MPLS VPN Carrier Supporting Carrier with BGP

Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) Carrier Supporting Carrier (CSC) enables one MPLS VPN-based service provider to allow other service providers to use a segment of its backbone network. This module explains how to configure an MPLS VPN CSC network that uses Border Gateway Protocol (BGP) to distribute routes and MPLS labels.

Finding Feature Information

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

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

Prerequisites for MPLS VPN CSC with BGP

- You should be able to configure MPLS VPNs with end-to-end (CE-to-CE router) pings working. To accomplish this, you need to know how to configure Interior Gateway Protocols (IGPs), MPLS Label Distribution Protocol (LDP), and Multiprotocol Border Gateway Protocol (MP-BGP).

- Make sure that the CSC-PE routers and the CSC-CE routers run images that support BGP label distribution. Otherwise, you cannot run external BGP (EBGP) between them. Ensure that connectivity between the customer carrier and the backbone carrier. EBGP-based label distribution is configured on these links to enable MPLS between the customer and backbone carriers.

Restrictions for MPLS VPN CSC with BGP

On a provider edge (PE) router, you can configure an interface for either BGP with labels or LDP. You cannot enable both types of label distribution on the same interface. If you switch from one protocol to the other, then you must disable the existing protocol on all interfaces before enabling the other protocol.

This feature does not support the following:

- EBGP multihop between CSC-PE and CSC-CE routers
- EIBGP multipath load sharing

The physical interfaces that connect the BGP speakers must support Cisco Express Forwarding or distributed Cisco Express Forwarding and MPLS.

Information About MPLS VPN CSC with BGP

MPLS VPN CSC Introduction

Carrier supporting carrier is where one service provider allows another service provider to use a segment of its backbone network. The service provider that provides the segment of the backbone network to the other provider is called the backbone carrier. The service provider that uses the segment of the backbone network is called the customer carrier.

A backbone carrier offers Border Gateway Protocol and Multiprotocol Label Switching (BGP/MPLS) VPN services. The customer carrier can be either:

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

Benefits of Implementing MPLS VPN CSC

The MPLS VPN CSC network provides the following benefits to service providers who are backbone carriers and to customer carriers.

Benefits to the Backbone Carrier

- The backbone carrier can accommodate many customer carriers and give them access to its backbone. The backbone carrier does not need to create and maintain separate backbones for its customer carriers. Using one backbone network to support multiple customer carriers simplifies the backbone carrier's VPN operations. The backbone carrier uses a consistent method for managing and maintaining the backbone network. This is also cheaper and more efficient than maintaining separate backbones.

- The MPLS VPN carrier supporting carrier feature is scalable. Carrier supporting carrier can change the VPN to meet changing bandwidth and connectivity needs. The feature can accommodate unplanned growth and changes. The carrier supporting carrier feature enables tens of thousands of VPNs to be set up over the same network, and it allows a service provider to offer both VPN and Internet services.
- The MPLS VPN carrier supporting carrier feature is a flexible solution. The backbone carrier can accommodate many types of customer carriers. The backbone carrier can accept customer carriers who are ISPs or VPN service providers or both. The backbone carrier can accommodate customer carriers that require security and various bandwidths.

Benefits to the Customer Carriers

- The MPLS VPN carrier supporting carrier feature removes from the customer carrier the burden of configuring, operating, and maintaining its own backbone. The customer carrier uses the backbone network of a backbone carrier, but the backbone carrier is responsible for network maintenance and operation.
- Customer carriers who use the VPN services provided by the backbone carrier receive the same level of security that Frame Relay or ATM-based VPNs provide. Customer carriers can also use IPSec in their VPNs for a higher level of security; it is completely transparent to the backbone carrier.
- Customer carriers can use any link layer technology (SONET, DSL, Frame Relay, and so on) to connect the CE routers to the PE routers and the PE routers to the P routers. The MPLS VPN carrier supporting carrier feature is link layer independent. The CE routers and PE routers use IP to communicate, and the backbone carrier uses MPLS.
- The customer carrier can use any addressing scheme and still be supported by a backbone carrier. The customer address space and routing information are independent of the address space and routing information of other customer carriers or the backbone provider.

Benefits of Implementing MPLS VPN CSC with BGP

You can configure your CSC network to enable BGP to transport routes and MPLS labels between the backbone carrier PE routers and the customer carrier CE routers using multiple paths. The benefits of using BGP to distribute IPv4 routes and MPLS label routes are:

- BGP takes the place of an IGP and LDP in a VPN forwarding/routing instance (VRF) table. You can use BGP to distribute routes and MPLS labels. Using a single protocol instead of two simplifies the configuration and troubleshooting.
- BGP is the preferred routing protocol for connecting two ISPs, mainly because of its routing policies and ability to scale. ISPs commonly use BGP between two providers. This feature enables those ISPs to use BGP.

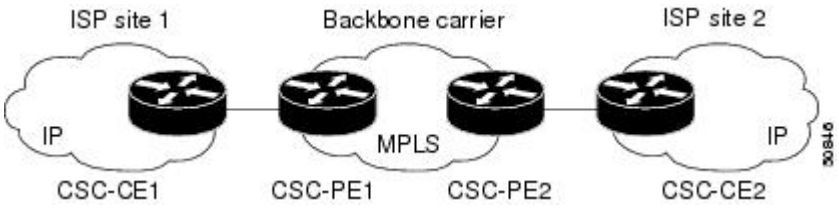
Configuration Options for MPLS VPN CSC with BGP

The following sections explain how the backbone and customer carriers distribute IPv4 routes and MPLS labels. The backbone carrier offers BGP and MPLS VPN services. The customer carrier can be either of the following:

Customer Carrier Is an ISP with an IP Core

The figure below shows a network configuration where the customer carrier is an ISP. The customer carrier has two sites, each of which is a point of presence (POP). The customer carrier connects these sites using a VPN service provided by the backbone carrier. The backbone carrier uses MPLS. The ISP sites use IP.

Figure 1: Network Where the Customer Carrier Is an ISP



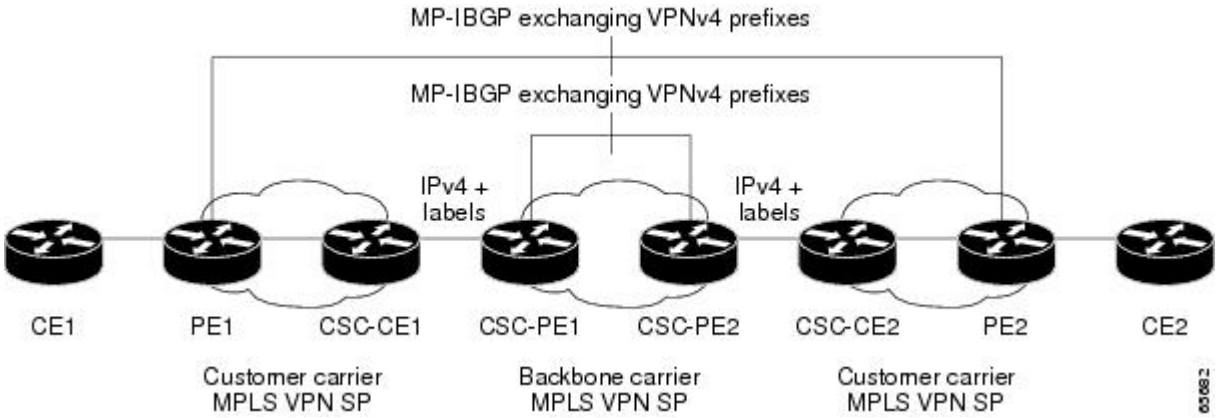
The links between the CE and PE routers use EBGP to distribute IPv4 routes and MPLS labels. Between the links, the PE routers use multiprotocol IBGP to distribute VPNv4 routes.

Note If a router other than a Cisco router is used as a CSC-PE or CSC-CE, that router must support IPv4 BGP label distribution (RFC 3107). Otherwise, you cannot run EBGP with labels between the routers.

Customer Carrier Is an MPLS Service Provider With or Without VPN Services

The figure below shows a network configuration where the backbone carrier and the customer carrier are BGP/MPLS VPN service providers. This is known as hierarchical VPNs. The customer carrier has two sites. Both the backbone carrier and the customer carrier use MPLS in their networks.

Figure 2: Network Where the Customer Carrier Is an MPLS VPN Service Provider



In this configuration, the customer carrier can configure its network in one of the following ways:

- The customer carrier can run IGP and LDP in its core network. In this case, the CSC-CE1 router in the customer carrier redistributes the EBGP routes it learns from the CSC-PE1 router of the backbone carrier to IGP.
- The CSC-CE1 router of the customer carrier system can run an IPv4 and labels IBGP session with the PE1 router.

How to Configure MPLS VPN CSC with BGP

Identifying the Carrier Supporting Carrier Topology

Before you configure the MPLS VPN CSC with BGP, you need to identify both the backbone and customer carrier topology.

For hierarchical VPNs, the customer carrier of the MPLS VPN network provides MPLS VPN services to its own customers. In this instance, you need to identify the type of customer carrier as well as the topology of the customer carriers. Hierarchical VPNs require extra configuration steps, which are noted in the configuration sections.

Note You can connect multiple CSC-CE routers to the same PE, or you can connect a single CSC-CE router to CSC-PEs using more than one interface to provide redundancy and multiple path support in CSC topology.

Perform this task to identify the carrier supporting carrier topology.

SUMMARY STEPS

1. Identify the type of customer carrier, ISP or MPLS VPN service provider.
2. (For hierarchical VPNs only) Identify the CE routers.
3. (For hierarchical VPNs only) Identify the customer carrier core router configuration.
4. Identify the customer carrier edge (CSC-CE) routers.
5. Identify the backbone carrier router configuration.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Identify the type of customer carrier, ISP or MPLS VPN service provider.	Sets up requirements for configuration of carrier supporting carrier network. • For an ISP, customer site configuration is not required. • For an MPLS VPN service provider, the customer site needs to be configured, as well as any task or step designated “for hierarchical VPNs only.”

	Command or Action	Purpose
Step 2	(For hierarchical VPNs only) Identify the CE routers.	Sets up requirements for configuration of CE to PE connections.
Step 3	(For hierarchical VPNs only) Identify the customer carrier core router configuration.	Sets up requirements for connection configuration between core (P) routers and between P routers and edge routers (PE and CSC-CE routers).
Step 4	Identify the customer carrier edge (CSC-CE) routers.	Sets up requirements for configuration of CSC-CE to CSC-PE connections.
Step 5	Identify the backbone carrier router configuration.	Sets up requirements for connection configuration between CSC core routers and between CSC core routers and edge routers (CSC-CE and CSC-PE routers).

What to Do Next

Set up your carrier supporting carrier networks with the Configuring the Backbone Carrier Core, on page 6.

Configuring the Backbone Carrier Core

Configuring the backbone carrier core requires setting up connectivity and routing functions for the CSC core and the CSC-PE routers.

Configuring and verifying the CSC core (backbone carrier) involves the following tasks:

Prerequisites

Before you configure a backbone carrier core, configure the following on the CSC core routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on.
- Label Distribution Protocol (LDP). For information, see How to Configure MPLS LDP.

Verifying IP Connectivity and LDP Configuration in the CSC Core

Perform this task to verify IP connectivity and LDP configuration in the CSC core.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
5. **show mpls ldp discovery** [**vrf** *vrf-name* | **all**]
6. **show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**]
7. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
8. **show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**]
9. **show ip route**
10. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name*	*system-address*} **Example:** `Router# ping ip 10.1.0.0`	(Optional) Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. • Use the **ping ip** command to verify the connectivity from one CSC core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.2.0.0`	(Optional) Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.
Step 4	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask*	*length*}	**labels** *label* [- *label*]	**interface** *interface*	**next-hop** *address*	**lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table`	(Optional) Displays the contents of the MPLS label forwarding information base (LFIB). • Use the **show mpls forwarding-table** command to verify that MPLS packets are being forwarded.
Step 5	**show mpls ldp discovery** [**vrf** *vrf-name*	**all**]	(Optional) Displays the status of the LDP discovery process.

	Command or Action	Purpose
	Example: `Router# show mpls ldp discovery`	• Use the **show mpls ldp discovery** command to verify that LDP is operational in the CSC core.
Step 6	**show mpls ldp neighbor** [[**vrf** *vrf-name*] [*address* | *interface*] [**detail**] | **all**] **Example:** `Router# show mpls ldp neighbor`	(Optional) Displays the status of LDP sessions. • Use the **show mpls ldp neighbor** command to verify LDP configuration in the CSC core.
Step 7	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef`	(Optional) Displays entries in the forwarding information base (FIB). • Use the **show ip cef** command to check the forwarding table (prefixes, next hops, and interfaces).
Step 8	**show mpls interfaces** [[**vrf** *vrf-name*] [*interface*] [**detail**] | **all**] **Example:** `Router# show mpls interfaces`	(Optional) Displays information about one or more or all interfaces that are configured for label switching. • Use the **show mpls interfaces** command to verify that the interfaces are configured to use LDP.
Step 9	**show ip route** **Example:** `Router# show ip route`	(Optional) Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, interface, and so forth.
Step 10	**disable** **Example:** `Router# disable`	(Optional) Returns to privileged EXEC mode.

Troubleshooting Tips

You can use the **ping** and **trace** commands to verify complete MPLS connectivity in the core. You also get useful troubleshooting information from the additional **show** commands.

Configuring VRFs for CSC-PE Routers

Perform this task to configure VPN forwarding/routing instances (VRFs) for the backbone carrier edge (CSC-PE) routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **exit**
8. **interface** *type number*
9. **ip vrf forwarding** *vrf-name*
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 vpn1`	Defines the VPN routing instance by assigning a VRF name and enters VRF configuration mode. • The *vrf-name* argument is the name assigned to a VRF.

Step 4	**rd** *route-distinguisher*
	
	Example:
	`Router(config-vrf)# rd 100:1`
		Creates routing and forwarding tables.
		• The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix. You can enter an RD in either of these formats:
		  • 16-bit AS number: your 32-bit number, for example, 101:3
		  • 32-bit IP address: your 16-bit number, for example, 192.168.122.15:1

Step 5	**route-target** {**import**	**export**	**both**} *route-target-ext-community*	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community.

	Command or Action	Purpose
	Example: `Router(config-vrf)# route-target import 100:1`	• The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and exports routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map vpn1-route-map`	(Optional) Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**exit** **Example:** `Router(config-vrf)# exit`	(Optional) Exits to global configuration mode.
Step 8	**interface** *type number* **Example:** `Router(config)# interface Ethernet5/0`	Specifies the interface to configure. • The *type* argument specifies the type of interface to be configured. • The *number* argument specifies the port, connector, or interface card number.
Step 9	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding vpn1`	Associates a VRF with the specified interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.
Step 10	**end** **Example:** `Router(config-if)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip vrf detail** command and make sure the MPLS VPN is up and associated with the right interfaces.

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier

Perform this task to configure Multiprotocol BGP (MP-BGP) connectivity in the backbone carrier.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
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*} **send-community extended**
9. **neighbor** {*ip-address* | *peer-group-name*} **activate**
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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**no bgp default ipv4-unicast** **Example:** `Router(config-router)# no bgp default ipv4-unicast`	(Optional) Disables the IPv4 unicast address family on all neighbors. • Use the **no bgp default-unicast** command if you are using this neighbor for MPLS routes only.

	Command or Action	Purpose
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.5.5.5 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type* **Example:** `Router(config-router)# neighbor 10.2.0.0 update-source loopback0`	Allows BGP sessions to use a specific operational interface for TCP connections. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *interface-type* argument specifies the interface to be used as the source.
Step 7	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community extended** **Example:** `Router(config-router-af)# neighbor 10.0.0.1 send-community extended`	Specifies that a communities attribute should be sent to a BGP neighbor. • The *ip-address* argument specifies the IP address of the BGP-speaking neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.4.0.0 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 10	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

You can enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. If this command is not successful, enter a **debug ip bgp x.x.x.x events** command, where *x.x.x.x* is the IP address of the neighbor.

Configuring the CSC-PE and CSC-CE Routers

Perform the following tasks to configure and verify links between a CSC-PE router and the carrier CSC-CE router for an MPLS VPN CSC network that uses BGP to distribute routes and MPLS labels.

The figure below shows the configuration for the peering with directly connected interfaces between CSC-PE and CSC-CE routers. This configuration is used as the example in the tasks that follow.

Figure 3: Configuration for Peering with Directly Connected Interfaces Between CSC-PE and CSC-CE Routers

Configuring CSC-PE Routers

Perform this task to configure the CSC-PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** *ip-address* **as-override**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 vrf vpn1`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.0.0.1 remote-as 200`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** *ip-address* **as-override** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 as-override`	Configures a PE router to override the autonomous system number (ASN) of a site with the ASN of a provider. • The *ip-address* argument specifies the IP address of the router that is to be overridden with the ASN provided.

	Command or Action	Purpose
Step 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

Enter a **show ip bgp neighbor** command to verify that the neighbors are up and running. Make sure you see the following line in the command output under Neighbor capabilities:

```
IPv4 MPLS Label capability:advertised and received
```

Configuring CSC-CE Routers

Perform this task to configure the CSC-CE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **redistribute** *protocol*
6. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *peer-group-name*} **activate**
8. **neighbor** *ip-address* **send-label**
9. **exit-address-family**
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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.

Step 5	**redistribute** *protocol*
	
	Example:
	
	`Router(config-router-af)# redistribute static`
		Redistributes routes from one routing domain into another routing domain.
		
		• The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **egp**, **igrp**, **isis**, **ospf**, **mobile**, **static** [**ip**], **connected**, and **rip**.
		
		  • The **static** [**ip**] keyword redistributes IP static routes. The optional **ip** keyword is used when you redistribute static routes into IS-IS.
		
		  • The **connected** keyword refers to routes which are established automatically when IP is enabled on an interface. For routing protocols such as OSPF and IS-IS, these routes are redistributed as external to the autonomous system.

Step 6	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor.

	Command or Action	Purpose
	Example: `Router(config-router-af)# neighbor 10.5.0.2 remote-as 100`	• The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 7	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.3.0.2 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 8	**neighbor** *ip-address* **send-label** **Example:** `Router(config-router-af)# neighbor 10.0.0.2 send-label`	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighboring router.
Step 9	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits from the address family configuration mode.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Verifying Labels in the CSC-PE Routers

Perform this task to verify the labels in the CSC-PE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
3. **show mpls interfaces** [**all**]
4. **show ip route vrf** *vrf-name* [*prefix*]
5. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**]
6. **show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**]
7. **show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
8. **traceroute vrf** [*vrf-name*] *ip-address*
9. **disable**

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*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 all summary`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4 all summary** command to check that the BGP session is up and running between the CSC-PE routers and the CSC-CE routers. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.
Step 3	**show mpls interfaces** [**all**] **Example:** `Router# show mpls interfaces all`	(Optional) Displays information about one or more interfaces that have been configured for label switching. • Use the **show mpls interfaces all** command to check that MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. Check that LDP is turned off on the VRF because EBGP distributes the labels.
Step 4	**show ip route vrf** *vrf-name* [*prefix*] **Example:** `Router# show ip route vrf vpn1 10.5.5.5`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to check that the prefixes for the PE routers are in the routing table of the CSC-PE routers. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.

	Command or Action	Purpose
Step 5	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher* | **vrf** *vrf-name*} [**summary**] [**labels**] **Example:** `Router# show ip bgp vpnv4 vrf vpn1 labels`	(Optional) Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4 vrf** *vrf-name* **labels** command to check that the prefixes for the customer carrier MPLS service provider networks are in the BGP table and have the appropriate labels. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 6	**show ip cef** [**vrf** *vrf-name*] [*network* [*mask*]] [**longer-prefixes**] [**detail**] **Example:** `Router# show ip cef vrf vpn1 10.1.0.0 detail`	(Optional) Displays entries in the forwarding information base (FIB) or displays a summary of the FIB. • Use the **show ip cef vrf** and the **show ip cef vrf detail** commands to check that the prefixes of the PE routers are in the CEF table.
Step 7	**show mpls forwarding-table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn1 10.1.0.0 detail`	(Optional) Displays the contents of the MPLS lable forwarding information base (LFIB). • Use the **show mpls forwarding-table** command with the **vrf** keyword and both the **vrf** and **detail** keywords to check that the prefixes for the PE routers in the local customer MPLS VPN service provider are in the LFIB. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 8	**traceroute vrf** [*vrf-name*] *ip-address* **Example:** `Router# traceroute vrf vpn2 10.2.0.0`	Shows the routes that packets follow traveling through a network to their destination. • Use the **traceroute vrf** command to check the data path and transport labels from a PE to a destination CE router. **Note** This command works with MPLS-aware traceroute only if the backbone routers are configured to propagate and generate IP Time to Live (TTL) information. For more information, see the documentation on the **mpls ip propagate-ttl** command. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 9	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Verifying Labels in the CSC-CE Routers

Perform this task to verify the labels in the CSC-CE routers.

SUMMARY STEPS

1. **enable**
2. **show ip bgp summary**
3. **show ip route** [*address*]
4. **show mpls ldp bindings** [*network* {*mask* | *length*}]
5. **show ip cef** [*network* [*mask*]] [**longer-prefixes**] [**detail**]
6. **show mpls forwarding table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**]
7. **show ip bgp labels**

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 summary** **Example:** `Router# show ip bgp summary`	(Optional) Displays the status of all BGP connections. • Use the **show ip bgp summary** command to check that the BGP session is up and running on the CSC-CE routers.
Step 3	**show ip route** [*address*] **Example:** `Router# show ip route 10.1.0.0`	(Optional) Displays IP routing table entries. • Use the **show ip route** to check that the loopback address of the local and remote PE routers are in the routing table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 4	**show mpls ldp bindings** [*network* {*mask* | *length*}] **Example:** `Router# show mpls ldp bindings 10.2.0.0 255.255.255.255`	(Optional) Displays the contents of the label information base (LIB). • Use the **show mpls ldp bindings** command to check that the prefix of the local PE router is in the MPLS LDP bindings.
Step 5	**show ip cef** [*network* [*mask*]] [**longer-prefixes**] [**detail**]	(Optional) Displays entries in the forwarding information base (FIB) or a summary of the FIB.

	Command or Action	Purpose
	Example: `Router# show ip cef 10.5.0.0 detail`	• Use the **show ip cef** and the **show ip cef detail** commands to check that the prefixes of the local and remote PE routers are in the Cisco Express Forwarding table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and the labels for the same destination learned from the CSC-CE are installed in the corresponding VRF table.
Step 6	**show mpls forwarding table** [**vrf** *vrf-name*] [{*network* {*mask* | *length*} | **labels** *label* [- *label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*]}] [**detail**] **Example:** `Router# show mpls forwarding-table 10.2.0.0 detail`	(Optional) Displays the contents of the MPLS LFIB. • Use the **show mpls forwarding-table** and **show mpls forwarding-table detail** commands to check that the prefixes of the local and remote PE routers are in the MPLS forwarding table. **Note** If you have multiple paths configured between CSC-PE and CSC-CE, verify that the multiple routes and labels for the same destination learned from the CSC-CE are installed in the corresponding VRF routing table.
Step 7	**show ip bgp labels** **Example:** `Router# show ip bgp labels`	(Optional) Displays information about MPLS labels from the EBGP route table. • Use the **show ip bgp labels** command to check that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks.

Configuring the Customer Carrier Network

Perform the following tasks to configure and verify the customer carrier network. This requires setting up connectivity and routing functions for the customer carrier core (P) routers and the customer carrier edge (PE) routers.

Prerequisites

Before you configure an MPLS VPN CSC network that uses BGP to distribute routes and MPLS labels, you must configure the following on your customer carrier routers:

- An IGP routing protocol--BGP, OSPF, IS-IS, EIGRP, static, and so on. For information, see Configuring a Basic BGP Network, Configuring OSPF, Configuring a Basic IS-IS Network, and Configuring EIGRP.
- MPLS VPN functionality on the PE routers (for hierarchical VPNs only).
- Label Distribution Protocol (LDP) on P and PE routers (for hierarchical VPNs only). For information, see How to Configure MPLS LDP.

Note You must configure the items in the preceding list before performing the tasks in this section.

Verifying IP Connectivity in the Customer Carrier

Perform this task to verify IP connectivity in the customer carrier.

SUMMARY STEPS

1. **enable**
2. **ping** [*protocol*] {*host-name* | *system-address*}
3. **trace** [*protocol*] [*destination*]
4. **show ip route**
5. **disable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ping** [*protocol*] {*host-name* | *system-address*} **Example:** `Router# ping ip 10.2.0.0`	Diagnoses basic network connectivity on AppleTalk, CLNS, IP, Novell, Apollo, VINES, DECnet, or XNS networks. • Use the **ping** command to verify the connectivity from one customer carrier core router to another.
Step 3	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.1.0.0`	Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to verify the path that a packet goes through before reaching the final destination. The **trace** command can help isolate a trouble spot if two routers cannot communicate.
Step 4	**show ip route** **Example:** `Router# show ip route`	Displays IP routing table entries. • Use the **show ip route** command to display the entire routing table, including host IP address, next hop, interface, and so forth.

	Command or Action	Purpose
Step 5	**disable** **Example:** `Router# disable`	Returns to user mode.

Configuring a Customer Carrier Core Router as a Route Reflector

Perform this task to configure a customer carrier core (P) router as a route reflector of multiprotocol BGP prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family vpnv4** [**unicast**]
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **neighbor** *ip-address* **route-reflector-client**
8. **exit-address-family**
9. **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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and labels

	Command or Action	Purpose
		the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 4	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.1.1.1 remote-as 100`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 5	**address-family vpnv4** [**unicast**] **Example:** `Router(config-router)# address-family vpnv4`	Enters address family configuration mode for configuring routing sessions, such as BGP, that use standard VPNv4 address prefixes. • The optional **unicast** keyword specifies VPNv4 unicast address prefixes.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.1.1.1 activate`	Enables the exchange of information with a neighboring BGP router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**neighbor** *ip-address* **route-reflector-client** **Example:** `Router(config-router-af)# neighbor 10.1.1.1 route-reflector-client`	Configures the router as a BGP route reflector and configures the specified neighbor as its client. • The *ip-address* argument specifies the IP address of the BGP neighbor being identified as a client.
Step 8	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 9	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Troubleshooting Tips

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only unicast address prefixes. For neighbors to exchange other address prefix types, such as multicast and VPNv4, you must also activate neighbors using the **neighbor activate** command in address family configuration mode, as shown.

Route reflectors and clients (neighbors or internal BGP peer groups) that are defined in router configuration mode using the **neighbor route-reflector-client** command reflect unicast address prefixes to and from those clients by default. To cause them to reflect prefixes for other address families, such as multicast, define the reflectors and clients in address family configuration mode, using the **neighbor route-reflector-client** command, as shown.

Configuring the Customer Site for Hierarchical VPNs

Note This section applies only to customer carrier networks that use BGP to distribute routes and MPLS labels.

Perform the following tasks to configure and verify the customer site for hierarchical VPNs:

Note This section applies to hierarchical VPNs only.

Defining VPNs on PE Routers for Hierarchical VPNs

Perform this task to define VPNs on PE routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
6. **import map** *route-map*
7. **ip vrf forwarding** *vrf-name*
8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• 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 vpn2`	Creates a VRF routing table and a Cisco Express Forwarding table and enters VRF configuration mode. • The *vrf-name* argument is a name you assign to a VRF.
Step 4	**rd** *route-distinguisher* **Example:** `Router(config-vrf)# rd 200:1`	Creates routing and forwarding tables for a VRF. • The *route-distinguisher* argument adds an 8-byte value to an IPv4 prefix to create a VPN IPv4 prefix.
Step 5	**route-target** {**import** | **export** | **both**} *route-target-ext-community* **Example:** `Router(config-vrf)# route-target export 200:1`	Creates a route-target extended community for a VRF. • The **import** keyword imports routing information from the target VPN extended community. • The **export** keyword exports routing information to the target VPN extended community. • The **both** keyword imports routing information from and export routing information to the target VPN extended community. • The *route-target-ext-community* argument adds the route-target extended community attributes to the VRF's list of import, export, or both (import and export) route-target extended communities.
Step 6	**import map** *route-map* **Example:** `Router(config-vrf)# import map map23`	Configures an import route map for a VRF. • The *route-map* argument specifies the route map to be used as an import route map for the VRF.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-vrf)# ip vrf forwarding vpn2`	Associates a VPN VRF instance with an interface or subinterface. • The *vrf-name* argument is the name assigned to a VRF.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits to global configuration mode.

Configuring BGP Routing Sessions on the PE Routers for Hierarchical VPNs

Perform this task to configure BGP routing sessions on the PE routers for PE-to-CE router communication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name*} **activate**
7. **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	**router bgp** *as-number* **Example:** `Router(config)# router bgp 200`	Configures the router to run a BGP process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.

	Command or Action	Purpose
Step 4	**address-family ipv4** [**multicast** | **unicast** | **vrf** *vrf-name*] **Example:** `Router(config-router)# address-family ipv4 multicast`	Specifies the IPv4 address family type and enters address family configuration mode. • The **multicast** keyword specifies IPv4 multicast address prefixes. • The **unicast** keyword specifies IPv4 unicast address prefixes. • The **vrf** *vrf-name* keyword and argument specify the name of the VRF to associate with subsequent IPv4 address family configuration mode commands.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router-af)# neighbor 10.5.5.5 remote-as 300`	Adds an entry to the BGP or multiprotocol BGP neighbor table. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 6	**neighbor** {*ip-address* | *peer-group-name*} **activate** **Example:** `Router(config-router-af)# neighbor 10.1.0.0 activate`	Enables the exchange of information with a neighboring router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group.
Step 7	**end** **Example:** `Router(config-router-af)# end`	(Optional) Exits to privileged EXEC mode.

Verifying Labels in Each PE Router for Hierarchical VPNs

Perform this task to verify labels in each PE router for hierarchical VPNs.

SUMMARY STEPS

1. **enable**
2. **show ip route vrf** *vrf-name* [*prefix*]
3. **show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**]
4. **show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**]
5. **show ip cef vrf** *vrf-name* [*ip-prefix*]
6. **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 ip route vrf** *vrf-name* [*prefix*] **Example:** `Router# show ip route vrf vpn2 10.5.5.5`	(Optional) Displays the IP routing table associated with a VRF. • Use the **show ip route vrf** command to check that the loopback addresses of the local and remote CE routers are in the routing table of the PE routers.
Step 3	**show mpls forwarding-table** [**vrf** *vrf-name*] [*prefix*] [**detail**] **Example:** `Router# show mpls forwarding-table vrf vpn2 10.1.0.0`	(Optional) Displays the contents of the LFIB. • Use the **show mpls forwarding-table** command to check that the prefixes for the local and remote CE routers are in the MPLS forwarding table, and that the specified prefix is untagged.
Step 4	**show ip cef** [*network* [*mask* [**longer-prefix**]]] [**detail**] **Example:** `Router# show ip cef 10.2.0.0`	(Optional) Displays specific entries in the FIB based on IP address information. • Use the **show ip cef** command to check that the prefixes of the local and remote PE routers are in the Cisco Express Forwarding table.
Step 5	**show ip cef vrf** *vrf-name* [*ip-prefix*] **Example:** `Router# show ip cef vrf vpn2 10.3.0.0`	(Optional) Displays the Cisco Express Forwarding table associated with a VRF. • Use the **show ip cef vrf** command to check that the prefix of the remote CE router is in the Cisco Express Forwarding table.
Step 6	**exit** **Example:** `Router# exit`	(Optional) Exits to user EXEC mode.

Configuring CE Routers for Hierarchical VPNs

Perform this task to configure CE routers for hierarchical VPNs. This configuration is the same as that for an MPLS VPN that is not in a hierarchical topology.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip cef** [**distributed**]
4. **interface** *type number*
5. **ip addres** *ip-address mask* [**secondary**]
6. **exit**
7. **router bgp** *as-number*
8. **redistribute** *protocol*
9. **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-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 cef** [**distributed**] **Example:** `Router(config)# ip cef distributed`	Enables Cisco Express Forwarding on the route processor card. • The **distributed** keyword enables distributed Cisco Express Forwarding operation. Cisco Express Forwarding information is distributed to the line cards. Line cards perform express forwarding. **Note** For the Cisco ASR 1000 Series Aggregation Services Router, the **distributed** keyword is required.
Step 4	**interface** *type number* **Example:** `Router(config)# interface loopback 0`	Configures an interface type and enters interface configuration mode. • The *type* argument specifies the type of interface to be configured. • A loopback interface indicates a software-only interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. • The *number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.

	Command or Action	Purpose
Step 5	**ip addres** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.8.0.0 255.255.255.255`	Sets a primary or secondary IP address for an interface. • The *ip-address* argument is the IP address. • The *mask* argument is the mask for the associated IP subnet. • The **secondary** keyword specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**router bgp** *as-number* **Example:** `Router(config)# router bgp 100`	Configures a BGP routing process and enters router configuration mode. • The *as-number* argument indicates the number of an autonomous system that identifies the router to other BGP routers and tags the routing information passed along. Valid numbers are from 0 to 65535. Private autonomous system numbers that can be used in internal networks range from 64512 to 65535.
Step 8	**redistribute** *protocol* **Example:** `Router(config-router)# redistribute connected`	Redistributes routes from one routing domain into another routing domain. • The *protocol* argument specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **bgp**, **connected**, **egp**, **igrp**, **isis**, **mobile**, **ospf**, **static** [**ip**], or **rip**. The **connected** keyword refers to routes that are established automatically when IP is enabled on an interface. For routing protocols such as Open Shortest Path First (OSPF) and IS-IS, these routes are redistributed as external to the autonomous system.
Step 9	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *as-number* **Example:** `Router(config-router)# neighbor 10.8.0.0 remote-as 100`	Adds the IP address of the neighbor in the remote autonomous system to the multiprotocol BGP neighbor table of the local router. • The *ip-address* argument specifies the IP address of the neighbor. • The *peer-group-name* argument specifies the name of a BGP peer group. • The *as-number* argument specifies the autonomous system to which the neighbor belongs.
Step 10	**end** **Example:** `Router(config-router)# end`	(Optional) Exits to privileged EXEC mode.

Verifying IP Connectivity in the Customer Site

Perform this task to verify IP connectivity in the customer site.

SUMMARY STEPS

1. **enable**
2. **show ip route** [*ip-address* [*mask*]] [**longer-prefixes**] | *protocol* [*process-id*] | **list** [*access-list-number* | *access-list-name*] | **static download**
3. **ping** [*protocol*] {*host-name* | *system-address*}
4. **trace** [*protocol*] [*destination*]
5. **disable**

DETAILED STEPS

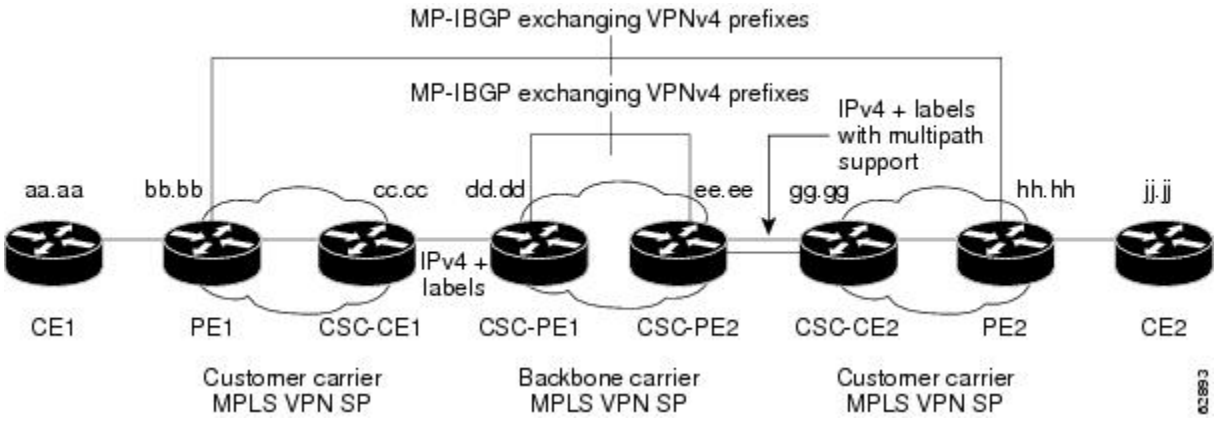
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route** [*ip-address* [*mask*]] [**longer-prefixes**]	*protocol* [*process-id*]	**list** [*access-list-number*	*access-list-name*]	**static download** **Example:** `Router# show ip route 10.5.5.5`	(Optional) Displays the current state of the routing table. • Use the **show ip route ip-address** command to check that the loopback addresses of the remote CE routers learned through the PE router are in the routing table of the local CE routers.
Step 3	**ping** [*protocol*] {*host-name*	*system-address*} **Example:** `Router# ping 10.5.5.5`	Diagnoses basic network connectivity on Apollo, AppleTalk, Connectionless Network Service (CLNS), DECnet, IP, Novell IPX, VINES, or XNS networks. • Use the **ping** command to check connectivity between customer site routers.
Step 4	**trace** [*protocol*] [*destination*] **Example:** `Router# trace ip 10.5.5.5`	Discovers the routes that packets will actually take when traveling to their destination. • Use the **trace** command to follow the path of the packets in the customer site. • To use nondefault parameters and invoke an extended **trace** test, enter the **trace** command without a destination argument. You will be stepped through a dialog to select the desired parameters.

	Command or Action	Purpose
Step 5	**disable** **Example:** `Router# disable`	(Optional) Exits to user EXEC mode.

Configuration Examples for MPLS VPN CSC with BGP

The figure below shows a sample CSC topology for exchanging IPv4 routes and MPLS labels. Use this figure as a reference for configuring and verifying carrier supporting carrier routers to exchange IPv4 routes and MPLS labels.

Figure 4: Sample CSC Topology for Exchanging IPv4 Routes and MPLS Labels



The table below describes the sample configuration shown in the figure above.

Table 1: Description of Sample Configuration Shown in figure 1

Routers	Description
CE1 and CE2	Belong to an end customer. CE1 and CE2 routers exchange routes learned from PE routers. The end customer is purchasing VPN services from a customer carrier.
PE1 and PE2	Part of a customer carrier network that is configured to provide MPLS VPN services. PE1 and PE2 are peering with a VPNv4 IBGP session to form an MPLS VPN network.

Routers	Description
CSC-CE1 and CSC-CE2	Part of a customer carrier network. CSC-CE1 and CSC-CE2 routers exchange IPv4 BGP updates with MPLS labels and redistribute PE loopback addressees to and from the IGP (OSPF in this example). The customer carrier is purchasing carrier supporting carrier VPN services from a backbone carrier.
CSC-PE1 and CSC-PE2	Part of the backbone carrier's network configured to provide carrier supporting carrier VPN services. CSC-PE1 and CSC-PE2 are peering with a VPNv4 IP BGP session to form the MPLS VPN network. In the VRF, CSC-PE1 and CSC-PE2 are peering with the CSC-CE routers, which are configured for carrying MPLS labels with the routes, with an IPv4 EBGP session.

Configuring the Backbone Carrier Core Examples

Configuration and verification examples for the backbone carrier core included in this section are as follows:

Verifying IP Connectivity and LDP Configuration in the CSC Core Example

Check that CSC-PE2 is reachable from CSC-PE1 by entering the following command on CSC-CE1:

```
Router# ping 10.5.5.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Verify the path from CSC-PE1 to CSC-PE2 by entering the following command on CSC-CE1:

```
Router# trace 10.5.5.5
Type escape sequence to abort.
Tracing the route to 10.5.5.5
  1 10.5.5.5 0 msec 0 msec *
```

Check that CSC-PE router prefixes are in the MPLS forwarding table:

```
Router# show mpls forwarding-table
Local     Outgoing     Prefix or          Bytes tag    Outgoing      Next Hop
tag       tag or VC    Tunnel Id          switched     interface
16        2/nn         dd.dd.dd.dd/32       0          AT2/1/0.1     point2point
17        16           bb.bb.bb.bb/32[V]    30204      Et1/0         pp.0.0.1
21        Pop tag      cc.cc.cc.cc/32[V]    0          Et1/0         pp.0.0.1
22        Pop tag      nn.0.0.0/8[V]        570        Et1/0         pp.0.0.1
23        Aggregate    pp.0.0.0/8[V]        0
2         2/nn         gg.gg.gg.gg/32[V]    0          AT3/0.1       point2point
8         2/nn         hh.hh.hh.hh/32[V]    15452      AT3/0.1       point2point
29        2/nn         qq.0.0.0/8[V]        0          AT3/0.1       point2point
30        2/nn         ss.0.0.0/8[V]        0          AT3/0.1       point2point
```

Check the status of LDP discovery processes in the core:

```
Router# show mpls ldp discovery
Local LDP Identifier:
  ee.ee.ee.ee:0
  Discovery Sources:
  Interfaces:
    ATM2/1/0.1 (ldp): xmit/recv
      TDP Id: dd.dd.dd.dd:1
```

Check the status of LDP sessions in the core:

```
Router# show mpls ldp neighbor
Peer LDP Ident: dd.dd.dd.dd:1; Local LDP Ident ee.ee.ee.ee:1
       TCP connection: dd.dd.dd.dd.646 - ee.ee.ee.ee.11007
       State: Oper; Msgs sent/rcvd: 20/21; Downstream on demand
       Up time: 00:14:56
       LDP discovery sources:
         ATM2/1/0.1, Src IP addr: dd.dd.dd.dd
```

Check the forwarding table (prefixes, next-hops, and interfaces):

```
Router# show ip cef
Prefix              Next Hop          Interface
0.0.0.0/0           drop              Null0 (default route handler entry)
0.0.0.0/32          receive
dd.dd.dd.dd/32      dd.dd.dd.dd       ATM2/1/0.1
ee.ee.ee.ee/32      receive
224.0.0.0/4         drop
224.0.0.0/24        receive
255.255.255.255/32 receive
```

Note Also see the Verifying Labels in the CSC-CE Routers Examples, on page 40.

Verify that interfaces are configured to use LDP:

```
Router# show mpls interfaces
Interface              IP            Tunnel   Operational
Ethernet0/1            Yes (ldp)     No       Yes
```

Display the entire routing table, including host IP address, next hop, interface, and so forth:

```
Router# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
Gateway of last resort is not set
     dd.0.0.0/32 is subnetted, 1 subnets
O       dd.dd.dd.dd [110/7] via dd.dd.dd.dd, 00:16:42, ATM2/1/0.1
     ee.0.0.0/32 is subnetted, 1 subnets
C       ee.ee.ee.ee is directly connected, Loopback0
```

Configuring VRFs for CSC-PE Routers Example

The following example shows how to configure a VPN routing and forwarding (VRF) instance for a CSC-PE router:

```
ip cef distributed
ip vrf vpn1
rd 100:1
```

```
route target both 100:1
!
```

Configuring Multiprotocol BGP for VPN Connectivity in the Backbone Carrier Example

The following example shows how to configure Multiprotocol BGP (MP-BGP) for VPN connectivity in the backbone carrier:

```
ip cef distributed
ip vrf vpn1
rd 100:1
route target both 100:1
hostname csc-pe1
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
 !
router bgp 100
. . .
! (BGP IPv4 to CSC-CE router from CSC-PE router)
!
 address-family ipv4 vrf vpn1
 neighbor ss.0.0.2 remote-as 200
 neighbor ss.0.0.2 activate
 neighbor ss.0.0.2 as-override
 neighbor ss.0.0.2 advertisement-interval 5
 neighbor ss.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
!
```

Configuring the Links Between CSC-PE and CSC-CE Routers Examples

This section contains the following examples:

Configuring the CSC-PE Routers Examples

The following example shows how to configure a CSC-PE router:

```
ip cef
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
mpls label protocol ldp
!
interface Loopback0
 ip address dd.dd.dd.dd 255.255.255.255
!
```

```
interface Ethernet3/1
 ip vrf forwarding vpn1
 ip address pp.0.0.2 255.0.0.0
!
interface ATM0/1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache distributed
 atm clock INTERNAL
 no atm enable-ilmi-trap
 no atm ilmi-keepalive
!
interface ATM0/1/0.1 mpls
 ip unnumbered Loopback0
 no ip directed-broadcast
 no atm enable-ilmi-trap
 mpls label protocol ldp
 mpls atm vpi 2-5
 mpls ip
!
router ospf 100
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/1
 network dd.dd.dd.dd 0.0.0.0 area 100
!
router bgp 100
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor ee.ee.ee.ee remote-as 100
 neighbor ee.ee.ee.ee update-source Loopback0
!
address-family vpnv4                                  !VPNv4 session with CSC-PE2
 neighbor ee.ee.ee.ee activate
 neighbor ee.ee.ee.ee send-community extended
 bgp dampening 30
 exit-address-family
!
 address-family ipv4 vrf vpn1
 neighbor pp.0.0.1 remote-as 200
 neighbor pp.0.0.1 activate
 neighbor pp.0.0.1 as-override
 neighbor pp.0.0.1 advertisement-interval 5
 neighbor pp.0.0.1 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Configuring the CSC-CE Routers Examples

The following example shows how to configure a CSC-CE router:

```
ip cef
!
mpls label protocol ldp
!
interface Loopback0
 ip address cc.cc.cc.cc 255.255.255.255
!
interface Ethernet3/0
 ip address pp.0.0.1 255.0.0.0
!
interface Ethernet4/0
 ip address nn.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
```

```
 mpls ip
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets                     !Exchange routes
 redistribute bgp 200 metric 3 subnets              !learned from PE1
 passive-interface ATM1/0
 passive-interface Ethernet3/0
 network cc.cc.cc.cc 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor pp.0.0.2 remote-as 100
 neighbor pp.0.0.2 update-source Ethernet3/0
 no auto-summary
!
address-family ipv4
 redistribute connected
 redistribute ospf 200 metric 4 match internal
 neighbor pp.0.0.2 activate
 neighbor pp.0.0.2 send-label
 no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Verifying Labels in the CSC-PE Routers Examples

The following examples show how to verify the configurations of the CSC-PE routers.

Verify that the BGP session is up and running between the CSC-PE router and the CSC-CE router. Check the data in the State/PfxRcd column to verify that prefixes are learned during each session.

```
Router# show ip bgp vpnv4 all summary
BBGP router identifier 10.5.5.5, local AS number 100
BGP table version is 52, main routing table version 52
12 network entries and 13 paths using 2232 bytes of memory
6 BGP path attribute entries using 336 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
1 BGP extended community entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 16/4 prefixes, 27/14 paths, scan interval 5 secs
Neighbor        V   AS     MsgRcvd MsgSent  TblVer InQ  OutQ Up/Down  State/PfxRcd
10.5.5.5        4  100     7685    7686      52    0       0 21:17:04       6
10.0.0.2        4  200     7676    7678      52    0       0 21:16:43       7
```

Verify that the MPLS interfaces are up and running, and that LDP-enabled interfaces show that LDP is up and running. LDP is turned off on the VRF because EBGP distributes the labels.

```
Router# show mpls interfaces all
Interface              IP            Tunnel   Operational
GigabitEthernet6/0     Yes (ldp)     No       Yes
VRF vpn1:
Ethernet3/1            No            No       Yes
```

Verify that the prefix for the local PE router is in the routing table of the CSC-PE router:

```
Router# show ip route vrf vpn2 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 100", distance 20, metric 4
  Tag 200, type external
  Last update from pp.0.0.2 21:28:39 ago
  Routing Descriptor Blocks:
```

```
  * pp.0.0.2, from pp.0.0.2, 21:28:39 ago
      Route metric is 4, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix for the remote PE router is in the routing table of the CSC-PE router:

```
Router# show ip route vrf vpn2 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 100", distance 200, metric 4
  Tag 200, type internal
  Last update from 10.1.0.0 21:27:39 ago
  Routing Descriptor Blocks:
  * 10.1.0.0 (Default-IP-Routing-Table), from 10.1.0.0, 21:27:39 ago
      Route metric is 4, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefixes for the customer carrier MPLS VPN service provider networks are in the BGP table, and have appropriate labels:

```
Router# show ip bgp vpnv4 vrf vpn2 labels

   Network          Next Hop      In label/Out label
Route Distinguisher: 100:1 (vpn1)
   cc.cc.cc.cc/32   pp.0.0.2        22/imp-null
   bb.bb.bb.bb/32   pp.0.0.2        27/20
   hh.hh.hh.hh/32   ee.ee.ee.ee     34/35
   gg.gg.gg.gg/32   ee.ee.ee.ee     30/30
   nn.0.0.0         pp.0.0.2        23/imp-null
   ss.0.0.0         ee.ee.ee.ee     33/34
   pp.0.0.0         pp.0.0.2        25/aggregate(vpn1)
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.1.0.0
10.1.0.0/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 27
    fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
  via pp.0.0.2, 0 dependencies, recursive
    next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
    valid cached adjacency
    tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}

Router# show ip cef vrf vpn2 10.1.0.0 detail
10.1.0.0/32, version 19, cached adjacency pp.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 27
    fast tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
  via pp.0.0.2, 0 dependencies, recursive
    next hop pp.0.0.2, Ethernet3/1 via pp.0.0.2/32
    valid cached adjacency
    tag rewrite with Et3/1, pp.0.0.2, tags imposed {20}
```

Verify that the prefix of the PE router in the local customer carrier MPLS VPN service provider is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 10.1.0.0
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
27     20          10.1.0.0/32[V]   958048     Et3/1       pp.0.0.2

Router# show mpls forwarding-table vrf vpn2 10.1.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
27     20 10.1.0.0/32[V]            958125     Et3/1      pp.0.0.2
        MAC/Encaps=14/18, MTU=1500, Tag Stack{20}
        00B04A74A05400B0C26E10558847 00014000
        VPN route: vpn1
```

```
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.3.0.0
10.3.0.0/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 34
    fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
  via ee.ee.ee.ee, 0 dependencies, recursive
    next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
    valid cached adjacency
    tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}

Router# show ip cef vrf vpn2 10.3.0.0 detail
hh.hh.hh.hh/32, version 25, cached adjacency rr.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 34
    fast tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
  via ee.ee.ee.ee, 0 dependencies, recursive
    next hop rr.0.0.2, GigabitEthernet6/0 via ee.ee.ee.ee/32
    valid cached adjacency
    tag rewrite with Gi6/0, rr.0.0.2, tags imposed {35}
```

Verify that the prefix of the PE router in the remote customer carrier MPLS VPN service provider is in the MPLS forwarding table:

```
Router# show mpls forwarding-table vrf vpn2 10.3.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
34     35          hh.hh.hh.hh/32[V] 139034     Gi6/0      rr.0.0.2

Router# show mpls forwarding-table vrf vpn2 10.3.0.0 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
34     35          hh.hh.hh.hh/32[V] 139034     Gi6/0      rr.0.0.2
        MAC/Encaps=14/18, MTU=1500, Tag Stack{35}
        00B0C26E447000B0C26E10A88847 00023000
        VPN route: vpn1
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verifying Labels in the CSC-CE Routers Examples

The following examples show how to verify the configurations of the CSC-CE routers.

Verify that the BGP session is up and running:

```
Router# show ip bgp summary
BGP router identifier cc.cc.cc.cc, local AS number 200
BGP table version is 35, main routing table version 35
14 network entries and 14 paths using 2030 bytes of memory
3 BGP path attribute entries using 168 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Dampening enabled. 1 history paths, 0 dampened paths
BGP activity 17/67 prefixes, 29/15 paths, scan interval 60 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
pp.0.0.1        4   100    7615    7613       35    0    0 21:06:19        5
```

Verify that the loopback address of the local PE router is in the routing table:

```
Router# show ip route 10.1.0.0
Routing entry for 10.1.0.0/32
```

```
  Known via "ospf 200", distance 110, metric 101, type intra area
  Redistributing via bgp 200
  Advertised by bgp 200 metric 4 match internal
  Last update from nn.0.0.1 on Ethernet4/0, 00:34:08 ago
  Routing Descriptor Blocks:
  * nn.0.0.1, from bb.bb.bb.bb, 00:34:08 ago, via Ethernet4/0
      Route metric is 101, traffic share count is 1
```

Verify that the loopback address of the remote PE router is in the routing table:

```
Router# show ip route 10.5.5.5
Routing entry for 10.5.5.5/32
  Known via "bgp 200", distance 20, metric 0
  Tag 100, type external
  Redistributing via ospf 200
  Advertised by ospf 200 metric 3 subnets
  Last update from pp.0.0.1 00:45:16 ago
  Routing Descriptor Blocks:
  * pp.0.0.1, from pp.0.0.1, 00:45:16 ago
      Route metric is 0, traffic share count is 1
      AS Hops 2, BGP network version 0
```

Verify that the prefix of the local PE router is in the MPLS LDP bindings:

```
Router# show mpls ldp bindings 10.1.0.0 255.255.255.255
  tib entry: 10.1.0.0/32, rev 20
        local binding:  tag: 20
        remote binding: tsr: 10.1.0.0:0, tag: imp-null
```

Verify that the prefix of the local PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.1.0.0
10.1.0.0/32, version 46, cached adjacency nn.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 20
  via nn.0.0.1, Ethernet4/0, 0 dependencies
    next hop nn.0.0.1, Ethernet4/0
    unresolved
    valid cached adjacency
    tag rewrite with Et4/0, nn.0.0.1, tags imposed {}
```

Verify that the prefix of the local PE router is in the MPLS forwarding table:

```
Router# show mpls forwarding-table 10.1.0.0
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
20     Pop tag     bb.bb.bb.bb/32   893397     Et4/0      nn.0.0.1

Router# show mpls forwarding-table 10.1.0.0 detail
Local  Outgoing    Prefix           Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
20     Pop tag     bb.bb.bb.bb/32   893524     Et4/0      nn.0.0.1
        MAC/Encaps=14/14, MTU=1504, Tag Stack{}
        00074F83685400B04A74A0708847
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Verify that the BGP routing table contains labels for prefixes in the customer carrier MPLS VPN service provider networks:

```
Router# show ip bgp labels
Network          Next Hop         In Label/Out Label
cc.cc.cc.cc/32   0.0.0.0          imp-null/exp-null
bb.bb.bb.bb/32   nn.0.0.1         20/exp-null
hh.hh.hh.hh/32   pp.0.0.1         26/34
gg.gg.gg.gg/32   pp.0.0.1         23/30
nn.0.0.0         0.0.0.0          imp-null/exp-null
ss.0.0.0         pp.0.0.1         25/33
pp.0.0.0         0.0.0.0          imp-null/exp-null
pp.0.0.1/32      0.0.0.0          16/exp-null
```

Verify that the prefix of the remote PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.5.5.5
10.5.5.5/32, version 54, cached adjacency pp.0.0.1
0 packets, 0 bytes
  tag information set
    local tag: 26
    fast tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
  via pp.0.0.1, 0 dependencies, recursive
    next hop pp.0.0.1, Ethernet3/0 via pp.0.0.1/32
    valid cached adjacency
    tag rewrite with Et3/0, pp.0.0.1, tags imposed {34}
```

Verify that the prefix of the remote PE router is in the MPLS forwarding table:

```
Router# show mpls forwarding-table 10.5.5.5
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     34          hh.hh.hh.hh/32    81786      Et3/0      pp.0.0.1

Router# show mpls forwarding-table 10.5.5.5 detail
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
26     34          hh.hh.hh.hh/32    81863      Et3/0      pp.0.0.1
        MAC/Encaps=14/18, MTU=1500, Tag Stack{34}
        00B0C26E105500B04A74A0548847 00022000
        No output feature configured
    Per-packet load-sharing, slots: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Configuring the Customer Carrier Network Examples

Customer carrier configuration and verification examples in this section include:

Verifying IP Connectivity in the Customer Carrier Example

Verify the connectivity from one customer carrier core router to another (from CE1 to CE2) by entering the following command:

```
Router# ping 10.2.0.0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to jj.jj.jj.jj, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/12 ms
```

Verify the path that a packet goes through on its way to its final destination from CE1 to CE2:

```
Router# trace 10.2.0.0
Type escape sequence to abort.
Tracing the route to 10.2.0.0
  1 mm.0.0.2 0 msec 0 msec 4 msec
  2 nn.0.0.2 [MPLS: Labels 20/21 Exp 0] 8 msec 8 msec 12 msec
  3 pp.0.0.2 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 12 msec
  4 ss.0.0.1 [MPLS: Labels 17/21 Exp 0] 8 msec 8 msec 12 msec
  5 ss.0.0.2 [MPLS: Labels 16/21 Exp 0] 8 msec 8 msec 12 msec
  6 tt.0.0.1 [AS 200] [MPLS: Label 21 Exp 0] 8 msec 8 msec 8 msec
  7 tt.0.0.2 [AS 200] 8 msec 4 msec *
```

Verify the path that a packet goes through on its way to its final destination from CE2 to CE1:

```
Router# trace 10.1.0.0
Type escape sequence to abort.
Tracing the route to 10.1.0.0
  1 tt.0.0.1 0 msec 0 msec 0 msec
  2 qq.0.0.2 [MPLS: Labels 18/21 Exp 0] 8 msec 12 msec 12 msec
  3 ss.0.0.1 [MPLS: Labels 28/21 Exp 0] 8 msec 8 msec 8 msec
```

```
  4 pp.0.0.2 [MPLS: Labels 17/21 Exp 0] 12 msec 8 msec 8 msec
  5 pp.0.0.1 [MPLS: Labels 16/21 Exp 0] 12 msec 12 msec 8 msec
  6 mm.0.0.2 [AS 200] [MPLS: Label 21 Exp 0] 12 msec 8 msec 12 msec
  7 mm.0.0.1 [AS 200] 4 msec 4 msec *
```

Configuring a Customer Carrier Core Router as a Route Reflector Example

The following example shows how to use an address family to configure internal BGP peer 10.1.1.1 as a route-reflector client for both unicast and multicast prefixes:

```
router bgp 200
 address-family vpnv4
  neighbor 10.1.1.1 activate
  neighbor 10.1.1.1 route-reflector-client

router bgp 100
 address-family vpnv4
  neighbor xx.xx.xx.xx activate
  neighbor xx.xx.xx.xx route-reflector-client
   ! xx.xx.xx,xx is a PE router
  neighbor xx.xx.xx.xx send-community extended
 exit address-family
! You need to configure your peer BGP neighbor.
```

Configuring the Customer Site for Hierarchical VPNs Examples

This section contains the following configuration and verification examples for the customer site:

Configuring PE Routers for Hierarchical VPNs Examples

This example shows how to configure a PE router:

```
ip cef
!
ip vrf vpn2
 rd 200:1
 route-target export 200:1
 route-target import 200:1
mpls label protocol ldp
!
interface Loopback0
 ip address bb.bb.bb.bb 255.255.255.255
!
interface Ethernet3/0
 ip address nn.0.0.1 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
 mpls label protocol ldp
 mpls ip
!
interface Ethernet3/3
 ip vrf forwarding vpn2
 ip address mm.0.0.2 255.0.0.0
 no ip directed-broadcast
 no ip mroute-cache
!
router ospf 200
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 redistribute connected subnets
 passive-interface Ethernet3/3
 network bb.bb.bb.bb 0.0.0.0 area 200
 network nn.0.0.0 0.255.255.255 area 200
```

```
!
router bgp 200
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 timers bgp 10 30
 neighbor hh.hh.hh.hh remote-as 200
 neighbor hh.hh.hh.hh update-source Loopback0
!
address-family vpnv4                                  !VPNv4 session with PE2
 neighbor hh.hh.hh.hh activate
 neighbor hh.hh.hh.hh send-community extended
 bgp dampening 30
 exit-address-family
!
 address-family ipv4 vrf vpn2
 neighbor mm.0.0.1 remote-as 300
 neighbor mm.0.0.1 activate
 neighbor mm.0.0.1 as-override
 neighbor mm.0.0.1 advertisement-interval 5
no auto-summary
 no synchronization
 bgp dampening 30
 exit-address-family
```

Verifying Labels in Each PE Router for Hierarchical VPNs Examples

The following examples show how to verify the configuration of PE router in hierarchical VPNs.

Verify that the loopback address of the local CE router is in the routing table of the PE1 router:

```
Router# show ip route vrf vpn2 10.2.2.2
Routing entry for 10.2.2.2/32
  Known via "bgp 200", distance 20, metric 0
  Tag 300, type external
  Last update from mm.0.0.2 20:36:59 ago
  Routing Descriptor Blocks:
  * mm.0.0.2, from mm.0.0.2, 20:36:59 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix for the local CE router is in the MPLS forwarding table, and that the prefix is untagged:

```
Router# show mpls forwarding-table vrf vpn2 10.2.2.2
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
23     Untagged    aa.aa.aa.aa/32[V] 0          Et3/3      mm.0.0.2
```

Verify that the prefix of the remote PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.5.5.5

10.5.5.5/32, version 31, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: 31
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
  via nn.0.0.2, Ethernet3/0, 2 dependencies
    next hop nn.0.0.2, Ethernet3/0
    unresolved
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26}
```

Verify that the loopback address of the remote CE router is in the routing table:

```
Router# show ip route vrf vpn2 10.2.0.0
Routing entry for 10.2.0.0/32
  Known via "bgp 200", distance 200, metric 0
  Tag 300, type internal
  Last update from hh.hh.hh.hh 20:38:49 ago
  Routing Descriptor Blocks:
```

```
  * hh.hh.hh.hh (Default-IP-Routing-Table), from hh.hh.hh.hh, 20:38:49 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1, BGP network version 0
```

Verify that the prefix of the remote CE router is in the MPLS forwarding table, and that an outgoing interface exists:

```
Router# show mpls forwarding-table vrf vpn2 10.2.0.0
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
None   26          jj.jj.jj.jj/32    0          Et3/0      nn.0.0.2
```

Verify that the prefix of the remote CE router is in the Cisco Express Forwarding table:

```
Router# show ip cef vrf vpn2 10.2.0.0
10.2.0.0/32, version 12, cached adjacency nn.0.0.2
0 packets, 0 bytes
  tag information set
    local tag: VPN route head
    fast tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
  via hh.hh.hh.hh, 0 dependencies, recursive
    next hop nn.0.0.2, Ethernet3/0 via hh.hh.hh.hh/32
    valid cached adjacency
    tag rewrite with Et3/0, nn.0.0.2, tags imposed {26 32}
```

Verify that the prefix of the local PE router is in the Cisco Express Forwarding table:

```
Router# show ip cef 10.1.0.0
10.1.0.0/32, version 9, connected, receive
  tag information set
    local tag: implicit-null
```

Configuring CE Routers for Hierarchical VPNs Examples

The following example shows how to configure a CE router:

```
ip cef distributed
interface Loopback0
ip address 10.3.0.0 255.255.255.255
!
interface FastEthernet0/3/3
 ip address mm.0.0.1 255.0.0.0
!
router bgp 300
 no synchronization
 bgp log-neighbor-changes
 timers bgp 10 30
 redistribute connected                              !Redistributing routes into BGP
 neighbor mm.0.0.2 remote-as 200                     !to send to PE1
 neighbor mm.0.0.2 advertisement-interval 5
 no auto-summary
```

Verifying IP Connectivity in the Customer Site Examples

The following examples show how to verify IP connectivity at the customer site.

Verify that the loopback address of the remote CE router, learned from the PE router, is in the routing table of the local router:

```
Router# show ip route 10.2.0.0
Routing entry for 10.2.0.0/32
  Known via "bgp 300", distance 20, metric 0
  Tag 200, type external
  Redistributing via ospf 300
  Advertised by ospf 300 subnets
  Last update from mm.0.0.1 20:29:35 ago
```

```
Routing Descriptor Blocks:
* mm.0.0.1, from mm.0.0.1, 20:29:35 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2
```

Additional References

Related Documents

Related Topic	Document Title
LDP	MPLS Label Distribution Protocol
MPLS	MPLS Product Literature

Standards

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

MIBs

MIB	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 locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1164	*Application of the Border Gateway Protocol in the Internet*
RFC 1171	*A Border Gateway Protocol 4*
RFC 1700	*Assigned Numbers*
RFC 1966	*BGP Route Reflection: An Alternative to Full Mesh IBGP*

RFC	Title
RFC 2283	*Multiprotocol Extensions for BGP-4*
RFC 2547	*BGP/MPLS VPNs*
RFC 2842	*Capabilities Advertisement with BGP-4*
RFC 2858	*Multiprotocol Extensions for BGP-4*
RFC 3107	*Carrying Label Information in BGP-4*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS VPN CSC with BGP

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

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

Table 2: Feature Information for MPLS VPN CSC with BGP

Feature Name	Releases	Feature Information
MPLS VPN--Carrier Supporting Carrier--IPv4 BGP Label Distribution	12.0(21)ST 12.0(22)S 12.0(23)S 12.2(13)T 12.0(24)S 12.2(14)S 12.0(27)S 12.0(29)S Cisco IOS XE Release 2.2	This feature enables you to create an MPLS VPN CSC network that uses BGP to transport routes and MPLS labels. In 12.0(21)ST, this feature was introduced. In 12.0(22)S, this feature was integrated. In 12.0(23)S, this feature was integrated. In 12.2(13)T, this feature was integrated. 12.0(24)S, this feature was integrated. In 12.2(14)S, this feature was integrated. In 12.0(27)S, this feature was integrated. In 12.0(29)S, this feature was integrated. In Cisco IOS XE Release 2.2, this feature was implemented on the Cisco ASR 1000 Series Routers. This feature uses no new or modified commands.

Glossary

ASBR -- Autonomous System Boundary router. A router that connects one autonomous system to another.

autonomous system --A collection of networks under a common administration sharing a common routing strategy.

BGP --Border Gateway Protocol. An interdomain routing protocol that exchanges network reachability information with other BGP systems (which may be within the same autonomous system or between multiple autonomous systems).

CE router--customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router. CE routers do not recognize associated MPLS VPNs.

CSC --Carrier Supporting Carrier. A hierarchical VPN model that allows small service providers, or customer carriers, to interconnect their IP or MPLS networks over an MPLS backbone. This eliminates the need for customer carriers to build and maintain their own MPLS backbone.

eBGP --external Border Gateway Protocol. A BGP between routers located within different autonomous systems. When two routers, located in different autonomous systems, are more than one hop away from one another, the eBGP session between the two routers is considered a multihop BGP.

edge router--A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

iBGP --internal Border Gateway Protocol. A BGP between routers within the same autonomous system.

IGP --Interior Gateway Protocol. Internet protocol used to exchange routing information within a single autonomous system. Examples of common Internet IGP protocols include IGRP, OSPF, IS-IS, and RIP.

IP --Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

LDP --Label Distribution Protocol. A standard protocol between MPLS-enabled routers to negotiate the labels (addresses) used to forward packets.

LFIB --Label Forwarding Information Base. Data structure used in MPLS to hold information about incoming and outgoing labels and associated Forwarding Equivalence Class (FEC) packets.

MP-BGP --Multiprotocol BGP.

MPLS --Multiprotocol Label Switching. The name of the IETF working group responsible for label switching, and the name of the label switching approach it has standardized.

NLRI --Network Layer Reachability Information. The BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and extended community values.

NSF --Nonstop forwarding enables routers to continuously forward IP packets following a Route Processor takeover or switchover to another Route Processor. NSF maintains and updates Layer 3 routing and forwarding information in the backup Route Processor to ensure that IP packets and routing protocol information are forwarded continuously during the switchover and route convergence process.

PE router--provider edge router. A router that is part of a service provider's network. It is connected to a customer edge (CE) router. All MPLS VPN processing occurs in the PE router.

QoS --quality of service. Measure of performance for a transmission system that indicates its transmission quality and service availability.

RD --route distinguisher. An 8-byte value that is concatenated with an IPv4 prefix to create a unique VPN-IPv4 prefix.

RT --route target. Extended community attribute used to identify the VRF routing table into which a prefix is imported.

SLA --Service Level Agreement given to VPN subscribers.

VPN --Virtual Private Network. A secure MPLS-based network that shares resources on one or more physical networks (typically implemented by one or more service providers). A VPN contains geographically dispersed sites that can communicate securely over a shared backbone network.

VRF --VPN routing and forwarding instance. Routing information that defines a VPN site that is attached to a PE router. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table.