0.0.0.0 0.0.0.0 tunnel 1`**	Sets the packet forwarding to the Resolve-in-VRF (RiV). **Note** A 0.0.0.0 0.0.0.0 default route must be configured for the RiV. If the default route is not configured, the next hop may not be resolvable.
Step 4	**router bgp** *as-number* **Example:** `Router (config)#` **`router bgp 100`**	Specifies the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)#` **`neighbor 172.16.1.2 remote-as 100`**	Adds an entry to the BGP or multiprotocol BGP neighbor table.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type* **Example:** `Router(config-router)#` **`neighbor 172.16.1.2 update-source Loopback 0`**	Specifies a specific operational interface that BGP sessions use for TCP connections.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)#` **`address-family vpnv4 unicast`**	Specifies address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)#` **`neighbor 172.16.1.2 activate`**	Enables the exchange of information with a neighboring router. Use the **neighbor activate** command in address family configuration or router configuration mode

	Command or Action	Purpose
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**} **Example:** `Router(config-router-af)# neighbor 172.16.1.2 route-map SELECT_UPDATE_FOR_L3VPN in`	Applies a route map to incoming or outgoing routes. Use once for each inbound route.
Step 10	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode, and enters router configuration mode.
Step 11	**address-family ipv4** [**tunnel**] **Example:** `Router(config-router)# address-family ipv4 tunnel`	Enter address family configuration mode for the IPv4 tunnel SAFI. • The configuration of this SAFI allows BGP to advertise the tunnel endpoints and SAFI specific attribute (which contains the tunnel type and the tunnel capabilities) between the PE routers. **Note** Redistribution is enabled automatically within this SAFI.
Step 12	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 172.16.1.2 activate`	Enables the exchange of information with a neighboring router. Use the **neighbor activate** command in address family configuration or router configuration mode
Step 13	**end** **Example:** `Router(config-router-af)# end`	Exits address-family configuration mode and enters privileged EXEC mode.

Proceed to the next task "Configuring Tunnel Marking on an Encapsulation Interface."

Configuring Tunnel Marking on an Encapsulation Interface

QoS can optionally be configured to control bandwidth. As part of encapsulation, the precedence or Differentiated Services Code Point (DSCP) value in the IP carrier header can be set using the MQC. This can be achieved by configuring a standalone set action or by configuring a policing action using the keyword **tunnel**. In the input policy on the encapsulation interface, the precedence or DSCP value in the IP payload header can be set using MQC commands without using the keyword **tunnel**. Sample configurations appear below.

Note The policy must be attached to the physical interface—*not* the tunnel interface.

If Modified Deficit Round Robin (MDRR)/Weighted Random Early Detection (WRED) is configured for the encapsulation interface in the input direction, the final value of the precedence or DSCP field in the IP carrier header is used to determine the precedence class for which the MDRR/WRED policy is applied.

Release 12.0(32)SY adds scaled QoS capabilities on the Cisco 12000 platform by manipulating the tunnel header on the ingress PE. This will allow you to provide transparent QoS services by deliver end-to-end QoS enabled MVPN services,

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **set ip dscp tunnel** *dscp-value*

   or

   set ip precedence tunnel *precedence-value*

   or

   police *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action*
6. **exit**
7. **exit**
8. **interface** *type number* [*name-tag*]
9. **description** *string*
10. **ip vrf forwarding** *vrf-name*
11. **ip address** *ip-address mask* [*secondary*]
12. **service-policy** {**input** | **output**} *policy-map-name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

Step 3	`policy-map` *policy-map-name* **Example:** `Router(config)# policy-map set_prec_tun`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters policy-map configuration mode. • Enter the policy map name.

	Command or Action	Purpose

Step 4	`class {class-name | class-default}`
	
	Example:
	`Router(config-pmap)# class class-default`
		Specifies the name of the class whose policy you want to create or change, or specifies the default class (commonly known as the class-default class) before you configure its policy. Also enters policy-map class mode.
		• Enter the class name or enter the **class-default** keyword.

Step 5	`set ip dscp tunnel dscp-value`
	
	Example:
	`Router(config-pmap-c)# set ip dscp tunnel 2`
		(Optional) Sets or marks the differentiated services code point (DSCP) value in the tunnel header on the ingress interface. The tunnel marking value is a number from 0 to 63 when configuring DSCP.
		• Enter the tunnel value.

	or
	`set ip precedence tunnel precedence-value`
	
	Example:
	`Router(config-pmap-c)# set ip precedence tunnel 2`
		(Optional) Sets or marks the IP precedence value in the tunnel header on the ingress interface. The tunnel marking value is a number from 0 to 7 when configuring IP precedence.
		• Enter the tunnel value.

	or
	`police bps [burst-normal] [burst-max] conform-action action exceed-action action`
	
	Example:
	`Router(config-pmap-c)# police 1280000 conform-action set-dscp-tunnel-transmit 3 exceed-action drop`
	
	or
	`Router(config-pmap-c)# police cir 1280000 conform-action set-prec-tunnel-transmit 3 exceed-action drop`
		(Optional) Configures traffic policing on the basis of the bits per second (bps) specified and the actions specified.
		
		If you use traffic policing in your network, you can implement the L2TPv3 tunnel marking feature with the **set-dscp-tunnel-transmit** or **set-prec-tunnel-transmit** traffic policing commands instead of the **set ip dscp tunnel** or the **set ip precedence tunnel** commands shown in Step 5.
		
		The tunnel marking value for the traffic policing commands is from 0 to 63 when using **set-dscp-tunnel-transmit** and from 0 to 7 when using **set-prec-tunnel-transmit**.
		• Enter the bps, any optional burst sizes, and the desired conform and exceed actions.
		• Enter the **set-dscp-tunnel-transmit** or **set-prec-tunnel-transmit** commands after the **conform-action** keyword.
		
		Note This is an example of one QoS feature you can configure at this step. Other QoS features include Weighted Random Early Detection (WRED), Weighted Fair Queueing (WFQ), and traffic shaping. Enter the command for the specific QoS feature you want to configure. For more information about QoS features, refer to *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.3.

Step 6	`exit`
	
	Example:
	`Router(config-pmap-c)# exit`
		Exits policy-map class configuration mode and enters policy-map configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config-pmap)#` **`exit`**	Exits policy-map configuration mode and enters global configuration mode.
Step 8	**interface type number [name-tag]** **Example:** `Router(config)#` **`interface POS0/0`**	Configures the interface type specified and enters interface configuration mode. • Enter interface type.
Step 9	**description string** **Example:** `Router(config-if)#` **`description IP VPN Encapsulation - Customer Facing`**	Adds a description to the interface configuration.
Step 10	**ip vrf forwarding vrf-name** **Example:** `Router(config-if)#` **`ip vrf forwarding IP_VPN`**	Associates the VRF with an interface or the subinterface.
Step 11	**ip address ip-address mask [secondary]** **Example:** `Router(config-if)#` **`ip address 192.168.123.4 255.255.255.0`**	Sets a primary or secondary IP address for an interface.
Step 12	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)#` **`service-policy input set_prec_tun`**	Specifies the name of the policy map to be attached to the *input or output* direction of the interface. • Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. • Enter the **input** keyword followed by the policy map name. **Note** For this feature, only the incoming interface configured with the **input** keyword is supported.
Step 13	**end** **Example:** `Router(config-if)#` **`end`**	(Optional) Exits interface configuration mode, and enters privileged EXEC mode.

Proceed to the next task “Mapping Tunnel Marking to Qos-group and Discard-Class.”

Mapping Tunnel Marking to QoS-group and Discard-Class

On the decapsulation interface in the input direction, the QoS policy can be constructed based on the IP precedence or DSCP value in the IP carrier header of the received packet. In this case, an MQC policy with a class to match on precedence or DSCP value matches the IP precedence or DSCP value in the received IP carrier header.

Similarly, the IP precedence class (for which the MDRR/WRED policy is applied on the decapsulation input direction) is also determined by the IP precedence or DSCP value in the IP carrier header. Qos-group and discard-class values can then be used to construct an output policy for the Decapsulation interface to configure Modified Deficit Round Robin (MDRR)/Weighted Random Early Detection (WRED).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **set qos-group** {*group-id*}
6. **set discard-class** *value*
7. **exit**
8. **exit**
9. **interface** *type number* [*name-tag*]
10. **description** *string*
11. **ip address** *ip-address mask* [*secondary*]
12. **no ip directed-broadcast** [*access-list-number*] | [*extended access-list-number*]
13. **service-policy** {**input** | **output**} *policy-map-name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)#` **`policy-map set_qos_disc_from_prec`**	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters policy-map configuration mode. • Enter the policy map name.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)#` **`class match_prec1`**	Specifies the name of the class whose policy you want to create or change, or specifies the default class (commonly known as the class-default class) before you configure its policy. Also enters policy-map class mode. • Enter the class name or enter the **class-default** keyword. • A class policy is defined for each class group.
Step 5	**set qos-group {group-id}** **Example:** `Router(config-pmap-c)#` **`set qos-group 1`**	Sets a QoS group identifier (ID) that can be used to classify packets.
Step 6	**set discard-class value** **Example:** `Router(config-pmap-c)#` **`set discard-class 1`**	Marks a packet with a discard-class value.
Step 7	**exit** **Example:** `Router(config-pmap-c)#` **`exit`**	Exits policy-map class configuration mode and enters policy-map configuration mode.
Step 8	**exit** **Example:** `Router(config-pmap)#` **`exit`**	Exits policy-map configuration mode and enters global configuration mode.
Step 9	**interface type number [name-tag]** **Example:** `Router(config)#` **`interface POS1/0`**	Configures the interface type specified and enters interface configuration mode. • Enter interface type.
Step 10	**description string** **Example:** `Router(config-if)#` **`description IP VPN Decapsulation - Backbone Facing`**	Adds a description to the interface configuration.
Step 11	**ip directed-broadcast [access-list-number] | [extended access-list-number]** **Example:** `Router(config-if)#` **`no ip directed-broadcast`**	Enables the translation of a directed broadcast to physical broadcasts. • This configuration is designed to disable this functionality.

	Command or Action	Purpose
Step 12	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 192.168.234.5 255.255.255.0`	Sets a primary or secondary IP address for an interface.
Step 13	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input set_qos_disc_from_prec`	Specifies the name of the policy map to be attached to the *input or output* direction of the interface. • Policy maps can be configured on ingress or egress routers. They can also be attached in the input or output direction of an interface. The direction (input or output) and the router (ingress or egress) to which the policy map should be attached varies according your network configuration. • Enter the **input** keyword followed by the policy map name. **Note** For this feature, only the incoming interface configured with the **input** keyword is supported.
Step 14	**end** **Example:** `Router(config-if)# end`	(Optional) Exits interface configuration mode, and enters privileged EXEC mode.

Proceed to the sections, Verifying the VRF and RiV, page 26 and Verifying the Multipoint L2TPv3 Tunnel, page 27, to verify the configuration of the MPLS-VPNs over IP Tunnels feature.

Configuring MPLS VPN Carrier Supporting Carrier over IP Tunnels

To enable a backbone carrier to share its backbone network, configured for the MPLS VPNs over IP Tunnels feature with a customer carrier, you must perform the following tasks:

1. Configure the PE router in the MPLS VPNs over IP Tunnels backbone-carrier network.
2. Configure the CE router in the customer-carrier network that links to the edge router of the backbone carrier.

This section describes how to perform each task. For more detailed information about the configuration procedure and command syntaxes, refer to the "Configuration Tasks" section in the *MPLS VPN—Carrier Supporting Carrier* and *MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution* documents.

Prerequisites

- The PE routers of the backbone carrier require 128 MB of memory.
- The backbone carrier must enable the PE router to check that the packets it receives from the CE router contain only the labels that the PE router advertised to the CE router. This prevents data spoofing, which occurs when a packet from an unrecognized IP address is sent to a router.

- A routing protocol is required between the PE and CE routers that connect the backbone carrier to the customer carrier. The routing protocol enables the customer carrier to exchange IGP routing information with the backbone carrier. Use the same routing protocol that the customer carrier uses. RIP, OSPF, eBGP, and static routing are supported as the routing protocol.
- To connect the backbone carrier to the customer carrier, one of the following combinations of routing protocols is required on the PE and CE routers that connect the backbone carrier to the customer carrier:
  - IGP and LDP
  - eBGP and labels
- All PE routers that link the backbone carrier to the customer carrier must run this Cisco IOS software image. Other PE routers, CE routers, and P routers do not need to run this software image, but they must run a version of Cisco IOS software that supports MPLS VPNs.
- Every packet that crosses the backbone carrier must be encapsulated, so that the packet includes MPLS labels. To ensure that the packets are encapsulated, you must enter the mpls ip command on each PE router that connects to a CE router.
- The following features are not supported in the MPLS VPN—Carrier Supporting Carrier feature:
  - ATM MPLS
  - Carrier-supporting-carrier traffic engineering
  - Carrier-supporting-carrier class of service (CoS)
  - RSVP aggregation
  - VPN Multicast between the customer carrier and the backbone carrier network

Configuring the Backbone-Carrier PE Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **mpls ip**
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Sets the default label distribution protocol for all interfaces to be LDP.
Step 4	**mpls ip** **Example:** `Router(config)# mpls ip`	Enables MPLS on the VRF interface of the PE router in the backbone-carrier network, as configured in Configuring a VRF for an L2TPv3 Tunnel, page 10.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits global configuration mode, and enters privileged EXEC mode.

Configuring the Customer-Carrier PE Router

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls label protocol ldp**
4. **mpls ip**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**mpls label protocol ldp** **Example:** `Router(config)# mpls label protocol ldp`	Sets the default label distribution protocol for all interfaces to be LDP.

	Command or Action (continued)	Purpose
Step 4	**mpls ip** **Example:** Router(config)# **mpls ip**	Enables MPLS on the VRF interface of the CE router in the customer-carrier network, as configured in Configuring a VRF for an L2TPv3 Tunnel, page 10.
Step 5	**end** **Example:** Router(config-if)# **end**	Exits global configuration mode, and enters privileged EXEC mode.

1. Proceed to Verifying the MPLS VPN—Carrier Supporting Carrier Configuration, page 29 to verify the status of LDP sessions that were configured for the Carrier Supporting Carrier feature between the backbone carrier and customer carrier. The customer-carrier sites should appear as a VPN customer to the backbone carrier.configured for the MPLS-VPNs over IP Tunnels feature.
2. Then proceed to the sections, Verifying the VRF and RiV, page 26 and Verifying the Multipoint L2TPv3 Tunnel, page 27, to verify the configuration of the MPLS-VPNs over IP Tunnels feature.

Verifying the VRF and RiV

Use the following steps to verify the configuration of the VRF and RiV.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4 vrf** *vrf-name*
3. **show ip route vrf** *vrf-name*
4. **show ip cef vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Router> **enable**	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [*ip-prefix/length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**] **Example:** Router# **show ip bgp vpnv4 vrf MY_RIV**	Displays VPN address information from the BGP table. This command is used to verify that the specified VRF has been received by BGP. The BGP table entry should show that the route-map has worked and that the next hop is showing in the RiV. If the VRF route is not in the BGP VRF, reconfigure the VRF and route distinguisher.

	Command or Action	Purpose

Step 3	`show ip route vrf vrf-name [connected] [protocol [as-number] [tag] [output-modifiers]] [ip-prefix] [list number [output-modifiers]] [profile] [static [output-modifiers]] [summary [output-modifiers]] [supernets-only [output-modifiers]] [traffic-engineering [output-modifiers]]`
	
	Example:
	`Router# show ip route vrf MY_RIV`
		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. If the VRF is in the routing table but the PE router still cannot be reached using the **ping** command, reconfigure the VRF and route distinguisher.

Step 4	`show ip cef vrf vrf-name [ip-prefix [mask [longer-prefixes]] [detail] [output-modifiers]] [interface interface-number] [adjacency] [interface interface-number] [detail] [discard] [drop] [glean] [null] [punt] [output-modifiers]] [detail [output-modifiers]] [non-recursive [detail] [output-modifiers]] [summary [output-modifiers]] [traffic [prefix-length] [output-modifiers]] [unresolved [detail] [output-modifiers]]`  **Example:** `Router# show ip cef vrf MY_RIV`	Displays the CEF forwarding table associated with VRF that was configured for the VPN. This command is used to verify that the correct VRF routes are in the CEF table. If the VRF route is not in the CEF table, reconfigure the VRF and route distinguisher.

Verifying the Multipoint L2TPv3 Tunnel

Use the following steps to verify the configuration of the multipoint L2TPv3 tunnel.

SUMMARY STEPS

1. **enable**
2. **show interface** *interface*
3. **show l2tun**
4. **show tunnel endpoint** *vrf-name*
5. **show ip bgp ipv4 tunnel** [*ip-address* | **summary**]

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`show interface interface`
	
	Example:
	`Router# show interface Tunnel 1`
		Displays the information about the specified interface.
		• This command used to verify that the tunnel interface is correctly configured and functioning properly.

	Command or Action	Purpose

Step 3	`show l2tun`
	
	Example:
	`Router# show l2tun`
		Displays the state of L2TPv3 tunnels and currently configured sessions.
		• This command is used to verify tunnel and session information.

Step 4	`show tunnel endpoint interface`
	
	Example:
	`Router# show tunnel endpoint`
		Displays source and destination information for tunnel endpoints.
		• This command is used to verify that the tunnel endpoints were created correctly.

Step 5	`show ip bgp ipv4 tunnel [ip-address	summary]`  **Example:** `Router# show ip bgp ipv4 tunnel summary`	Displays "tunnel" SAFI specific information. • This command is used to verify the tunnel type, tunnel capabilities, tunnel-specific attributes, and tunnel endpoints.

Verifying the Modular QoS CLI Configuration

To verify that this feature is configured as intended and that either the IP precedence or DSCP value is set as expected, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **show policy-map interface** *interface-name*

   or

   show policy-map *policy-map*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`show policy-map interface` *`interface-name`* **Example:** `Router# show policy-map interface s4/0`	(Optional) Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface. • Enter the interface name.

	Command or Action	Purpose
	or **show policy-map** *policy-map* **Example:** `Router# show policy-map qos3`	(Optional) Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps. • Enter a policy map name.
Step 3	**exit** **Example:** `Router# exit`	Exits privileged EXEC mode.

Troubleshooting Tips

Use the commands in the "Verifying the Modular QoS CLI Configuration" section to verify that you achieved the intended configuration and that the feature is functioning correctly. If, after using the **show** commands listed above, you find that the configuration is not correct or the feature is not functioning as expected, perform these operations:

If the configuration is not the one you intended, complete the following procedures:

- Use the **show running-config** command and analyze the output of the command.
- If the policy map does not appear in the output of the **show running-config** command, enable the **logging console** command. Attach the policy map to the interface again.

Verifying the MPLS VPN—Carrier Supporting Carrier Configuration

To verify the MPLS VPN—Carrier Supporting Carrier feature on interfaces configured for MPLS VPNs over IP Tunnels on PE routers in a backbone carrier network:

1. Follow the procedure in this section to verify the status of LDP sessions between the backbone carrier and customer carrier configured for the Carrier Supporting Carrier feature.
2. Follow the procedures in Verifying the VRF and RiV, page 26 and Verifying the Multipoint L2TPv3 Tunnel, page 27 to verify the configuration of the MPLS-VPNs over IP Tunnels feature.

To verify the status of LDP sessions between the backbone carrier and customer carrier configured for the MPLS VPN—Carrier Supporting Carrier feature, complete the following steps. The customer-carrier sites should appear as a VPN customer to the backbone carrier.configured for the MPLS VPNs over IP Tunnels feature.

SUMMARY STEPS

1. **enable**
2. **show mpls ldp discovery vrf** *vpn-name*

   or

   show mpls ldp discovery all
3. **exit**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	**show mpls ldp discovery vrf** *vpn-name* **Example:** `Router# show mpls ldp discovery vrf vpn1`	(Optional) Displays the neighbor discovery information for the specified VPN routing/forwarding instance (*vpn-name*). • Enter the name of the VRF RiV instance created in Configuring a VRF for an L2TPv3 Tunnel, page 10 for the L2TPv3 tunnel.
	or **show mpls ldp discovery all** **Example:** `Router# show mpls ldp discovery all`	(Optional) Displays LDP discovery information for all VPNs, including those in the default routing domain.

Step 3	`exit`
	
	Example:
	`Router# exit`
		Exits privileged EXEC mode.

Configuration Examples for MPLS VPNs over IP Tunnels

- Configuring the VRF and RiV—Example, page 31
- Configuring the Multipoint L2TPv3 Tunnel—Example, page 31
- Configuring a Route Map for the Layer 3 VPN—Example, page 31
- Defining Address Space and Configuring BGP—Example, page 31
- Configuring Tunnel Marking on the Encapsulation Interface—Example, page 31
- Mapping Tunnel Marking to QoS-group and Discard-Class—Example, page 32
- Configuring MPLS VPN Carrier Supporting Carrier over IP Tunnels—Examples, page 33
- Verifying the VRF—Example, page 49
- Verifying the Multipoint L2TPv3 Tunnel—Examples, page 50
- Verifying QoS Configuration—Example, page 51
- Verifying MPLS VPN Carrier Supporting Carrier over IP Tunnels—Example, page 52

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

Configuring the VRF and RiV—Example

The following sample configuration creates and configures the VRF and RiV:

```
ip vrf vrf-name
 rd 100:110
 route-target import 100:1000
 route-target export 100:1000
 exit
ip vrf MY_RIV
 rd 1:1
 end
```

Configuring the Multipoint L2TPv3 Tunnel—Example

The following sample configuration creates and configures the L2TPv3 tunnel:

```
interface tunnel 1
 ip vrf forwarding MY_RIV
 ip-address 172.16.1.3 255.255.255.255
 tunnel source loopback 0
 tunnel mode l3vpn l2tpv3 multipoint
 end
```

Configuring a Route Map for the Layer 3 VPN—Example

The following sample configuration creates an inbound route map to set the next hop to be resolved within the VRF:

```
route-map SELECT_UPDATE_FOR_L3VPN permit 10
 set ip next-hop in-vrf MY_RIV
 end
```

Defining Address Space and Configuring BGP—Example

The following sample configuration defines address space for the VPN and configures BGP:

```
ip route vrf MY_RIV 0.0.0.0 0.0.0.0 tunnel 1
router bgp 100
 neighbor 172.16.1.2 remote-as 100
 neighbor 172.16.1.2 update-source Loopback 0
 address-family vpnv4 unicast
 neighbor 172.16.1.2 activate
 neighbor 172.16.1.2 route-map SELECT_UPDATE_FOR_L3VPN in
 exit-address-family
 address-family ipv4 tunnel
 neighbor 176.16.1.2 activate
 end
```

Configuring Tunnel Marking on the Encapsulation Interface—Example

The following examples show how to configure QoS to control bandwidth. These examples show how to configure IP precedence or DSCP using individual set actions or by configuring policing actions.

In the following example, a policy map named "set_prec_tun" is created and the IP precedence is configured in the policy map. You could use the **set ip dscp tunnel** command instead of the **set ip precedence tunnel** command if you do not use IP precedence in your network.

```
policy-map set_prec_tun
 class class-default
  set ip precedence tunnel 2
  set ip precedence 2
```

In the following example, the IP precedence is configured as a policing action using the **police** command. Like the previous example, DSCP or IP precedence can be configured.

```
policy-map policer_prec_tun
 class class-default
  police cir 1280000 conform-action set-prec-tunnel-transmit 3 exceed-action drop
```

The following example attaches the policy map to the interface. This step is required regardless of which method is used to configure IP precedence or DSCP.

```
interface POS0/0
 description IP VPN Encapsulation - Customer Facing
 ip vrf forwarding IP_VPN
 ip address 192.168.123.4 255.255.255.0
 service-policy input set_prec_tun
```

Mapping Tunnel Marking to QoS-group and Discard-Class—Example

The following example configures the policy on the decapsulation interface in the input direction. The QoS policy can be constructed based on the precedence or DSCP value in the IP carrier header of the received packet. The MQC policy is configured to match on the IP precedence or DSCP value received in the IP carrier header.

```
policy-map set_qos_disc_from_prec
 class match_prec1
  set qos-group 1
  set discard-class 1
 class match_prec2
  set qos-group 2
  set discard-class 2
 class match_prec3
  set qos-group 3
  set discard-class 3
 class class-default
  set discard-class 4
  set qos-group 4
```

The following example attaches the policy map to the interface:

```
interface POS1/0
 description IP VPN Decapsulation - Backbone Facing
 ip address 192.168.234.5 255.255.255.0
 no ip directed-broadcast
 ip router isis
 service-policy input set_qos_disc_from_prec
```

Configuring MPLS VPN Carrier Supporting Carrier over IP Tunnels—Examples

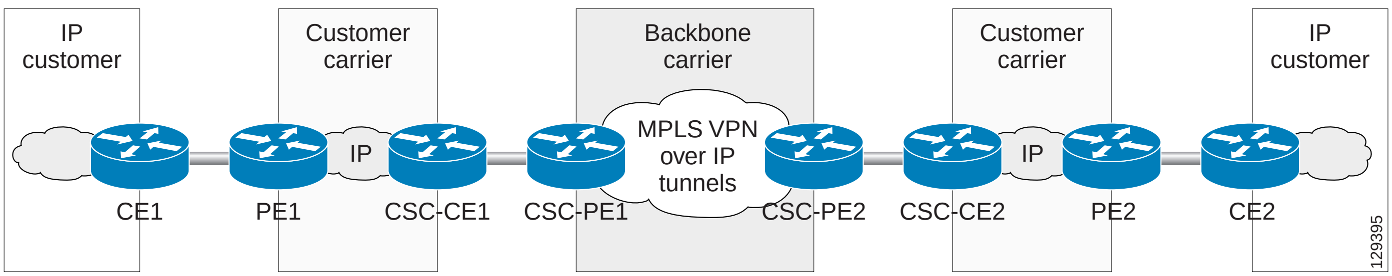
Figure 1 shows examples of how to configure the Carrier Supporting Carrier (CsC) feature on the interfaces of supported ISE and Engine 5 line cards in the PE routers of a backbone carrier (configured for the MPLS VPNs over IP Tunnels feature) and on the CE routers in one of the following types of customer carrier network:

- Internet service provider (ISP) with IP networks
- MPLS VPN customer carrier with MPLS/IP networks

Configuration for an ISP Customer Carrier

The following example shows how to configure the MPLS VPN Carrier Supporting Carrier over IP Tunnels feature for an ISP customer carrier to interconnect IP networks over a backbone MPLS VPNs over IP Tunnels network.

Figure 1 MPLS VPN Carrier Supporting Carrier over IP Tunnels for an ISP Customer Carrier



In this example, the following conditions apply:

- The customer carrier exchanges external IPv4 routes directly.
- The customer carrier exchanges internal IPv4 routes (and labels) with the backbone carrier.
- The backbone carrier exchanges customer carrier internal IPv4 routes as VPNv4 routes.
- The backbone carrier exchanges IPv4 tunnel endpoints internally.

For the MPLS VPN Carrier Supporting Carrier over IP Tunnels configuration shown in Figure 1, you must configure the two PE routers (CSC_PE1 and CSC-PE2) in the backbone carrier network and the two CE routers (CSC-CE1 and CSC-CE2) in the customer carrier networks, as described in the following sections.

CSC-CE1 Configuration

```
hostname csc-ce1

mpls label protocol ldp

interface Loopback0
 ip address 10.100.2.2 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache

interface Loopback1
 ip address 10.101.2.2 255.255.255.255
 no ip directed-broadcast
```

```
 no ip route-cache

interface Loopback2
 ip address 10.102.2.2 255.255.255.255
 no ip directed-broadcast

interface POS2/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no keepalive
 tag-switching ip
 crc 32
 clock source internal

interface POS2/0.100 point-to-point
 ip address 192.168.80.4 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 100

interface POS2/0.200 point-to-point
 ip address 192.168.80.5 255.255.255.0
 no ip directed-broadcast
 mpls bgp forwarding
 frame-relay interface-dlci 200

interface POS2/0.300 point-to-point
 ip address 192.168.80.6 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 300

interface POS2/0.400 point-to-point
 ip address 192.168.80.11 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 400

interface POS6/3
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no keepalive
 crc 32
 clock source internal

interface POS6/3.100 point-to-point
 ip address 192.168.70.4 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 70

interface POS6/3.200 point-to-point
 ip address 192.168.70.5 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 200

interface POS6/3.300 point-to-point
 ip address 192.168.70.6 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 300
```

```
router ospf 16
 log-adjacency-changes
 redistribute connected subnets
 network 192.168.70.4.0.0.0.255 area 100
 network 192.168.80.4.0.0.0.255 area 100

router ospf 26
 log-adjacency-changes
 redistribute connected subnets
 redistribute bgp 200 subnets
 network 192.168.70.5.0.0.0.255 area 200
 network 192.168.80.5.0.0.0.255 area 200

router ospf 3
 log-adjacency-changes
 redistribute connected subnets
 network 192.168.70.6.0.0.0.255 area 3
 network 192.168.80.6.0.0.0.255 area 3
 network 10.102.2.2 0.0.0.0 area 3

router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.80.5 remote-as 100
 neighbor 10.101.1.1 remote-as 200
 neighbor 10.101.1.1 update-source Loopback1
 neighbor 10.201.1.1 remote-as 200
 neighbor 10.201.1.1 update-source Loopback1
 neighbor 10.201.2.2 remote-as 200
 neighbor 10.201.2.2 update-source Loopback1

 address-family ipv4
 redistribute ospf 26
 neighbor 192.168.80.5 activate
 neighbor 192.168.80.5 send-label
 no neighbor 10.101.1.1 activate
 no neighbor 10.201.1.1 activate
 no neighbor 10.201.2.2 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family vpnv4
 neighbor 10.101.1.1 activate
 neighbor 10.101.1.1 send-community extended
 neighbor 10.201.1.1 activate
 neighbor 10.201.1.1 send-community extended
 neighbor 10.201.2.2 activate
 neighbor 10.201.2.2 send-community extended
 exit-address-family
```

CSC-PE1 Configuration

```
hostname csc-pe1

ip vrf forwarding

ip vrf my_riv
 rd 1:1

ip vrf vpn100
 rd 100:100
 route-target export 100:100
 route-target import 100:100
```

```
ip vrf vpn200
 rd 200:200
 route-target export 200:200
 route-target import 200:200

ip vrf vpn300
 rd 300:300
 route-target export 300:300
 route-target import 300:300

mpls label protocol ldp

interface Loopback0
 ip address 10.127.80.80 255.255.255.255
 no ip directed-broadcast
 ip router isis
 no ip route-cache

interface Tunnel100
 bandwidth 10000
 ip vrf forwarding my_riv
 ip address 192.168.1.1 255.255.255.255
 no ip redirects
 no ip directed-broadcast
 tunnel source Loopback0
 tunnel mode l3vpn l2tpv3 multipoint

interface POS4/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no keepalive
 mpls label protocol ldp
 tag-switching ip
 crc 32
 clock source internal

interface POS4/0.100 point-to-point
 ip vrf forwarding vpn100
 ip address 192.168.80.4 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 100

interface POS4/0.200 point-to-point
 ip vrf forwarding vpn200
 ip address 192.168.80.5 255.255.255.0
 no ip directed-broadcast
 mpls bgp forwarding
 frame-relay interface-dlci 200

interface POS4/0.300 point-to-point
 ip vrf forwarding vpn300
 ip address 192.168.80.6 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 300

interface POS5/0
 ip address 192.168.90.4 255.255.255.0
 no ip directed-broadcast
```

```
 ip router isis
 no keepalive
 crc 32
 clock source internal

router ospf 16 vrf vpn100
 log-adjacency-changes
 redistribute bgp 100 subnets
 network 192.168.80.4.0.0.0.255 area 100

router ospf 3 vrf vpn300
 log-adjacency-changes
 redistribute bgp 100 subnets
 network 192.168.80.6.0.0 0.255 area 3

router isis
 net 49.0001.0000.0000.000a.00

router bgp 100
 bgp log-neighbor-changes
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.10.10.10 update-source Loopback0
 neighbor 10.20.20.20 remote-as 100
 neighbor 10.20.20.20 update-source Loopback0
 neighbor 10.30.30.30 remote-as 100
 neighbor 10.30.30.30 update-source Loopback0

 address-family ipv4
 no neighbor 10.10.10.10 activate
 no neighbor 10.20.20.20 activate
 no neighbor 10.30.30.30 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 tunnel
 neighbor 10.10.10.10 activate
 neighbor 10.20.20.20 activate
 neighbor 10.30.30.30 activate
 exit-address-family

 address-family vpnv4
 neighbor 10.10.10.10 activate
 neighbor 10.10.10.10 send-community both
 neighbor 10.10.10.10 route-map rmap1 in
 neighbor 10.20.20.20 activate
 neighbor 10.20.20.20 send-community both
 neighbor 10.20.20.20 route-map rmap1 in
 neighbor 10.30.30.30 activate
 neighbor 10.30.30.30 send-community both
 neighbor 10.30.30.30 route-map rmap1 in
 exit-address-family

 address-family ipv4 vrf vpn300
 redistribute connected
 redistribute ospf 3 vrf vpn300
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf vpn200
 neighbor 192.168.80.5 remote-as 200
 neighbor 192.168.80.5 activate
 neighbor 192.168.80.5 as-override
```

```
 neighbor 192.168.80.5 send-label
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf vpn100
 redistribute connected
 redistribute ospf 16 vrf vpn100
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf my_riv
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf forwarding
 no auto-summary
 no synchronization
 exit-address-family

ip route vrf my_riv 0.0.0.0 0.0.0.0 Tunnel100

route-map rmap1 permit 10
 set ip next-hop in-vrf my_riv
```

CSC-CE2 Configuration

```
hostname csc-ce2

mpls label protocol ldp
no mpls traffic-eng auto-bw timers frequency 0

interface Loopback1
 ip address 10.201.2.2 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache

interface POS12/0
 ip address 192.168.110.4 255.255.255.0
 no ip directed-broadcast
 no keepalive
 mpls ldp discovery transport-address 11.11.11.11
 mpls label protocol ldp
 tag-switching ip
 crc 32
 clock source internal

router ospf 17
 log-adjacency-changes
 redistribute connected subnets
 network 192.168.110.4.0.0.0.255 area 100

router bgp 200
 bgp log-neighbor-changes
 neighbor 10.101.1.1 remote-as 200
 neighbor 10.101.1.1 update-source Loopback1
 neighbor 10.101.2.2 remote-as 200
 neighbor 10.101.2.2 update-source Loopback1
 neighbor 10.201.1.1 remote-as 200
 neighbor 10.201.1.1 update-source Loopback1
```

```
 address-family ipv4
 no neighbor 10.101.1.1 activate
 no neighbor 10.101.2.2 activate
 no neighbor 10.201.1.1 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family vpnv4
 neighbor 10.101.1.1 activate
 neighbor 10.101.1.1 send-community extended
 neighbor 10.101.2.2 activate
 neighbor 10.101.2.2 send-community extended
 neighbor 10.201.1.1 activate
 neighbor 10.201.1.1 send-community extended
 exit-address-family

ip classless
ip route 192.168.80.8 255.255.255.0 110.8.0.1
ip route 192.168.80.11 255.255.255.0 110.11.0.1
```

CSC-PE2 Configuration

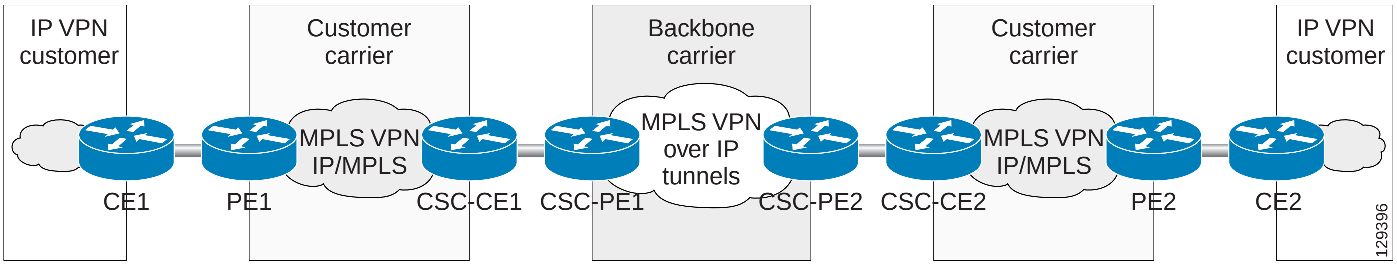
```
hostname csc-pe2

ip vrf forwarding

ip vrf my_riv
 rd 1:1

ip vrf vpn100
 rd 100:100
 route-target export 100:100
 route-target import 100:100

mpls label protocol ldp

interface Loopback0
 ip address 10.127.10.10 255.255.255.255
 no ip directed-broadcast
 ip router isis

interface Loopback1
 ip vrf forwarding vpn100
 ip address 10.127.12.12 255.255.255.255
 no ip directed-broadcast

interface Tunnel100
 ip vrf forwarding my_riv
 ip address 192.168.3.3 255.255.255.255
 no ip redirects
 no ip directed-broadcast
 tunnel source Loopback0
 tunnel mode l3vpn l2tpv3 multipoint

interface POS6/0
 ip address 192.168.100.4 255.255.255.0
 no ip directed-broadcast
 ip router isis
 crc 32
 clock source internal

interface POS8/0
 ip vrf forwarding vpn100
```

```
 ip address 192.168.110.4 255.255.255.0
 no ip directed-broadcast
 no keepalive
 mpls ldp discovery transport-address 12.12.12.12
 mpls label protocol ldp
 tag-switching ip
 crc 32
 clock source internal

router ospf 17 vrf vpn100
 log-adjacency-changes
 redistribute connected subnets
 redistribute bgp 100 subnets
 network 192.168.110.4.0.0.0.255 area 100

router isis
 net 49.0001.0000.0000.000c.00

router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 10.127.80.80 remote-as 100
 neighbor 10.127.80.80 update-source Loopback0
 no auto-summary

 address-family ipv4 tunnel
 neighbor 10.127.80.80 activate
 exit-address-family

 address-family vpnv4
 neighbor 10.127.80.80 activate
 neighbor 10.127.80.80 send-community both
 neighbor 10.127.80.80 route-map rmap1 in
 exit-address-family

 address-family ipv4 vrf vpn400
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf vpn100
 redistribute connected
 redistribute ospf 17 vrf vpn100
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf my_riv
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf forwarding
 no auto-summary
 no synchronization
 exit-address-family

ip route vrf my_riv 0.0.0.0 0.0.0.0 Tunnel100

route-map rmap1 permit 10
 set ip next-hop in-vrf my_riv
```

Configuration for an MPLS VPN Customer Carrier

Figure 2 shows how to configure the MPLS VPN Carrier Supporting Carrier over IP Tunnels feature for an MPLS VPN customer carrier to interconnect MPLS/IP networks over a backbone MPLS VPNs over IP Tunnels network.

Figure 2 MPLS VPN Carrier Supporting Carrier over IP Tunnels for an MPLS VPN Customer Carrier



In this example, the following conditions apply:

- The customer carrier exchanges external VPNv4 routes directly.
- The customer carrier exchanges internal IPv4 routes (and labels) with the backbone carrier.
- The backbone carrier exchanges customer carrier internal IPv4 routes as VPNv4 routes.
- The backbone carrier exchanges IPv4 tunnel endpoints internally.

For the MPLS VPN Carrier Supporting Carrier over IP Tunnels configuration shown in Figure 2, you must configure the two PE routers (CSC-PE1 and CSC-PE2) in the backbone carrier network and the two CE routers (CSC-CE1 and CSC-CE2) in the customer carrier networks, as described in the following sections.

CSC-CE1 Configuration

```
hostname csc-ce1

mpls label protocol ldp

interface Loopback0
 ip address 10.127.100.2 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache

interface Loopback1
 ip address 10.127.101.2 255.255.255.255
 no ip directed-broadcast
 no ip route-cache

interface Loopback2
 ip address 10.127.102.2 255.255.255.255
 no ip directed-broadcast

interface POS2/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no keepalive
 tag-switching ip
 crc 32
```

```
 clock source internal

interface POS2/0.100 point-to-point
 ip address 192.168.80.4 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 100

interface POS2/0.200 point-to-point
 ip address 192.168.80.5 255.255.255.0
 no ip directed-broadcast
 mpls bgp forwarding
 frame-relay interface-dlci 200

interface POS2/0.300 point-to-point
 ip address 192.168.80.6 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 300

interface POS2/0.400 point-to-point
 ip address 192.168.80.11 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 400

interface POS6/3
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no keepalive
 crc 32
 clock source internal

interface POS6/3.100 point-to-point
 ip address 192.168.70.4 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 70

interface POS6/3.200 point-to-point
 ip address 192.168.70.5 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 200

interface POS6/3.300 point-to-point
 ip address 192.168.70.6 255.255.255.0
 no ip directed-broadcast
 frame-relay interface-dlci 300

router ospf 16
 log-adjacency-changes
 redistribute connected subnets
 network 192.168.70.4.0.0.0.255 area 100
 network 192.168.80.4.0.0 0.255 area 100

router ospf 26
 log-adjacency-changes
 redistribute connected subnets
 redistribute bgp 200 subnets
 network 192.168.70.5.0.0.0.255 area 200
 network 192.168.80.5.0.0.0.255 area 200
```

```
router ospf 3
 log-adjacency-changes
 redistribute connected subnets
 network 192.168.70.6.0.0.0.255 area 3
 network 192.168.80.6.0.0.0.255 area 3
 network 10.102.2.2 0.0.0.0 area 3

router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.80.5.remote-as 100
 neighbor 10.101.1.1 remote-as 200
 neighbor 10.101.1.1 update-source Loopback1
 neighbor 10.201.1.1 remote-as 200
 neighbor 10.201.1.1 update-source Loopback1
 neighbor 10.201.2.2 remote-as 200
 neighbor 10.201.2.2 update-source Loopback1

 address-family ipv4
 redistribute ospf 26
 neighbor 192.168.80.5.activate
 neighbor 192.168.80.5 send-label
 no neighbor 10.101.1.1 activate
 no neighbor 10.201.1.1 activate
 no neighbor 10.201.2.2 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family vpnv4
 neighbor 10.101.1.1 activate
 neighbor 10.101.1.1 send-community extended
 neighbor 10.201.1.1 activate
 neighbor 10.201.1.1 send-community extended
 neighbor 10.201.2.2 activate
 neighbor 10.201.2.2 send-community extended
 exit-address-family
```

CSC-PE1 Configuration

```
hostname csc-pe1

ip vrf forwarding

ip vrf my_riv
 rd 1:1

ip vrf vpn100
 rd 100:100
 route-target export 100:100
 route-target import 100:100

ip vrf vpn200
 rd 200:200
 route-target export 200:200
 route-target import 200:200

ip vrf vpn300
 rd 300:300
 route-target export 300:300
 route-target import 300:300

mpls label protocol ldp
```

```
interface Loopback0
 ip address 10.127.80.80 255.255.255.255
 no ip directed-broadcast
 ip router isis
 no ip route-cache

interface Tunnel100
 bandwidth 10000
 ip vrf forwarding my_riv
 ip address 192.168.1.1 255.255.255.255
 no ip redirects
 no ip directed-broadcast
 tunnel source Loopback0
 tunnel mode l3vpn l2tpv3 multipoint

interface POS4/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no keepalive
 mpls label protocol ldp
 tag-switching ip
 crc 32
 clock source internal

interface POS4/0.100 point-to-point
 ip vrf forwarding vpn100
 ip address 192.168.80.4 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 100

interface POS4/0.200 point-to-point
 ip vrf forwarding vpn200
 ip address 192.168.80.5 255.255.255.0
 no ip directed-broadcast
 mpls bgp forwarding
 frame-relay interface-dlci 200

interface POS4/0.300 point-to-point
 ip vrf forwarding vpn300
 ip address 192.168.80.6 255.255.255.0
 no ip directed-broadcast
 mpls label protocol ldp
 tag-switching ip
 frame-relay interface-dlci 300

interface POS5/0
 ip address 192.168.90.4 255.255.255.0
 no ip directed-broadcast
 ip router isis
 no keepalive
 crc 32
 clock source internal

router ospf 16 vrf vpn100
 log-adjacency-changes
 redistribute bgp 100 subnets
 network 192.168.80.4.0.0.0.255 area 100

router ospf 3 vrf vpn300
 log-adjacency-changes
 redistribute bgp 100 subnets
```

```
 network 192.168.80.6.0.0.0.255 area 3

router isis
 net 49.0001.0000.0000.000a.00

router bgp 100
 bgp log-neighbor-changes
 neighbor 10.10.10.10 remote-as 100
 neighbor 10.10.10.10 update-source Loopback0
 neighbor 10.20.20.20 remote-as 100
 neighbor 10.20.20.20 update-source Loopback0
 neighbor 10.30.30.30 remote-as 100
 neighbor 10.30.30.30 update-source Loopback0

 address-family ipv4
 no neighbor 10.10.10.10 activate
 no neighbor 10.20.20.20 activate
 no neighbor 10.30.30.30 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 tunnel
 neighbor 10.10.10.10 activate
 neighbor 10.20.20.20 activate
 neighbor 10.30.30.30 activate
 exit-address-family

 address-family vpnv4
 neighbor 10.10.10.10 activate
 neighbor 10.10.10.10 send-community both
 neighbor 10.10.10.10 route-map rmap1 in
 neighbor 10.20.20.20 activate
 neighbor 10.20.20.20 send-community both
 neighbor 10.20.20.20 route-map rmap1 in
 neighbor 10.30.30.30 activate
 neighbor 10.30.30.30 send-community both
 neighbor 10.30.30.30 route-map rmap1 in
 exit-address-family

 address-family ipv4 vrf vpn300
 redistribute connected
 redistribute ospf 3 vrf vpn300
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf vpn200
 neighbor 192.168.80.5.remote-as 200
 neighbor 192.168.80.5 activate
 neighbor 192.168.80.5 as-override
 neighbor 192.168.80.5 send-label
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf vpn100
 redistribute connected
 redistribute ospf 16 vrf vpn100
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf my_riv
```

```
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf forwarding
 no auto-summary
 no synchronization
 exit-address-family

ip route vrf my_riv 0.0.0.0 0.0.0.0 Tunnel100

route-map rmap1 permit 10
 set ip next-hop in-vrf my_riv
```

CSC-CE2 Configuration

```
hostname csc-ce2

interface Loopback0
 ip address 10.127.200.2 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache

interface Loopback1
 ip address 10.127.201.2 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache

interface Loopback2
 ip address 10.127.202.2 255.255.255.255
 no ip directed-broadcast

interface Loopback3
 ip address 10.127.203.2 255.255.255.255
 no ip directed-broadcast

interface Loopback4
 ip address 10.127.204.2 255.255.255.255
 no ip directed-broadcast

interface Loopback5
 ip address 10.127.11.11 255.255.255.255
 no ip directed-broadcast

interface POS2/0
 ip address 192.168.110.5 255.255.255.0
 no ip directed-broadcast
 no keepalive
 mpls bgp forwarding
 tag-switching ip
 crc 32
 clock source internal

router ospf 27
 log-adjacency-changes
 redistribute connected subnets
 redistribute bgp 200 subnets
 network 192.168.110.5.0.0.0.255 area 200
 network 192.168.120.5.0.0.0.255 area 200
 network 10.201.2.2 0.0.0.0 area 200

router bgp 200
```

```
 bgp log-neighbor-changes
 neighbor 10.101.1.1 remote-as 200
 neighbor 10.101.1.1 update-source Loopback1
 neighbor 10.101.2.2 remote-as 200
 neighbor 10.101.2.2 update-source Loopback1
 neighbor 192.168.110.5 remote-as 100
 neighbor 192.168.110.7 remote-as 100
 neighbor 10.201.1.1 remote-as 200
 neighbor 10.201.1.1 update-source Loopback1

 address-family ipv4
 redistribute eigrp 27
 redistribute ospf 27
 no neighbor 10.101.1.1 activate
 no neighbor 10.101.2.2 activate
 neighbor 192.168.110.5 activate
 neighbor 192.168.110.5 send-label
 neighbor 192.168.110.7 activate
 neighbor 192.168.110.7 send-label
 no neighbor 10.201.1.1 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family vpnv4
 neighbor 10.101.1.1 activate
 neighbor 10.101.1.1 send-community extended
 neighbor 10.101.2.2 activate
 neighbor 10.101.2.2 send-community extended
 neighbor 10.201.1.1 activate
 neighbor 10.201.1.1 send-community extended
 exit-address-family

 address-family ipv4 vrf vpn201
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family

ip classless
ip route 192.168.80.8 255.255.255.0 110.8.0.1
ip route 192.168.80.11 255.255.255.0 110.11.0.1
```

CSC-PE2 Configuration

```
hostname csc-pe2-2

ip vrf forwarding

ip vrf my_riv
 rd 1:1

ip vrf vpn200
 rd 200:200
 route-target export 200:200
 route-target import 200:200

interface Loopback0
 ip address 10.127.20.20 255.255.255.255
 no ip directed-broadcast
 ip router isis
 no ip route-cache
 no ip mroute-cache
```

```
interface Tunnel100
 bandwidth 10000
 ip vrf forwarding my_riv
 ip address 192.168.5.5 255.255.255.255
 no ip redirects
 no ip directed-broadcast
 tunnel source Loopback0
 tunnel mode l3vpn l2tpv3 multipoint

interface POS3/0
 ip vrf forwarding vpn200
 ip address 192.168.110.5 255.255.255.0
 no ip directed-broadcast
 no keepalive
 mpls bgp forwarding
 crc 32
 clock source internal

interface POS6/0
 ip address 192.168.100.5 255.255.255.0
 no ip directed-broadcast
 ip router isis
 crc 32
 clock source internal

router isis
 net 49.0001.0000.0000.000d.00

router bgp 100
 bgp log-neighbor-changes
 neighbor 10.127.80.80 remote-as 100
 neighbor 10.127.80.80 update-source Loopback0

 address-family ipv4
 no neighbor 10.127.80.80 activate
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 tunnel
 neighbor 10.127.80.80 activate
 exit-address-family

 address-family vpnv4
 neighbor 10.127.80.80 activate
 neighbor 10.127.80.80 send-community both
 neighbor 10.127.80.80 route-map rmap1 in
 exit-address-family

 address-family ipv4 vrf vpn400
 redistribute connected
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf vpn200
 neighbor 192.168.110.5 remote-as 200
 neighbor 192.168.110.5 activate
 neighbor 192.168.110.5 as-override
 neighbor 192.168.110.5 send-label
 neighbor 192.168.110.7 remote-as 200
 neighbor 192.168.110.7 activate
 neighbor 192.168.110.7 as-override
 neighbor 192.168.110.7 send-label
```

```
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf my_riv
 no auto-summary
 no synchronization
 exit-address-family

 address-family ipv4 vrf forwarding
 no auto-summary
 no synchronization
 exit-address-family

ip route vrf my_riv 0.0.0.0 0.0.0.0 Tunnel100
```

Verifying the VRF—Example

Use the **show ip bgp vpnv4**, **show ip route vrf**, and **show ip cef vrf** commands to verify that VRF and RiV are configured correctly propagating to the appropriate routing and forwarding tables.

Verify that the specified VRF prefix has been received by BGP. The BGP table entry should show that the route-map has worked and that the next hop is showing in the RiV. Use the **show ip bgp vpnv4** command as shown in this example:

```
Router# show ip bgp vpnv4 vrf vrf-name 10.10.10.4
BGP routing table entry for 100:1:10.10.10.4/24, version 12
Paths: (1 available, best #1)
  Not advertised to any peer
  Local
    172.16.1.2 in "vrf-name" from 172.16.1.2 (172.16.1.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Extended Community: RT:100:1
```

Use the **show ip route vrf** command to confirm that the same information has been propagated to the routing table:

```
Router# show ip route vrf vrf-name 10.10.10.4
Routing entry for 10.10.10.4/24
  Known via "bgp 100", distance 200, metric 0, type internal
  Last update from 172.16.1.2 00:23:07 ago
  Routing Descriptor Blocks:
  * 172.16.1.2 (vrf-name), from 172.16.1.2, 00:23:07 ago
      Route metric is 0, traffic share count is 1
      AS Hops 0
```

Use the **show ip cef vrf** command to verify that the same information has been propagated to the CEF forwarding table:

```
Router# show ip cef vrf vrf-name
Prefix              Next Hop             Interface
0.0.0.0/0           attached             Tunnel1
0.0.0.0/32          receive
10.10.10.4/32       10.10.10.4           Tunnel1
172.16.1.2/32       receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive

Router# show ip cef vrf CUSTOMER_A
Prefix              Next Hop             Interface
0.0.0.0/0           drop                 Null0 (default route handler entry)
```

```
0.0.0.0/32          receive
192.168.0.0/8       10.10.10.4          Tunnel1
10.0.4.0/24         10.10.10.4          Tunnel1
10.0.6.0/24         attached            Serial2/0
10.0.6.0/32         receive
10.0.6.1/32         receive
10.0.6.255/32       receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32  receive
```

Verifying the Multipoint L2TPv3 Tunnel—Examples

Use the **show interface**, **show l2tun**, and **show tunnel endpoint** commands to verify the configuration of the, tunnel interface, L2TPv3 tunnel and tunnel endpoints.

Use the **show interface** command, as show in the display, to verify that the tunnel interface is up and configured correctly:

```
Router# show interface Tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.2/32
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.10.10.6 (Loopback0)
  Tunnel protocol/transport Multi-L2TPv3 (L3VPN), sequencing disabled
  Transporting l3vpn traffic to routes recursing through "MY_RIV"
  Key disabled
  Checksumming of packets disabled,  fast tunneling enabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Use the **show l2tun** command, as shown in the display, to verify tunnel and session information:

```
Router# show l2tun
 Tunnel and Session Information Total tunnels 0 sessions 1

L3VPN Session Information Total sessions 1

LocID      Cookie
1025       0xC0DEE550DEADBEEF
```

Use the **show tunnel endpoint** command, as shown in the display to verify that the tunnel endpoints were created correctly:

```
Router# show tunnel endpoint
 Tunnel1 running in Multi-L2TPv3 (L3VPN) mode
  RFC2547/L3VPN Tunnel endpoint discovery is active on Tu1
  Transporting l3vpn traffic to all routes recursing through "MY_RIV"
```

```
 Endpoint 10.10.10.4 via destination 10.10.10.4
```

Use the **show ip bgp ipv4 tunnel** command, as shown in the display to verify tunnel specific information configured under the “tunnel” SAFI:

```
Router# show ip bgp ipv4 tunnel

BGP table version is 3, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.3.3.3/32       0.0.0.0                 0         32768 ?
*>i10.5.5.5/32       10.5.5.5                0    100      0 ?
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 402,cookie_high D0338947,cookie_low 69DCF79E
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 401,cookie_high 6FB30F92,cookie_low A7E61105
```

Verifying QoS Configuration—Example

This section contains sample output from the **show policy-map interface** command and the **show policy-map** command. The output from these commands can be used to verify and monitor the feature configuration in your network.

The following is sample output from the **show policy-map interface** command. In this sample output, the character string “set precedence tunnel 6” indicates that the tunnel marking has been configured to set the IP precedence in the header of tunneled packets.

```
Router# show policy-map interface

POS0/0.1

  Service-policy input: tunnel (1196)

    Class-map: frde (match-any) (1197/18)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: fr-de  (1198)
      Set Policy:
        set precedence tunnel 6

    Class-map: class-default (match-any) (1200/0)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any  (1201)
      Set Policy:
        set precedence tunnel 3
```

The following is sample output from the **show policy-map** command. In this sample output, the character string “ip precedence tunnel 4” indicates that the tunnel marking on L2TPv3 feature has been configured to set the IP precedence in the header of an L2TPv3 tunneled packet.

```
Router# show policy-map

Policy Map TUNNEL_MARKING
    Class MATCH_FRDE
      set ip precedence tunnel 4
```

Verifying MPLS VPN Carrier Supporting Carrier over IP Tunnels—Example

This section shows how to use the **show mpls ldp discovery** command to display the status of the label distribution protocol (LDP) sessions between a PE router in the backbone-carrier (MPLS VPNs over IP Tunnels) network and the CE router in a customer carrier network.

The following example shows the LDP sessions in VRF Customer A of the PE router of the backbone carrier.

```
Router# show mpls ldp discovery vrf customer_a

Local LDP Identifier:
    139.0.0.0:0
Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 55.0.0.1:0
        POS6/0 (ldp): xmit
```

The next example shows how to display all LDP sessions that are active on the router.

```
Router# show mpls ldp discovery all
Local LDP Identifier:
    141.141.141.141:0
Discovery Sources:
    Interfaces:
        Ethernet1/5 (ldp): xmit/recv
            LDP Id: 5.5.5.5:0
VRF vpn1: Local LDP Identifier:
    139.0.0.1:0
Discovery Sources:
    Interfaces:
        Ethernet1/0 (ldp): xmit/recv
            LDP Id: 55.0.0.1:0
        POS6/0 (ldp): xmitLocal LDP Identifier:
```

The Local LDP Identifier field shows the LDP identifier for the local label switching router for this session. The Interfaces field displays the interfaces configured for the Carrier Supporting Carrier feature that are performing LDP discovery activity:

- xmit indicates that the interface is transmitting LDP discovery hello packets.
- recv indicates that the interface is receiving LDP discovery hello packets.

Additional References

For additional information related to this feature, refer to the following references:

Related Documents

Related Topic	Document Title
CEF switching	*Cisco IOS Switching Services Configuration Guide*, Release 12.3
QoS—Tunnel Marking	*QoS: Tunnel Marking for L2TPv3 Tunnels*
VPN configuration	*Cisco IOS Dial Services Configuration Guide*, Release 12.3 and *Cisco IOS Switching Services Configuration Guide*, Release 12.3

Related Topic	Document Title
VPN Routing and Forwarding (VRF) instances	*Cisco IOS Switching Services Configuration Guide*, Release 12.3
Globally configuring the Label Distribution Protocol (LDP) on every interface associated with a specified Interior Gateway Protocol (IGP) instance	*MPLS LDP Autoconfiguration*
Configuring PE routers in an MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network: • Using LDP to carry the labels and IGP to carry the routes between PE and CE routers. • Using the Border Gateway Protocol (BGP) to transport routes and MPLS labels between the PE routers and CE routers using multiple paths.	*MPLS VPN—Carrier Supporting Carrier* *MPLS VPN—Carrier Supporting Carrier—IPv4 BGP Label Distribution*

Standards

Standard	Title
IPv4-Tunnel SAFI	*IPv4-Tunnel SAFI* http://www.ietf.org/internet-drafts/draft-nalawade-kapoor-tunnel-safi-02.txt

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 2547	*BGP/MPLS VPNs*

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, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents modified commands.

- **address-family ipv4—BGP**
- **clear ip bgp**
- **clear tunnel l3vpn l2tpv3**
- **show ip bgp ipv4**

address-family ipv4—BGP

To enter address family configuration mode to configure an IPv4 routing session under a BGP routing process, use the **address-family ipv4** command in router configuration mode. To disable an address family specific IPv4 routing session, use the **no** form of this command.

address-family ipv4 [**multicast** | **unicast** | **vrf** *vrf-name* | **tunnel**]

no address-family ipv4 [**multicast** | **unicast** | **vrf** *vrf-name* | **tunnel**]

Syntax Description

multicast	(Optional) Specifies an IPv4 multicast address prefix routing session.
unicast	(Optional) Specifies an IPv4 unicast address prefix routing session.
vrf *vrf-name*	(Optional) Specifies the name of the virtual routing and forwarding (VRF) instance to associate with an IPv4 routing session and subsequent address family configuration mode commands.
tunnel	(Optional) Specifies a IPv4 routing session for multipoint tunnelling.

Defaults

IP Version 4 address prefixes are not enabled. Unicast address prefixes are the default when IP Version 4 address prefixes are configured.

Command Modes

Router configuration

Command History

Release	Modification
12.0(5)T	This command was introduced. The **address-family ipv4** command replaces the **match nlri** and **set nlri** commands.
12.0(28)S	The **tunnel** keyword was integrated in Cisco IOS Release 12.0(28)S.
12.0(30)S	Support for the Cisco 12000 series Internet router was added.

Usage Guidelines

The **address-family ipv4** command places the router in an address family configuration mode, from which you can configure address family and subaddress family specific routing sessions that use standard IP Version 4 address prefixes. To leave an address family configuration mode and return to router configuration mode, type **exit**.

Routing information for address family IP Version 4 is advertised by default when you configure a BGP routing session using the **neighbor remote-as** command unless you enter the **no bgp default ipv4-unicast** command.

The **tunnel** keyword is used to enable the tunnel subaddress family identifier (SAFI) under the IPv4 address family identifier. This SAFI is used to advertise the tunnel endpoints and the SAFI specific attributes (which contain the tunnel type and tunnel capabilities). Redistribution of tunnel endpoints into the BGP IPv4 tunnel SAFI table occurs automatically when the tunnel address-family is configured. However, peers need to be activated under the tunnel address-family before the sessions can exchange tunnel information.

Examples

The following example places the router in tunnel address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 tunnel
Router(config-router-af)#
```

The following example places the router in IPv4 address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4
Router(config-router-af)#
```

The following example places the router in IPv4 multicast address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 multicast
Router(config-router-af)#
```

The following example places the router in IPv4 unicast address family configuration mode:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 unicast
Router(config-router-af)#
```

The following example places the router in address family configuration mode and specifies **cisco** as the name of the VRF instance to associate with subsequent IP Version 4 address family configuration mode commands:

```
Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf cisco
Router(config-router-af)#
```

Note Use this form of the command, which specifies a VRF, only to configure routing exchanges between provider edge (PE) and customer edge (CE) devices.

Related Commands

Command	Description
address-family vpnv4	Places the router in address family configuration mode for configuring routing sessions such as BGP, RIP, or static routing sessions that use standard VPN Version 4 address prefixes.
neighbor activate	Enables the exchange of information with a BGP neighboring router.

clear ip bgp

To reset a BGP connection using BGP soft reconfiguration, use the **clear ip bgp** command in privileged EXEC mode.

clear ip bgp {***** | *neighbor-address* | **peer-group** *peer-group-name*} [**ipv4 tunnel**] [**soft** [**in** | **out**]]

Syntax Description

*****	Specifies that all current BGP sessions will be reset.
neighbor-address	Specifies that only the identified BGP neighbor will be reset.
peer-group *peer-group-name*	Specifies that only the specified BGP peer group will be reset.
ipv4 tunnel	(Optional) Specifies that only the “tunnel” SAFI IPv4 session will be reset.
soft	(Optional) Soft reset. Does not reset the session.
in | **out**	(Optional) Triggers inbound or outbound soft reconfiguration. If the **in** or **out** option is not specified, both inbound and outbound soft reset is triggered.

Defaults

No reset is initiated.

Command Modes

Privileged EXEC

Command History

Release	Modification
10.0	This command was introduced.
12.0(6)T 12.0(2)S	The dynamic inbound soft reset capability was added.

Usage Guidelines

You can reset inbound routing table updates dynamically or by generating new updates using stored update information. Using stored update information required additional memory for storing the updates.

To reset inbound routing table updates dynamically, all BGP routers must support the route refresh capability. To determine whether a BGP router supports this capability, use the **show ip bgp neighbors** command. If a router supports the route refresh capability, the following message is displayed:

```
Received route refresh capability from peer.
```

If all BGP routers support the route refresh capability, use the **clear ip bgp** {***** | *address* | *peer-group-name*} **in** command. You need not use the **soft** keyword, because soft reset is automatically assumed when the route refresh capability is supported.

To generate new inbound updates from stored update information (rather than dynamically) without resetting the BGP session, you must preconfigure the local BGP router using the **neighbor soft-reconfiguration inbound** command. This preconfiguration causes the software to store all received updates without modification regardless of whether an update is accepted by the inbound policy. Storing updates is memory intensive and should be avoided if possible.

Outbound BGP soft configuration has no memory overhead and does not require any preconfiguration. You can trigger an outbound reconfiguration on the other side of the BGP session to make the new inbound policy take effect.

Use this command whenever any of the following changes occur:

- Additions or changes to the BGP-related access lists
- Changes to BGP-related weights
- Changes to BGP-related distribution lists
- Changes to BGP-related route maps

Examples

The following example clears the inbound session with the neighbor 10.108.1.1 without resetting the session:

```
Router# clear ip bgp 10.108.1.1 soft in
```

The following example clears the outbound session with the peer group named corp without resetting the session:

```
Router# clear ip bgp peer-group corp soft out
```

Related Commands

Command	Description
neighbor soft-reconfiguration	Configures the Cisco IOS software to start storing updates.
show ip bgp	Displays entries in the BGP routing table.

clear tunnel l3vpn l2tpv3

To reset a Layer 3 VPN session over a L2TPv3 tunnel, use the **clear tunnel l3vpn l2tpv3** command in privileged EXEC mode.

clear tunnel l3vpn l2tpv3 [*hold-time*]

Syntax Description

hold-time	(optional) Configures the amount of time that the existing tunnel session will remain valid, while the new session is propagated to peers. The range of the *hold-time* argument is from 1 to 59 seconds. The default value is 15 seconds.

Defaults

No reset is initiated.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.0(28)S	This command was integrated in Cisco IOS Release 12.0(28)S.
12.0(30)S	Support for the Cisco 12000 series Internet router was added.

Usage Guidelines

This command is used to generate and distribute a new L2TPv3 session for a Layer 3 VPN. This command is issued on the PE router. The *hold-time* argument is used to configure the amount of time that the existing session will remain valid, while the new session is propagated to peers. The default value for the *hold-time* argument is 15 seconds. This should be enough time for most networks. However, this value can be increased if it takes longer for the new session to propagate to all other PE routers.

Examples

The following example resets the existing L2TPv3 session for a Layer 3 VPN and generates a new session:

```
Router# clear tunnel l3vpn l2tpv3
```

show ip bgp ipv4

To display entries in the IP version 4 (IPv4) Border Gateway Protocol (BGP) routing table, use the **show ip bgp ipv4** command in EXEC mode.

show ip bgp ipv4 {**multicast** | **unicast** | **tunnel** [*ip-address* | **summary**]}

Syntax Description

multicast	Displays entries for multicast routes.
unicast	Displays entries for unicast routes.
tunnel	Displays entries configured under the "tunnel" SAFI.
ip-address	Displays tunnel specific information for the specified IP address.
summary	Displays a summary of all routing information configured under the "tunnel" SAFI.

Command Modes

EXEC

Command History

Release	Modification
12.0(5)T	This command was introduced.

Examples

The following is sample output from the **show ip bgp ipv4 unicast** command:

```
Router# show ip bgp ipv4 unicast

 BGP table version is 4, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24   172.16.10.1              0             0  300 i
*> 10.10.20.0/24   172.16.10.1              0             0  300 i
*  10.20.10.0/24   172.16.10.1              0             0  300 i
```

The following is sample output from the **show ip bgp ipv4 multicast** command:

```
Router# show ip bgp ipv4 multicast

 BGP table version is 4, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24   172.16.10.1              0             0  300 i
*> 10.10.20.0/24   172.16.10.1              0             0  300 i
*  10.20.10.0/24   172.16.10.1              0             0  300 i
```

The following is sample output from the **show ip bgp ipv4 tunnel** command:

```
Router# show ip bgp ipv4 tunnel

BGP table version is 3, local router ID is 10.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            MetricLocPrf Weight  Path
*> 10.3.3.3/32       0.0.0.0             0           32768   ?
*>i10.5.5.5/32       10.5.5.5            0     100    0       ?
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 402,cookie_high D0338947,cookie_low 69DCF79E
ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 401,cookie_high 6FB30F92,cookie_low A7E61105
```

The following is sample output from the **show ip bgp ipv4 summary** command:

```
Router# show ip bgp ipv4 tunnel summary

BGP router identifier 10.3.3.3, local AS number 1
BGP table version is 3, main routing table version 3
..
2 BGP SAFI-Specific-Attr entries using 80 bytes of memory
..
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.5.5.5        4     1     422     413        3    0    0 05:28:23        1
```

The following is sample output from the **show ip bgp tunnel** command when a single IP address is specified:

```
Router# show ip bgp ip tunnel 10.5.5.5

BGP routing table entry for 10.5.5.5/32, version 2
Paths: (1 available, best #1, table IPv4-Tunnel-BGP-Table)
  Not advertised to any peer
  Local
    10.5.5.5 (metric 30) from 10.5.5.5 (10.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      SAFI Specific Attribute: ssacount=1, type L2TP, len 16
        pref 0,flags 0,cookielen 8,ss_id 402,cookie_high D0338947,cookie_low 69DCF79E
```

Table 3 describes the significant fields shown in the display.

Table 3 show ip bgp ipv4 unicast Field Descriptions

Field	Description
BGP table version	Internal version number of the table. This number is incremented whenever the table changes.
local router ID	IP address of the router.
Status codes	Status of the table entry. The status is displayed at the beginning of each line in the table. It can be one of the following values: s—Table entry is suppressed. d—Table entry is damped. h—Table entry history. *—Table entry is valid. >—Table entry is the best entry to use for that network. i—Table entry was learned through an internal BGP (iBGP) session.

Table 3 show ip bgp ipv4 unicast Field Descriptions (continued)

Field	Description
Origin codes	Origin of the entry. The origin code is displayed at the end of each line in the table. It can be one of the following values: i—Entry originated from an Interior Gateway Protocol (IGP) and was advertised with a **network** router configuration command. e—Entry originated from an Exterior Gateway Protocol (EGP). ?—Origin of the path is not clear. Usually, this is a router that is redistributed into BGP from an IGP.
Network	IP address of a network entity.
Next Hop	IP address of the next system that is used when forwarding a packet to the destination network. An entry of 0.0.0.0 indicates that the router has some non-BGP routes to this network.
Metric	If shown, the value of the interautonomous system metric.
LocPrf	Local preference value as set with the **set local-preference** route-map configuration command. The default value is 100.
Weight	Weight of the route as set using autonomous system filters.
Path	Autonomous system paths to the destination network. There can be one entry in this field for each autonomous system in the path.

Related Commands

Command	Description
show ip bgp	Displays entries in the BGP routing table.