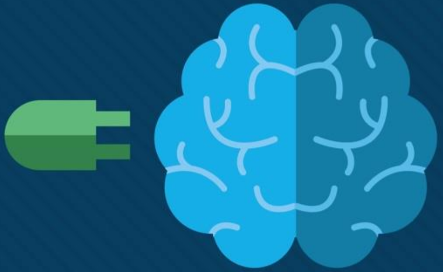
Chapter 18: VRF, MPLS, and MPLS Layer 3 VPNs

Instructor Materials



CCNP Enterprise: Advanced Routing

Chapter 18 Content

This chapter covers the following content:

- **Implementing and Verifying VRF-Lite -** This section introduces VRF and how to configure and verify a VRF-Lite implementation.
- **An Introduction to MPLS Operations -** This section introduces MPLS and explores the main MPLS topics, such as LSRs, LDP, LSP, and label switching.
- **An Introduction to MPLS Layer 3 VPNs -** This section introduces the concept of MPLS Layer 3 VPNs.

Implementing and Verifying VRF-Lite

- VRF is a technology for creating separate virtual routers on a single physical router.
- VRF-Lite provides VRF without MPLS. Router interfaces, routing tables, and forwarding tables are isolated on an instance-by-instance basis and therefore prevent traffic from one VRF instance from interfering with another VRF instance.
- VRF is an essential component of the MPLS L3VPN architecture and provides increased router functionality through segmentation in lieu of using multiple devices.
- This section introduces you to VRF and demonstrates how you can configure and verify VRF-Lite in a Cisco network.

Implementing and Verifying VRF-Lite

VRF-Lite Overview

- By default, all router interfaces, the routing table, and any forwarding tables are associated with the global VRF instance.
- What you’ve been calling your routing table is actually the routing table of the global VRF instance.
- If you need to divide your router up into multiple virtual routers, you can do so by creating additional VRF instances, which also creates additional routing and forwarding tables.

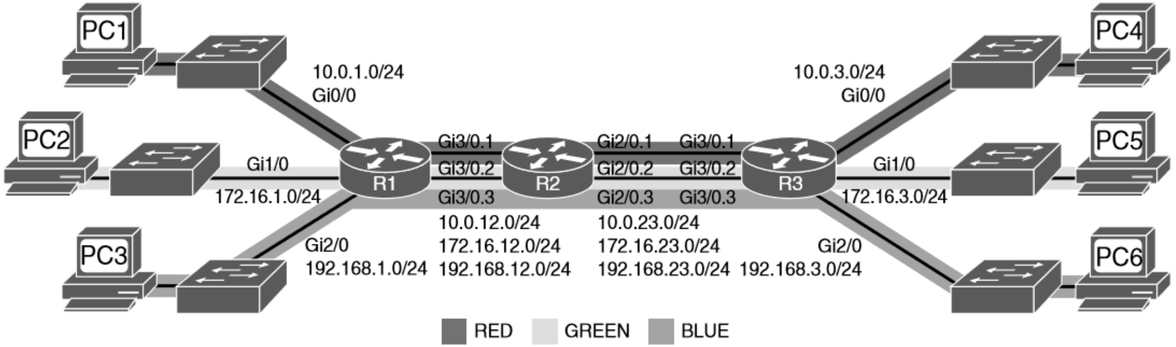


Figure 18-1 *An Example of Three VRF Instances (from top to bottom: red, green, and blue)*

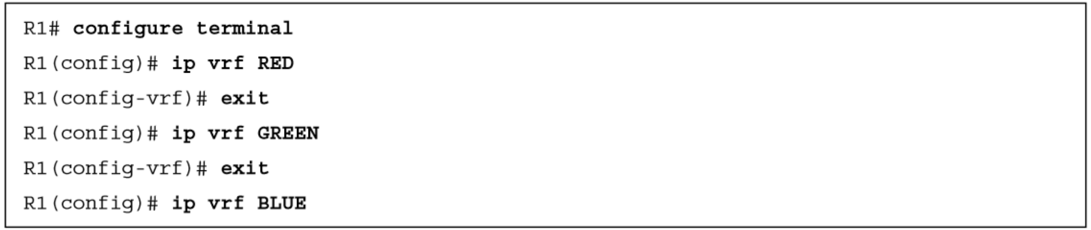
Consider this scenario: for security reasons, you need to build three different networks so that traffic in each network is isolated from traffic in the other networks. However, you only want to build a single physical network to accomplish this. You can do this by using VRF. Figure 18-1 shows a single physical topology that is divided into three different logically isolated networks.

Creating and Verifying VRF Instances

Example 18-1 shows how the **ip vrf** *vrf-name* command is used on each of the routers to create the VRF instances.

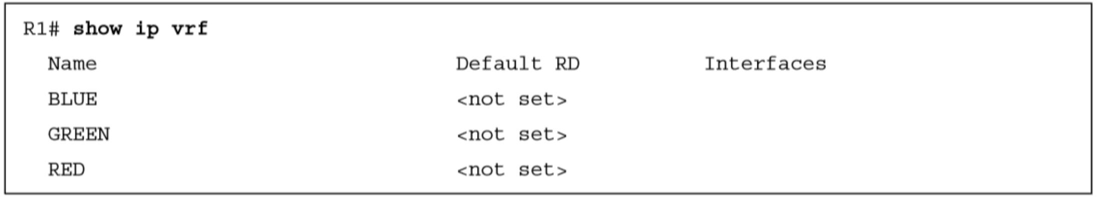
To verify that the VRF instances are created, use the **show ip vrf** command as shown in Example 18-2 for R1. Notice that the Interfaces column is empty. You need to assign interfaces to each of the VRF instances to separate and isolate the traffic.

Example 18-1 *Configuring VRF Instances on R1 with the* ***ip vrf*** *Command*



```
R1# configure terminal
R1(config)# ip vrf RED
R1(config-vrf)# exit
R1(config)# ip vrf GREEN
R1(config-vrf)# exit
R1(config)# ip vrf BLUE
```

Example 18-2 *Verifying That the VRF Instances Are Configured on R1*



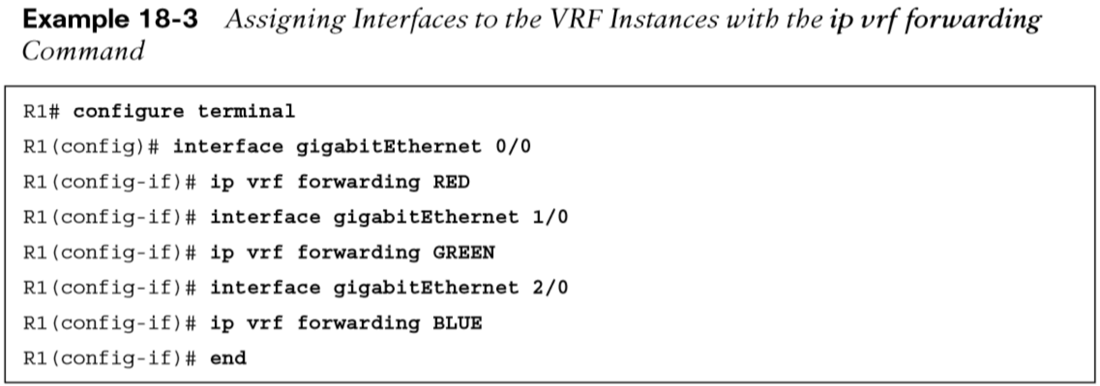
```
R1# show ip vrf
  Name                              Default RD          Interfaces
  BLUE                              <not set>
  GREEN                             <not set>
  RED                               <not set>
```

Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

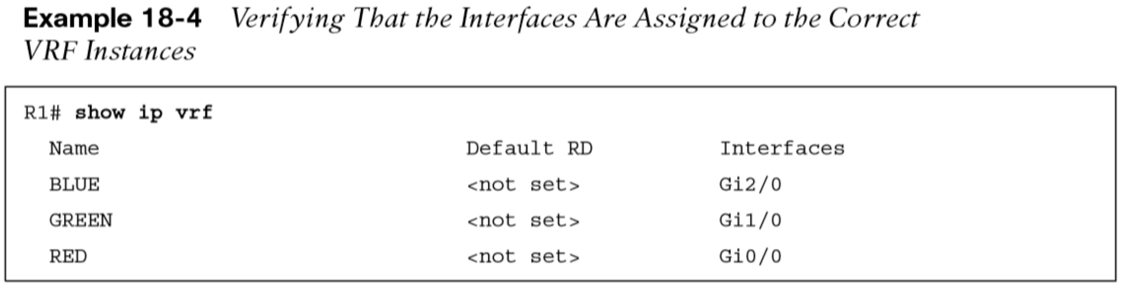
To assign an interface to a VRF, use the **ip vrf forwarding** *vrf-name* command in interface configuration mode, as shown in in Example 18-3.

Using the **show ip vrf** command again, you can verify that each interface has been assigned to the correct VRF instance, as shown in Example 18-4.



Example 18-3 *Assigning Interfaces to the VRF Instances with the* ***ip vrf forwarding*** *Command*

```
R1# configure terminal
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# ip vrf forwarding RED
R1(config-if)# interface gigabitEthernet 1/0
R1(config-if)# ip vrf forwarding GREEN
R1(config-if)# interface gigabitEthernet 2/0
R1(config-if)# ip vrf forwarding BLUE
R1(config-if)# end
```



Example 18-4 *Verifying That the Interfaces Are Assigned to the Correct VRF Instances*

```
R1# show ip vrf
  Name                             Default RD          Interfaces
  BLUE                             <not set>           Gi2/0
  GREEN                            <not set>           Gi1/0
  RED                              <not set>           Gi0/0
```

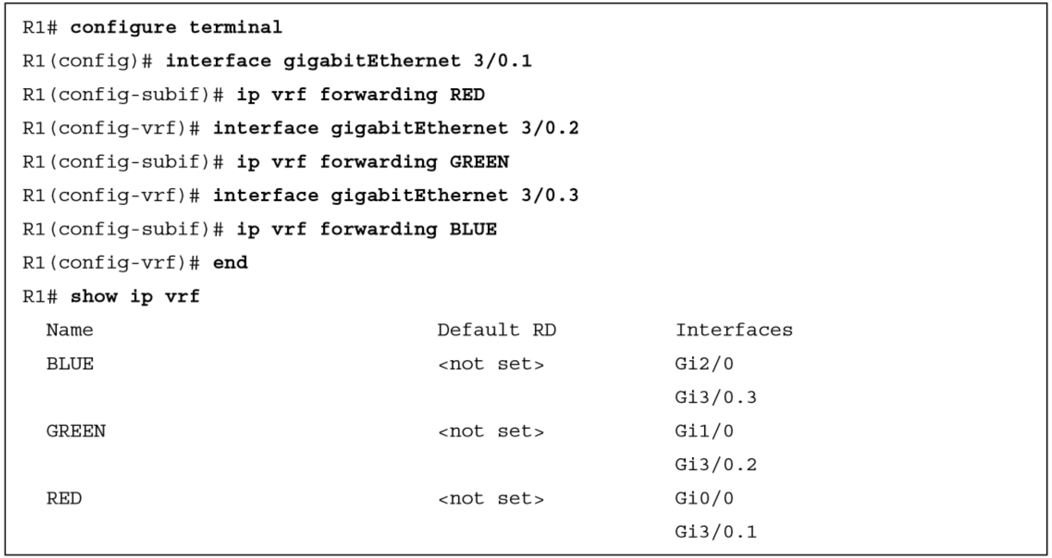
Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

If a single physical interface supports multiple VRF instances, the physical interface needs to be broken into subinterfaces. Therefore, Gi3/0 needs to be broken into subinterfaces.

Example 18-5 shows how to create the subinterfaces and assign them to the correct VRF instances. It also shows the use of the **show ip vrf** command to verify that the interfaces are in the correct VRF instances.

Example 18-5 *Creating Subinterfaces on R1 and Assigning Them to the Correct VRF Instances*



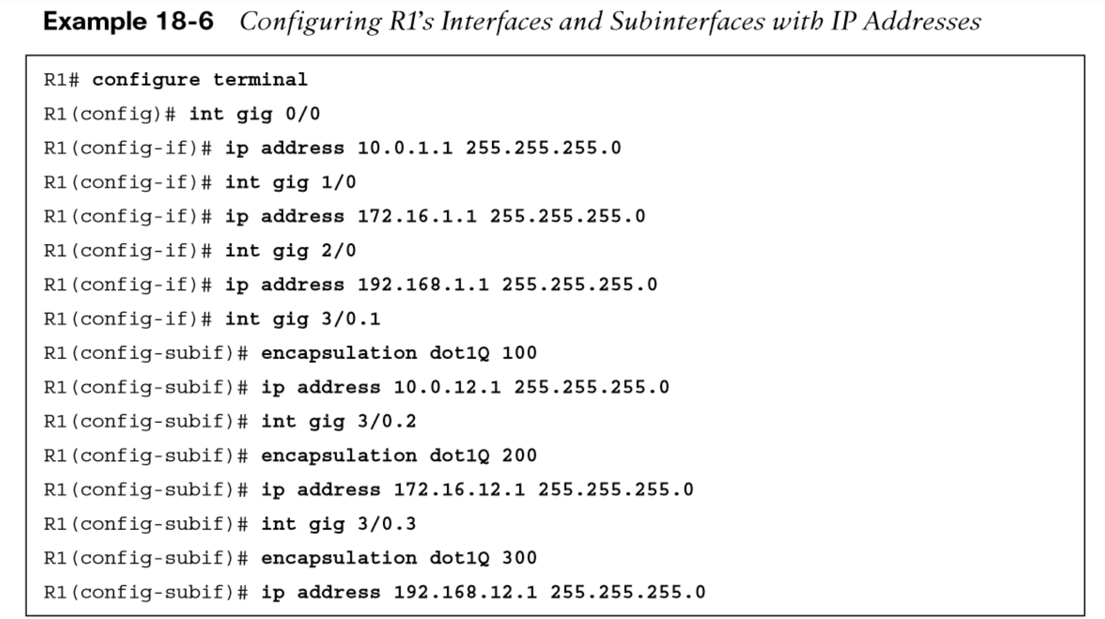
```
R1# configure terminal
R1(config)# interface gigabitEthernet 3/0.1
R1(config-subif)# ip vrf forwarding RED
R1(config-vrf)# interface gigabitEthernet 3/0.2
R1(config-subif)# ip vrf forwarding GREEN
R1(config-vrf)# interface gigabitEthernet 3/0.3
R1(config-subif)# ip vrf forwarding BLUE
R1(config-vrf)# end
R1# show ip vrf
  Name                              Default RD          Interfaces
  BLUE                              <not set>           Gi2/0
                                                        Gi3/0.3
  GREEN                             <not set>           Gi1/0
                                                        Gi3/0.2
  RED                               <not set>           Gi0/0
                                                        Gi3/0.1
```

Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

Next, you can configure network addressing on R1. Example 18-6 shows the IP addressing configuration of each of the interfaces on R1, based on Figure 18-1.

Notice that the subinterfaces must be configured with dot1q encapsulation, or you can’t assign an IP address to the interface. Also, note that when you configure R2’s subinterfaces connecting to R1, they need to be configured with the same VLAN numbers.



Example 18-6 *Configuring R1’s Interfaces and Subinterfaces with IP Addresses*

```
R1# configure terminal
R1(config)# int gig 0/0
R1(config-if)# ip address 10.0.1.1 255.255.255.0
R1(config-if)# int gig 1/0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# int gig 2/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# int gig 3/0.1
R1(config-subif)# encapsulation dot1Q 100
R1(config-subif)# ip address 10.0.12.1 255.255.255.0
R1(config-subif)# int gig 3/0.2
R1(config-subif)# encapsulation dot1Q 200
R1(config-subif)# ip address 172.16.12.1 255.255.255.0
R1(config-subif)# int gig 3/0.3
R1(config-subif)# encapsulation dot1Q 300
R1(config-subif)# ip address 192.168.12.1 255.255.255.0
```

You can use the **show ip vrf interfaces** command to verify the IP address assigned to the interface, the VRF instance the interface is in, and whether the interface is up or down.

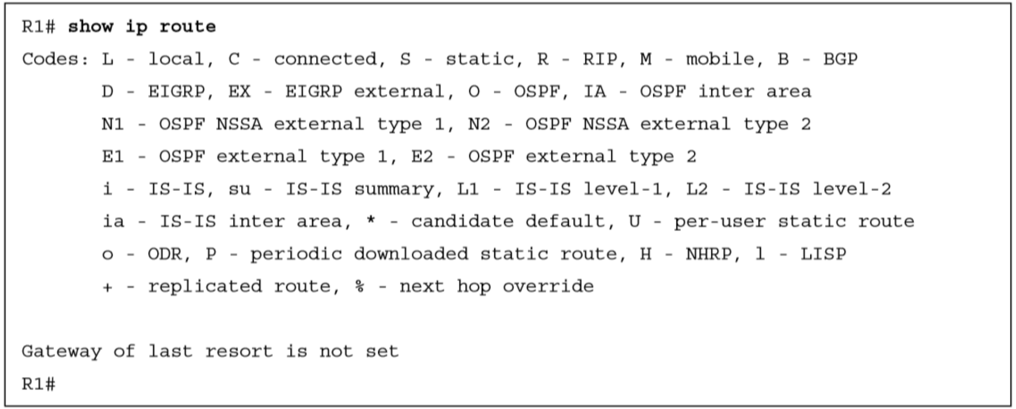
Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

As soon as you created the VRF instance with the **ip vrf** *vrf-name* command, the virtual routing table was created for the network. You can use the **show ip route** command to display the global routing table, as shown in Example 18-8.

To view the routing table for a VRF instance, use the show **ip route vrf** *vrf-name* command.

Example 18-8 *Verifying the Global Routing Table*

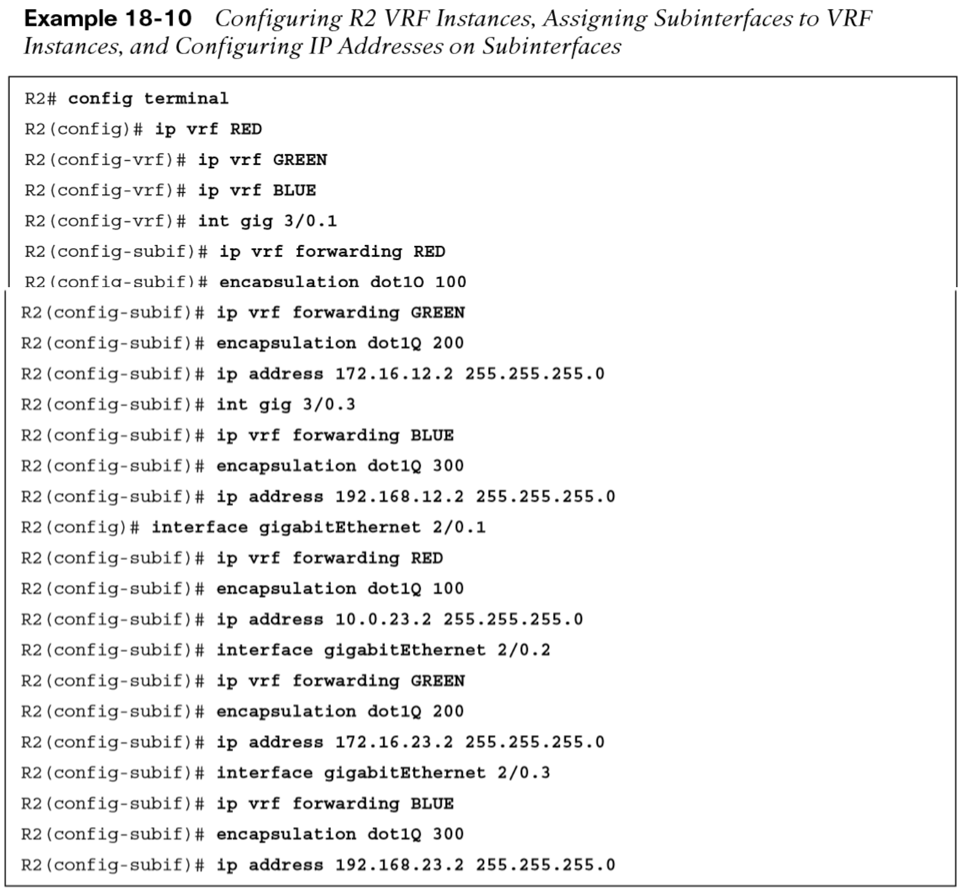


```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set
R1#
```

Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

You can now configure R2. Example 18-10 shows the configuration required on R2.



Example 18-10 *Configuring R2 VRF Instances, Assigning Subinterfaces to VRF Instances, and Configuring IP Addresses on Subinterfaces*

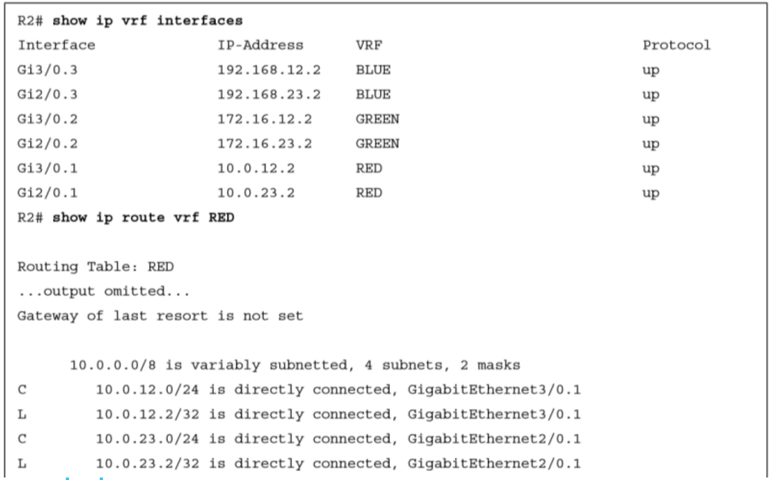
```
R2# config terminal
R2(config)# ip vrf RED
R2(config-vrf)# ip vrf GREEN
R2(config-vrf)# ip vrf BLUE
R2(config-vrf)# int gig 3/0.1
R2(config-subif)# ip vrf forwarding RED
R2(config-subif)# encapsulation dot1Q 100
R2(config-subif)# ip vrf forwarding GREEN
R2(config-subif)# encapsulation dot1Q 200
R2(config-subif)# ip address 172.16.12.2 255.255.255.0
R2(config-subif)# int gig 3/0.3
R2(config-subif)# ip vrf forwarding BLUE
R2(config-subif)# encapsulation dot1Q 300
R2(config-subif)# ip address 192.168.12.2 255.255.255.0
R2(config)# interface gigabitEthernet 2/0.1
R2(config-subif)# ip vrf forwarding RED
R2(config-subif)# encapsulation dot1Q 100
R2(config-subif)# ip address 10.0.23.2 255.255.255.0
R2(config-subif)# interface gigabitEthernet 2/0.2
R2(config-subif)# ip vrf forwarding GREEN
R2(config-subif)# encapsulation dot1Q 200
R2(config-subif)# ip address 172.16.23.2 255.255.255.0
R2(config-subif)# interface gigabitEthernet 2/0.3
R2(config-subif)# ip vrf forwarding BLUE
R2(config-subif)# encapsulation dot1Q 300
R2(config-subif)# ip address 192.168.23.2 255.255.255.0
```

Implementing and Verifying VRF-Lite

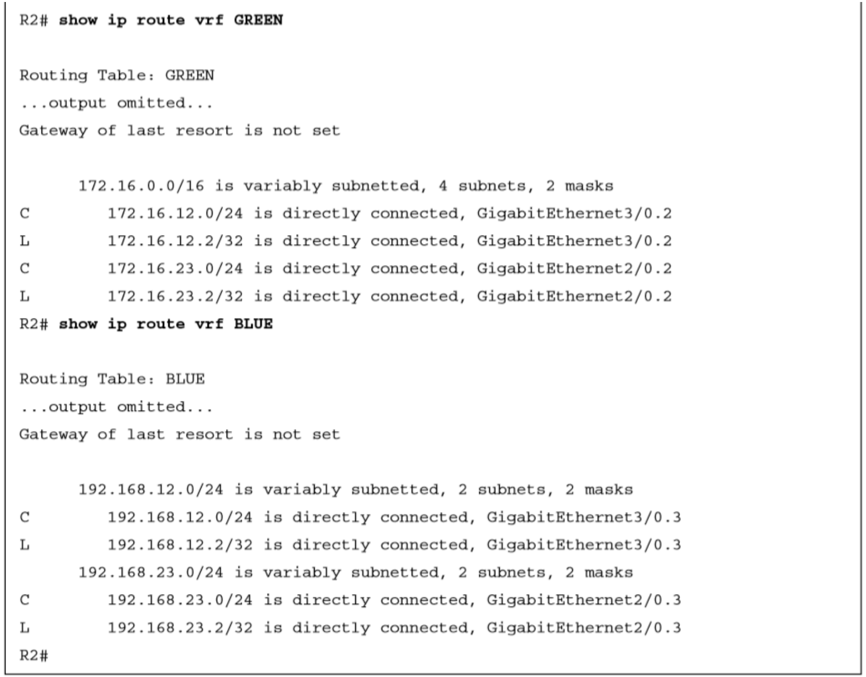
Creating and Verifying VRF Instances (Cont.)

Example 18-11 displays the output of the **show ip vrf interfaces** command and the **show ip route vrf** *vrf_name* commands to verify that R2 has been configured correctly.

Example 18-11 *Verifying R2's Configuration with the* ***show ip vrf interfaces*** *Command and the* ***show ip route vrf*** *Command*



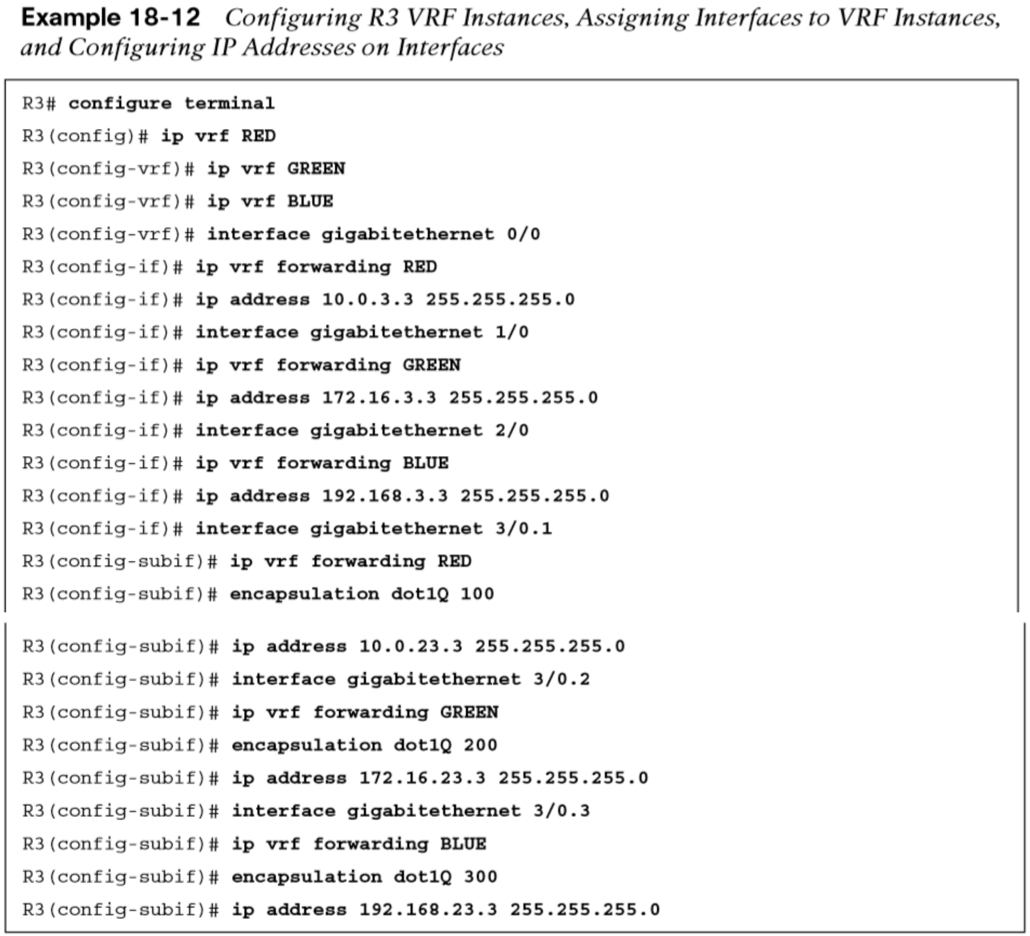
```
R2# show ip vrf interfaces
Interface              IP-Address      VRF                              Protocol
Gi3/0.3                192.168.12.2    BLUE                             up
Gi2/0.3                192.168.23.2    BLUE                             up
Gi3/0.2                172.16.12.2     GREEN                            up
Gi2/0.2                172.16.23.2     GREEN                            up
Gi3/0.1                10.0.12.2       RED                              up
Gi2/0.1                10.0.23.2       RED                              up
R2# show ip route vrf RED

Routing Table: RED
...output omitted...
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.12.0/24 is directly connected, GigabitEthernet3/0.1
L        10.0.12.2/32 is directly connected, GigabitEthernet3/0.1
C        10.0.23.0/24 is directly connected, GigabitEthernet2/0.1
L        10.0.23.2/32 is directly connected, GigabitEthernet2/0.1
```



```
R2# show ip route vrf GREEN

Routing Table: GREEN
...output omitted...
Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.16.12.0/24 is directly connected, GigabitEthernet3/0.2
L        172.16.12.2/32 is directly connected, GigabitEthernet3/0.2
C        172.16.23.0/24 is directly connected, GigabitEthernet2/0.2
L        172.16.23.2/32 is directly connected, GigabitEthernet2/0.2
R2# show ip route vrf BLUE

Routing Table: BLUE
...output omitted...
Gateway of last resort is not set

      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet3/0.3
L        192.168.12.2/32 is directly connected, GigabitEthernet3/0.3
      192.168.23.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.23.0/24 is directly connected, GigabitEthernet2/0.3
L        192.168.23.2/32 is directly connected, GigabitEthernet2/0.3
R2#
```

Creating and Verifying VRF Instances (Cont.)

Now you can configure R3. Example 18-12 shows the configuration required on R3.



Example 18-12 *Configuring R3 VRF Instances, Assigning Interfaces to VRF Instances, and Configuring IP Addresses on Interfaces*

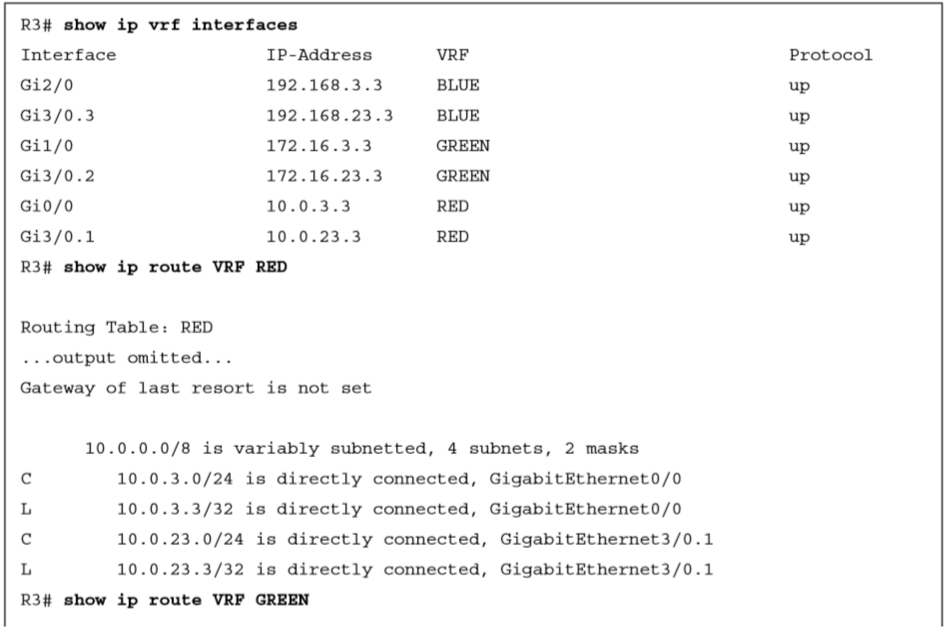
```
R3# configure terminal
R3(config)# ip vrf RED
R3(config-vrf)# ip vrf GREEN
R3(config-vrf)# ip vrf BLUE
R3(config-vrf)# interface gigabitethernet 0/0
R3(config-if)# ip vrf forwarding RED
R3(config-if)# ip address 10.0.3.3 255.255.255.0
R3(config-if)# interface gigabitethernet 1/0
R3(config-if)# ip vrf forwarding GREEN
R3(config-if)# ip address 172.16.3.3 255.255.255.0
R3(config-if)# interface gigabitethernet 2/0
R3(config-if)# ip vrf forwarding BLUE
R3(config-if)# ip address 192.168.3.3 255.255.255.0
R3(config-if)# interface gigabitethernet 3/0.1
R3(config-subif)# ip vrf forwarding RED
R3(config-subif)# encapsulation dot1Q 100
R3(config-subif)# ip address 10.0.23.3 255.255.255.0
R3(config-subif)# interface gigabitethernet 3/0.2
R3(config-subif)# ip vrf forwarding GREEN
R3(config-subif)# encapsulation dot1Q 200
R3(config-subif)# ip address 172.16.23.3 255.255.255.0
R3(config-subif)# interface gigabitethernet 3/0.3
R3(config-subif)# ip vrf forwarding BLUE
R3(config-subif)# encapsulation dot1Q 300
R3(config-subif)# ip address 192.168.23.3 255.255.255.0
```

Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

Example 18-13 shows the output of the **show ip vrf interfaces** command and the **show ip route vrf** command to verify that R3 has been configured correctly.

Example 18-13 *Verifying R3's Configuration with the show ip vrf interfaces Command and the show ip route vrf Command*



```
R3# show ip vrf interfaces
Interface              IP-Address      VRF                              Protocol
Gi2/0                  192.168.3.3     BLUE                             up
Gi3/0.3                192.168.23.3    BLUE                             up
Gi1/0                  172.16.3.3      GREEN                            up
Gi3/0.2                172.16.23.3     GREEN                            up
Gi0/0                  10.0.3.3        RED                              up
Gi3/0.1                10.0.23.3       RED                              up
R3# show ip route VRF RED

Routing Table: RED
...output omitted...
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.3.0/24 is directly connected, GigabitEthernet0/0
L        10.0.3.3/32 is directly connected, GigabitEthernet0/0
C        10.0.23.0/24 is directly connected, GigabitEthernet3/0.1
L        10.0.23.3/32 is directly connected, GigabitEthernet3/0.1
R3# show ip route VRF GREEN
```

Above is a partial image of Example 18-13

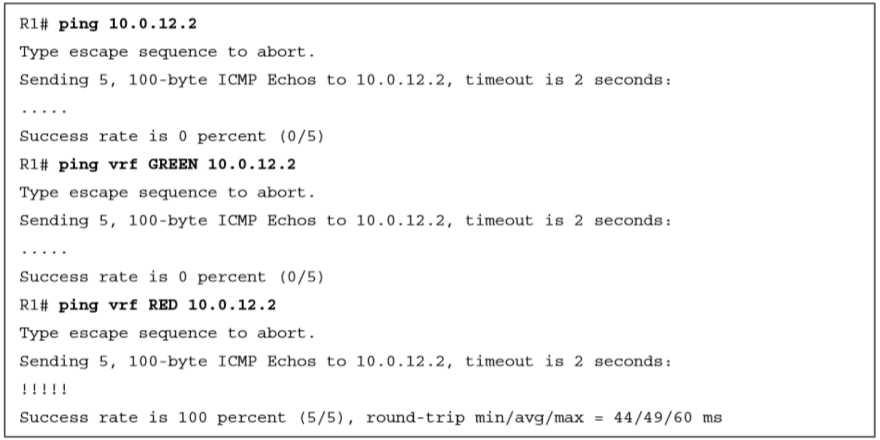
Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

To verify connectivity when using VRF instances, you must specify the VRF instance with the **ping** command. If you do not, the global routing table is used instead of the VRF routing table.

Example 18-14 shows a series of pings from R1 to R2. The first ping, with a destination of 10.0.12.2, fails because the VRF instance was not specified; therefore, the global routing table is being used. The second ping specifies the GREEN VRF but is using an IP address in the RED VRF; therefore, the ping fails. The last ping uses the correct VRF (RED) and an IP address in the RED VRF (10.0.12.2); therefore, the ping is successful. This is a great example of how VRF instances provide isolation.

Example 18-14 *Verifying Connecting with the ping Command*



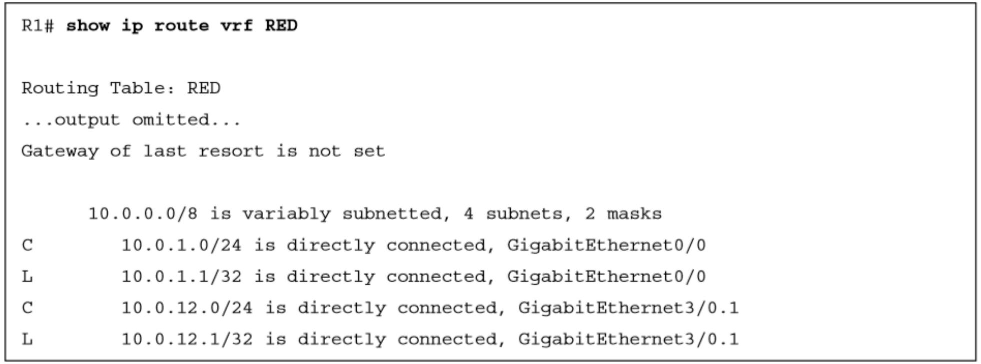
```
R1# ping 10.0.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.12.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1# ping vrf GREEN 10.0.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.12.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
R1# ping vrf RED 10.0.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/49/60 ms
```

Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

Example 18-15 shows the output of the RED VRF routing table. At this point, you have only directly connected and local routes. For all the routers to learn about all the other networks, you can use static or dynamic routing. The following examples use EIGRP as the dynamic routing protocol to provide full connectivity for each of the VRF instances.

Example 18-15 *Using the show ip route vrf RED Command to Verify the Contents of the RED VRF Routing Table*



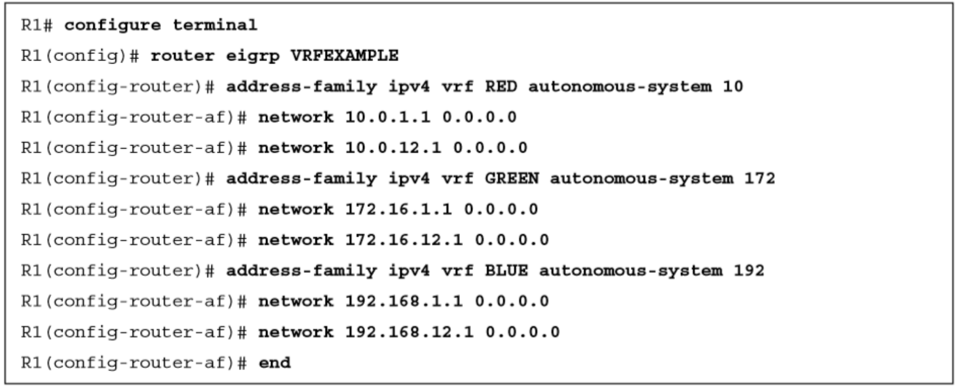
```
R1# show ip route vrf RED

Routing Table: RED
...output omitted...
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.1.0/24 is directly connected, GigabitEthernet0/0
L        10.0.1.1/32 is directly connected, GigabitEthernet0/0
C        10.0.12.0/24 is directly connected, GigabitEthernet3/0.1
L        10.0.12.1/32 is directly connected, GigabitEthernet3/0.1
```

Creating and Verifying VRF Instances (Cont.)

To configure EIGRP for multiple VRF instances, you use EIGRP named configuration mode because it permits you to create multiple address families, as shown in Example 18-16.

- Enter the EIGRP named configuration mode by using the **router eigrp** *name* command in global configuration mode.
- Next, create an address family for each of the VRF instances. You accomplish this with the **address-family ipv4 vrf** *vrf-name* **autonomous-system** *as-number* command.
- Then specify any EIGRP configuration commands that are needed for your scenario. In this case, you are enabling the routing process on only certain interfaces.

Example 18-16 *Configuring EIGRP for Multiple VRF Instances*



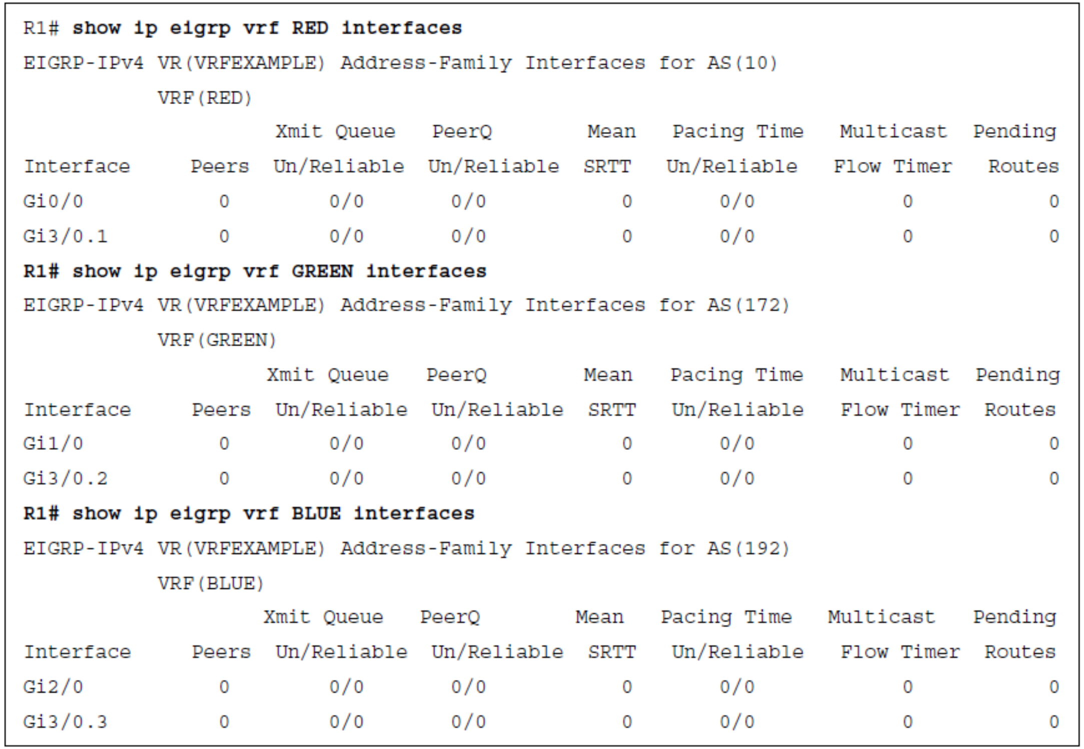
```
R1# configure terminal
R1(config)# router eigrp VRFEXAMPLE
R1(config-router)# address-family ipv4 vrf RED autonomous-system 10
R1(config-router-af)# network 10.0.1.1 0.0.0.0
R1(config-router-af)# network 10.0.12.1 0.0.0.0
R1(config-router)# address-family ipv4 vrf GREEN autonomous-system 172
R1(config-router-af)# network 172.16.1.1 0.0.0.0
R1(config-router-af)# network 172.16.12.1 0.0.0.0
R1(config-router)# address-family ipv4 vrf BLUE autonomous-system 192
R1(config-router-af)# network 192.168.1.1 0.0.0.0
R1(config-router-af)# network 192.168.12.1 0.0.0.0
R1(config-router-af)# end
```

Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

To verify that the interfaces are participating in the EIGRP process for the correct VRF instance, use the **show ip eigrp vrf** *vrf-name* **interfaces** command.

Example 18-17 *Verifying That the Interfaces Are Participating in the EIGRP Process for Each VRF*



```
R1# show ip eigrp vrf RED interfaces
EIGRP-IPv4 VR(VRFEXAMPLE) Address-Family Interfaces for AS(10)
           VRF(RED)
                    Xmit Queue   PeerQ        Mean   Pacing Time   Multicast   Pending
Interface      Peers  Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi0/0            0        0/0         0/0         0        0/0           0           0
Gi3/0.1          0        0/0         0/0         0        0/0           0           0
R1# show ip eigrp vrf GREEN interfaces
EIGRP-IPv4 VR(VRFEXAMPLE) Address-Family Interfaces for AS(172)
           VRF(GREEN)
                    Xmit Queue   PeerQ        Mean   Pacing Time   Multicast   Pending
Interface      Peers  Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi1/0            0        0/0         0/0         0        0/0           0           0
Gi3/0.2          0        0/0         0/0         0        0/0           0           0
R1# show ip eigrp vrf BLUE interfaces
EIGRP-IPv4 VR(VRFEXAMPLE) Address-Family Interfaces for AS(192)
           VRF(BLUE)
                    Xmit Queue   PeerQ        Mean   Pacing Time   Multicast   Pending
Interface      Peers  Un/Reliable  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Gi2/0            0        0/0         0/0         0        0/0           0           0
Gi3/0.3          0        0/0         0/0         0        0/0           0           0
```

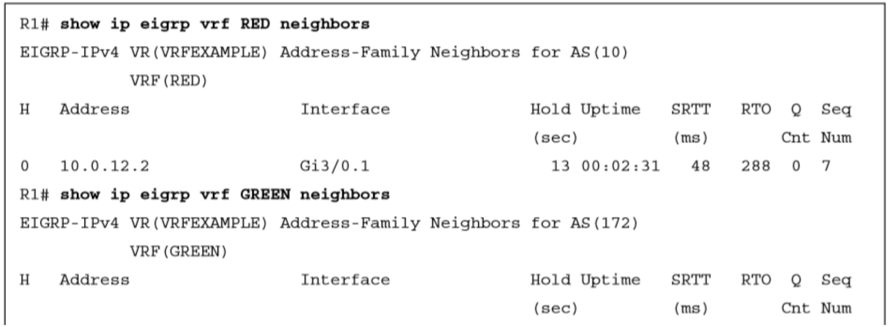
Implementing and Verifying VRF-Lite

Creating and Verifying VRF Instances (Cont.)

When all the other routers have been configured for EIGRP, you can verify neighbor adjacencies by using the **show ip eigrp vrf** *vrf-name neighbors* command. As before, because you are dealing with multiple VRF instances, you will notice that each **show** command in Example 18-18 displays only the neighbors that are within that VRF instance.

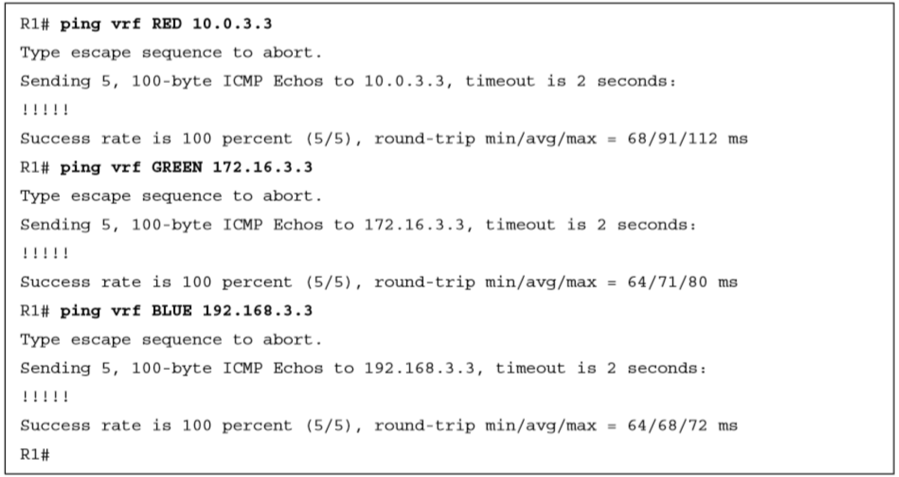
By using the **ping vrf** *vrf-name ipv4-address* command, as shown in Example 18-20, you can verify that connectivity exists from R1 to R3.

Example 18-18 *Verifying EIGRP Neighbors for Each VRF Instance with the show ip eigrp vrf vrf-name neighbors Command*



```
R1# show ip eigrp vrf RED neighbors
EIGRP-IPv4 VR(VRFEXAMPLE) Address-Family Neighbors for AS(10)
           VRF(RED)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   10.0.12.2               Gi3/0.1                  13 00:02:31  48    288  0  7
R1# show ip eigrp vrf GREEN neighbors
EIGRP-IPv4 VR(VRFEXAMPLE) Address-Family Neighbors for AS(172)
           VRF(GREEN)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
```

Example 18-20 *Verifying VRF Connectivity from R1 to R3*



```
R1# ping vrf RED 10.0.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/91/112 ms
R1# ping vrf GREEN 172.16.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/71/80 ms
R1# ping vrf BLUE 192.168.3.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/68/72 ms
R1#
```

An Introduction to MPLS Operations

- Multiprotocol Label Switching (MPLS) is a packet-forwarding method that makes forwarding decisions based on labels instead of on the Layer 3 destination of the packet.
- MPLS was designed to support many different Layer 3 protocols, but in this section we focus on IP only.
- MPLS is not much faster than traditional IP routing.
- MPLS decreases forwarding overhead on core routers, making them more efficient.
- MPLS can forward other Layer 3 protocols besides IPv4, and MPLS supports multiple services, such as unicast routing, multicast routing, VPNs, Traffic Engineering (TE), QoS, and Any Transport Over MPLS (AToM). Therefore, MPLS is very efficient and flexible.

MPLS LIB and LFIB

In Figure 18-2, notice that the control plane of the MPLS-enabled router is responsible for exchanging labels with other MPLS-enabled routers, using a label distribution protocol in addition to exchanging routing information using routing protocols to populate the IP routing table (RIB).

After labels have been exchanged, the label information is used to populate the LIB, and then the best label information can be used to populate the Forwarding Information Base (FIB) so unlabeled packets can be labeled and the LFIB labeled packets can be forwarded, or labels can be removed when packets need to be forwarded by the FIB.

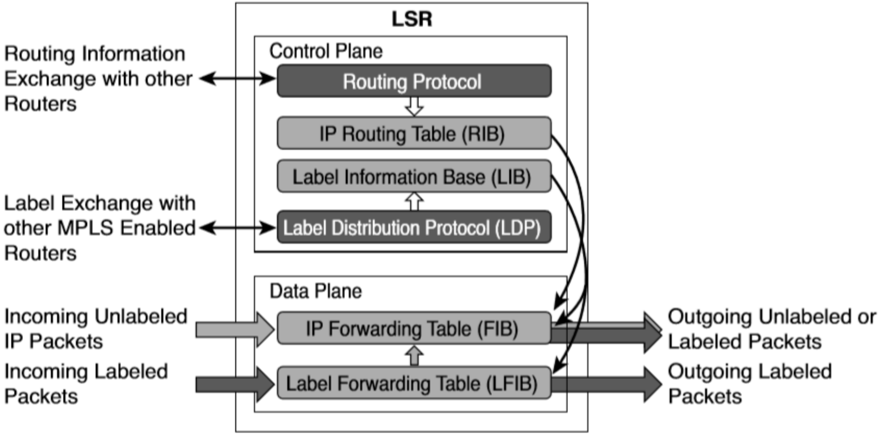

LSR
Control Plane
Routing Protocol
IP Routing Table (RIB)
Label Information Base (LIB)
Label Distribution Protocol (LDP)
Routing Information Exchange with other Routers
Label Exchange with other MPLS Enabled Routers
Data Plane
IP Forwarding Table (FIB)
Label Forwarding Table (LFIB)
Incoming Unlabeled IP Packets
Incoming Labeled Packets
Outgoing Unlabeled or Labeled Packets
Outgoing Labeled Packets

Figure 18-2 *Control Plane and Data Plane of an LSR*

Label Switching Routers

Examine Figure 18-3. Routers R1 through R5 are part of the MPLS domain. They are known as label switching routers (LSRs) because they support MPLS. They understand MPLS labels and can receive and transmit labeled packets on their interfaces.

In this case, R1 and R5 are considered edge LSRs, and R2, R3, and R4 are considered intermediate LSRs. An edge LSR sits at the edge of the MPLS domain and adds labels to packets that are entering the MPLS domain (known as an ingress LSR), removes labels from packets that will be leaving the MPLS domain (known as an egress LSR), and even forwards packets as needed based on labels or the lack of a label. An intermediate LSR sits within the MPLS domain and primarily forwards packets using label information.

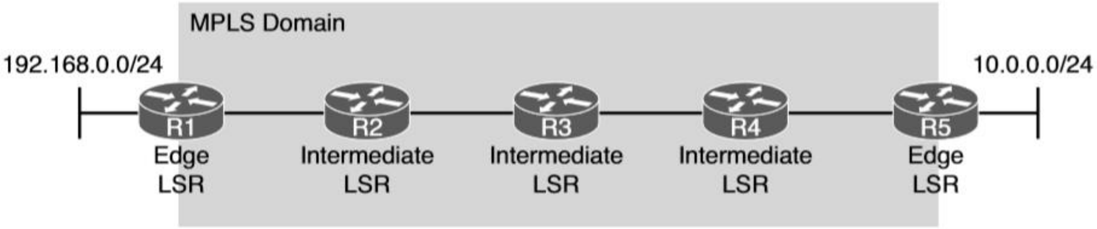
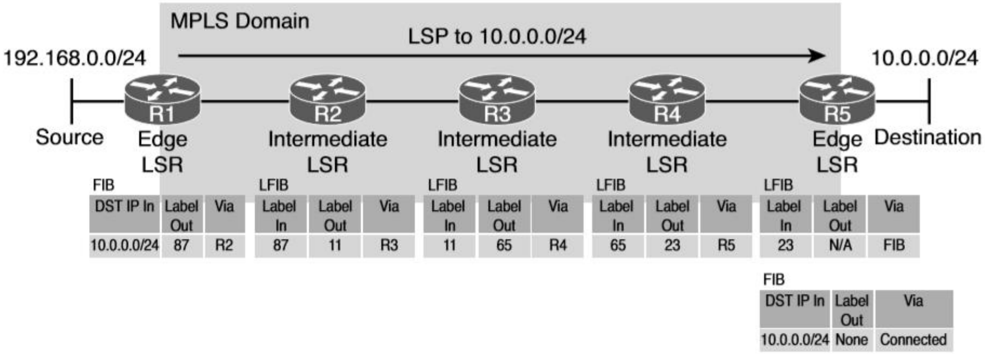


Figure 18-3 *Label Switching Routers in an MPLS Domain*

An Introduction to MPLS Operations
Label-Switched Path

The label-switched path (LSP) is the cumulative labeled path (sequence of routers) that a labeled packet takes through the MPLS domain.

- It is a unidirectional path, as shown in Figure 18-4; therefore, in a complex network with multiple potential paths between source and destination, it is possible that the LSP from source to destination could be different from the LSP that is used for the return traffic.
- Typically the same path in reverse is used for the return traffic because of the underlying dynamic routing protocols, such as OSPF and EIGRP, that are used to build the symmetrical network and its forwarding paths.
- In this case, the LSP from R1 to 10.0.0.0/24 uses labels 87, 11, 65, and 23. Along the path, each router examines the label to make a forwarding decision, removes the label, adds a new label if required, and then forwards the packet.



FIB

DST IP In	Label Out	Via
10.0.0.0/24	87	R2

LFIB

Label In	Label Out	Via
87	11	R3

LFIB

Label In	Label Out	Via
11	65	R4

LFIB

Label In	Label Out	Via
65	23	R5

LFIB

Label In	Label Out	Via
23	N/A	FIB

FIB

DST IP In	Label Out	Via
10.0.0.0/24	None	Connected

Figure 18-4 *The Label-Switched Path in an MPLS Domain*

An Introduction to MPLS Operations

Labels

For MPLS to work, a label needs to be added to the packet. The label is added as a shim header between the Layer 2 frame header and the Layer 3 packet header. Figure 18-5 shows the placement of the MPLS label shim header. The label is 4 bytes (32 bits) in size and contains four different fields, as shown in Figure 18-6.

MPLS-enabled routers automatically assign labels to every network that they know about. How does a router know about a network? It can be locally configured by configuring an IP address on a router interface and issuing the **no shutdown** command on the interface, or through the propagation of routing information with dynamic routing protocols such as OSPF and EIGRP.

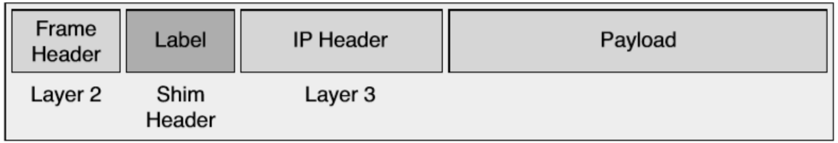


Figure 18-5 *Placement of the MPLS Label*

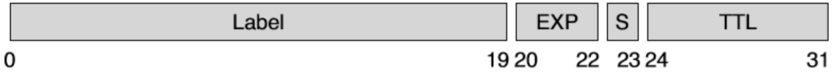
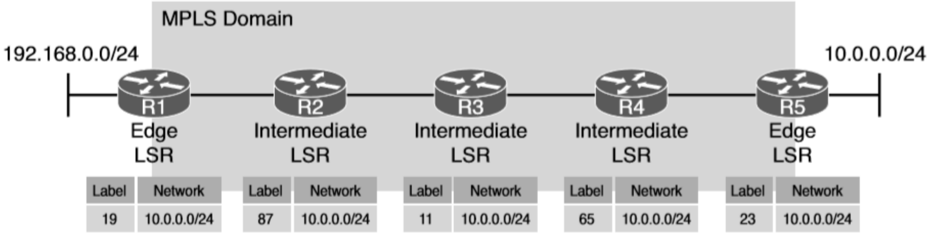


Figure 18-6 *Format of the MPLS Label*



R1:

Label	Network
19	10.0.0.0/24

R2:

Label	Network
87	10.0.0.0/24

R3:

Label	Network
11	10.0.0.0/24

R4:

Label	Network
65	10.0.0.0/24

R5:

Label	Network
23	10.0.0.0/24

Figure 18-7 *Routers Associating a Label with the 10.0.0.0/24 Network*

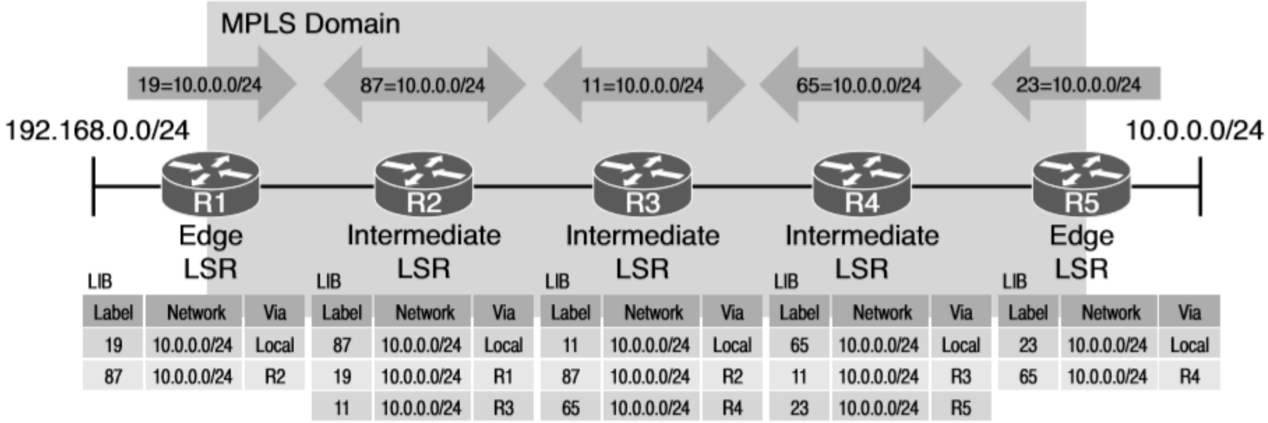
Label Distribution Protocol

- In order to build the LSP, labels need to be shared/distributed with directly connected LSRs. This is done using a label distribution protocol such as Label Distribution Protocol (LDP), which is the most common protocol in use when sharing/distributing labels for IPv4 prefixes.
- When MPLS has been enabled on an interface, LDP hello packets are sent out the interface to the destination multicast address 224.0.0.2 (the all routers multicast address), using UDP port 646. Any device on that same link that is also enabled for MPLS and that receives the hello packet forms an LDP TCP session using port 646 with the neighboring device so that label information can be exchanged.
- Within the hello packet is an LDP ID that is used to uniquely identify the neighbor and the label space, which will either be per platform (same label used out all interfaces for a single destination) or per interface (different label used out each interface for a single network).
- When establishing the LDP TCP session between two LSRs, one of the routers needs to be the active router. The active router is responsible for setting up the TCP session. The router with the higher LDP ID is selected as the active router and sets up the TCP session between the two routers.

An Introduction to MPLS Operations

Label Distribution Protocol (Cont.)

Figure 18-8 shows how R1 distributes its label of 19 for network 10.0.0.0/24 out all MPLS-enabled interfaces. R2 distributes its label of 87 for network 10.0.0.0/24 out all MPLS-enabled interfaces. R3 distributes its label of 11 for network 10.0.0.0/24 out all MPLS-enabled interfaces. R4 distributes its label of 65 for network 10.0.0.0/24 out all MPLS-enabled interfaces. And R5 distributes its label of 23 for network 10.0.0.0/24 out all MPLS-enabled interfaces after the TCP sessions have been established.



LIB (R1)

Label	Network	Via
19	10.0.0.0/24	Local
87	10.0.0.0/24	R2

LIB (R2)

Label	Network	Via
87	10.0.0.0/24	Local
19	10.0.0.0/24	R1
11	10.0.0.0/24	R3

LIB (R3)

Label	Network	Via
11	10.0.0.0/24	Local
87	10.0.0.0/24	R2
65	10.0.0.0/24	R4

LIB (R4)

Label	Network	Via
65	10.0.0.0/24	Local
11	10.0.0.0/24	R3
23	10.0.0.0/24	R5

LIB (R5)

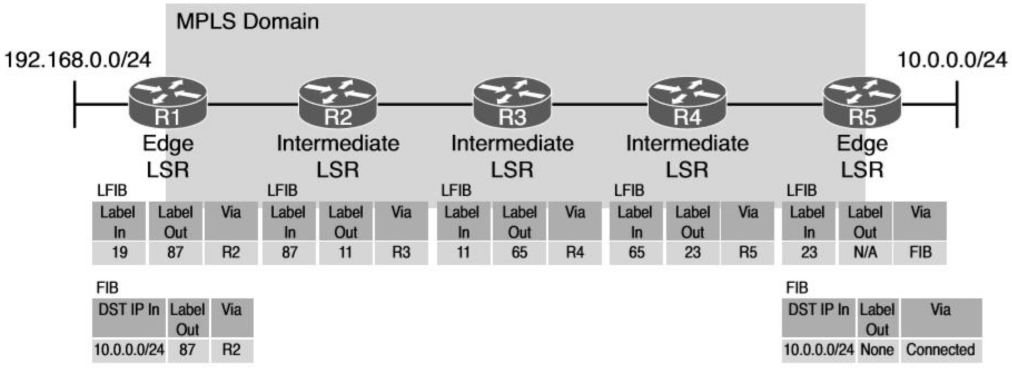
Label	Network	Via
23	10.0.0.0/24	Local
65	10.0.0.0/24	R4

Figure 18-8 *LSRs Using LDP to Distribute Labels Out All MPLS-Enabled Interfaces*

An Introduction to MPLS Operations

Penultimate Hop Popping

Review Figure 18-9 and pay close attention to R5. R5 must do two lookups when it receives a labeled packet destined to 10.0.0.0/24. First, it must look in the LFIB because it received a labeled frame. In this case, there is no label out. Therefore, it must remove the label and make a forwarding decision based on a second lookup, using the FIB to forward the packet. This is not efficient. The solution to this inefficiency is penultimate hop popping (PHP). With PHP, R4 pops the label before sending the packet to R5. IP address in network 10.0.0.0/24. In Figure 18-10, R5 has advertised the label "pop" to R4, and R4 has populated its LFIB accordingly.



R1 LFIB

Label In	Label Out	Via
19	87	R2

R2 LFIB

Label In	Label Out	Via
87	11	R3

R3 LFIB

Label In	Label Out	Via
11	65	R4

R4 LFIB

Label In	Label Out	Via
65	23	R5

R5 LFIB

Label In	Label Out	Via
23	N/A	FIB

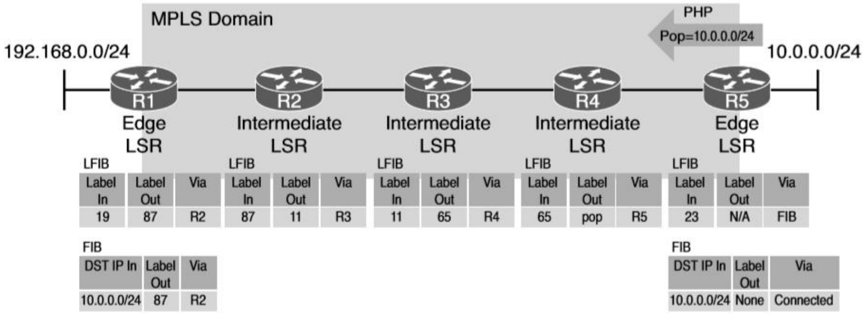
R1 FIB

DST IP In	Label Out	Via
10.0.0.0/24	87	R2

R5 FIB

DST IP In	Label Out	Via
10.0.0.0/24	None	Connected

Figure 18-9 *Populated FIBs and LFIBs*



R1 LFIB

Label In	Label Out	Via
19	87	R2

R2 LFIB

Label In	Label Out	Via
87	11	R3

R3 LFIB

Label In	Label Out	Via
11	65	R4

R4 LFIB

Label In	Label Out	Via
65	pop	R5

R5 LFIB

Label In	Label Out	Via
23	N/A	FIB

R1 FIB

DST IP In	Label Out	Via
10.0.0.0/24	87	R2

R5 FIB

DST IP In	Label Out	Via
10.0.0.0/24	None	Connected

Figure 18-10 *PHP: R5 Indicating to R4 to Pop the Label*

An Introduction to MPLS Layer 3 VPNs

- MPLS Layer 3 VPNs provide peer-to-peer connectivity between private customer sites across a shared network.
- The MPLS domain is referred to as the P-network, and the customer sites are referred to as the C-network.
- This section explores how you can use MPLS Layer 3 VPNs to connect your private networks over your provider’s public network.

An Introduction to MPLS Layer 3 VPNs

MPLS Domain Connecting Customer Sites Together Example

Figure 18-11 shows a provider (such as an ISP) MPLS domain, with Customer A and Customer B both using the same MPLS domain to connect their own private sites together.

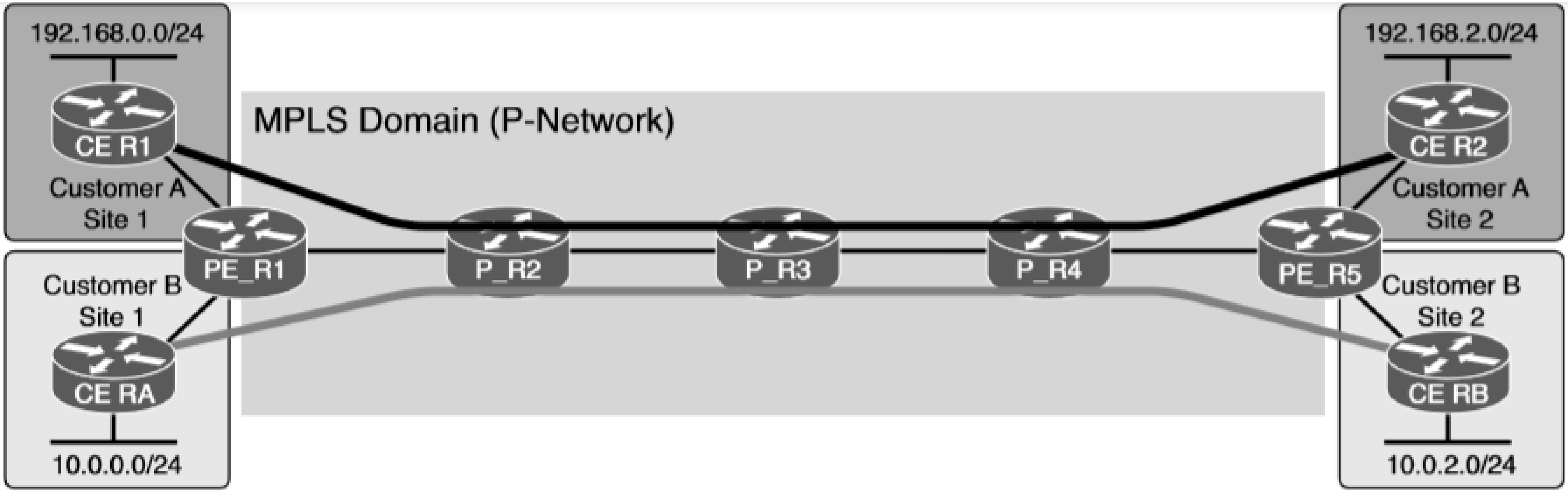


Figure 18-11 *A Sample MPLS Domain Connecting Customer Sites Together*

MPLS Domain Connecting Customer Sites Together Example (Cont.)

- The CE (customer edge) routers connect to the PE (provider edge) routers of the MPLS domain. The PE routers, PE_R1 and PE_R5 in Figure 18-11, are the ingress and egress LSRs for the MPLS domain.
- The P (provider) routers, such as P_R2, P_R3, and P_R4 in Figure 18-11, are the intermediate LSRs of the MPLS domain.
- The goal is to have Customer A Site 1 and Customer A Site 2 exchange their local routing information over the MPLS domain and then forward traffic as needed from Site 1 and Site 2 over the MPLS domain. The same would be true for Customer B Site 1 and Customer B Site 2.
- Due to the nature of the MPLS Layer 3 VPN, overlapping address spaces between customers is of no concern. Therefore, Customer A and Customer B can be using the same private IP address space.

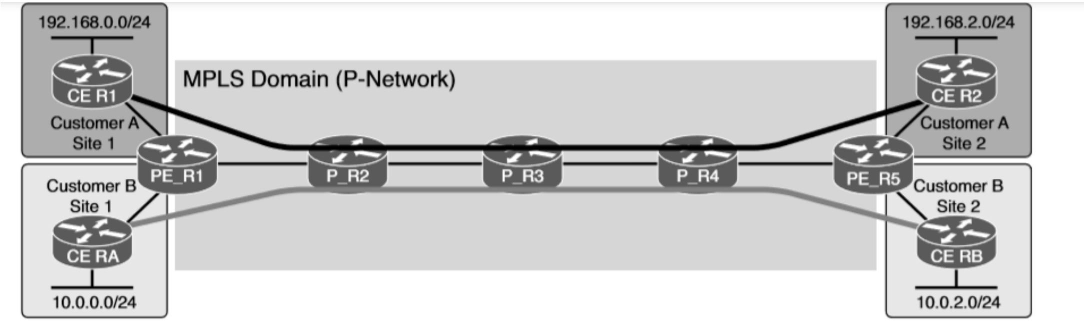


Figure 18-11 *A Sample MPLS Domain Connecting Customer Sites Together*

MPLS Domain Connecting Customer Sites Together Example (Cont.)

- In order to support multiple customers, the PE routers need to use VRF instances, as shown in Figure 18-12, to isolate customer information and traffic from other customers.
- A different VRF instance needs to be created for each customer, and the interface that connects to the customer's CE router needs to be associated with the VRF.
- The CE router and the PE router exchange IPv4 routes using a routing protocol such as RIP (Routing Information Protocol), EIGRP (Enhanced Interior Gateway Routing Protocol), OSPF (Open Shortest Path First), or BGP (Border Gateway Protocol), and the routes are placed in the customer-specific VRF table on the PE router.
- From the customer's perspective, the PE router is simply another router in the customer's network, but it is under the control of the provider. However, note that all P routers are hidden from the customer.

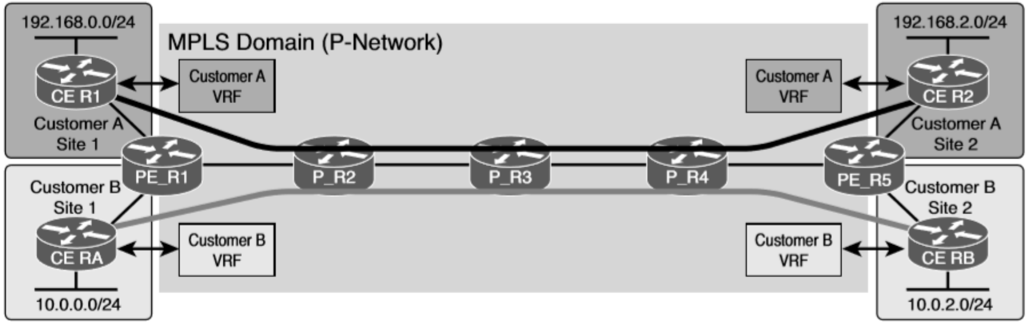


Figure 18-12 *Isolating Customer Routing Information Using VRF Instances*

MPLS Domain Connecting Customer Sites Together Example (Cont.)

- After the PE routers learn routes from the CE routers, the PE routers redistribute the routes into Multiprotocol-Interior Border Gateway Protocol (MP-BGP) so they can be exchanged with other PE routers.
- When another PE router receives the routes, they are redistributed into an IGP and placed in the correct customer VRF instance so they can be exchanged with the CE router, as shown in Figure 18-13.
- The P routers are not participating in BGP. Only the PE routers are. They are forming an MP-IBGP neighbor relationship with each other and exchanging the routes using the underlying network that is built with an Interior Gateway Protocol (IGP) such as OSPF or Intermediate System-to-Intermediate System (IS-IS).
- The PE routers and the P routers are using a dynamic routing protocol to learn about all the destinations in the P network, and only the PE routers are using MP-IBGP on top of that to exchange the customer routes.

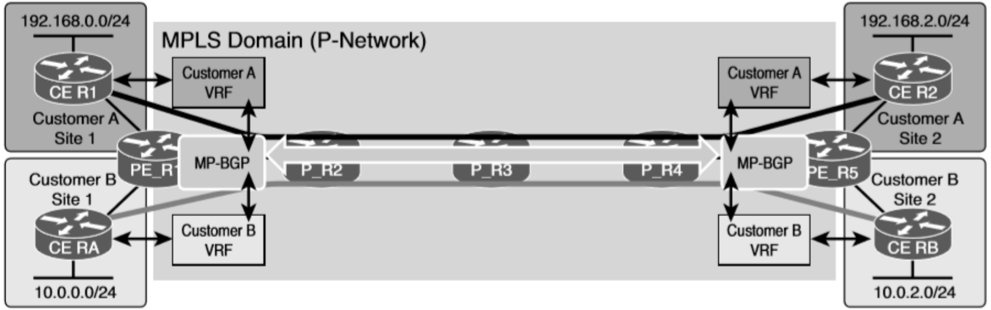


Figure 18-13 *PE Routers Redistributing Customer Information into MP-BGP*

An Introduction to MPLS Layer 3 VPNs

MPLS Layer 3 VPNv4 Address

- If all customer routes are being redistributed into MP-BGP, BGP handles identical network prefixes that belong to different customers by using a route distinguisher (RD) to expand the customer’s IP prefix so that it includes a unique value that distinguishes it from the other identical prefixes.
- The RD is generated and used by the PE routers on a per-customer VRF instance basis. The RD is used regardless of whether there are overlapping address spaces.
- The unique 64-bit RD is prepended to the 32-bit customer prefix (IPv4 route) to create a 96-bit unique prefix called a VPNv4 address, as shown in Figure 18-14. This VPNv4 address is exchanged by the MP-IBGP neighboring routers.

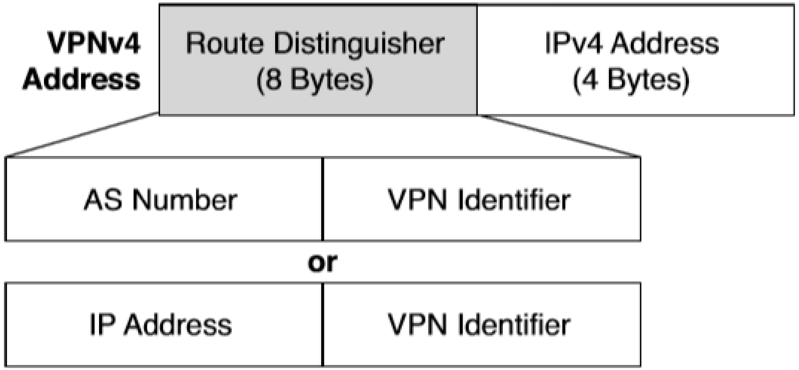


Figure 18-14 *Format of a VPNv4 Address*

An Introduction to MPLS Layer 3 VPNs

PE Routers Exchanging VPNv4 Routes

The following steps occur:

Step 1. The CE router and PE router exchange routes using a dynamic routing protocol such as OSPF or EIGRP.

Step 2. The PE router places the customer-specific routes in the customer-specific VRF table.

Step 3. The routes in the customer's VRF table are redistributed into MP-BGP as VPNv4 routes.

Step 4. The PE routers exchange VPNv4 routes over their MP-IBGP peering.

Step 5. The PE router redistributes the VPNv4 routes as OSPF, EIGRP, and other types of routes into the customer-specific VRF table.

Step 6. The PE router and CE router exchange routes using a dynamic routing protocol such as OSPF or EIGRP.

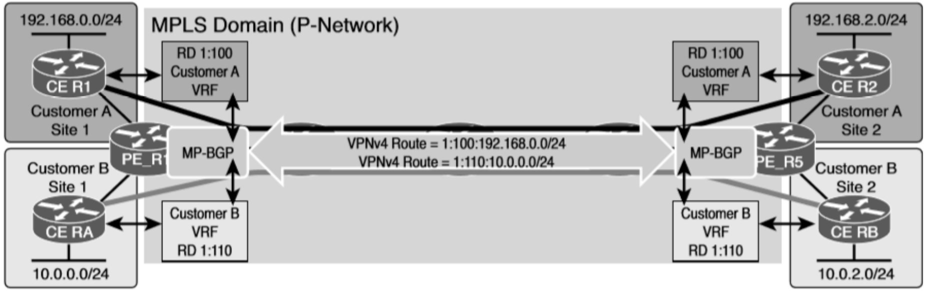


Figure 18-15 *PE Routers Exchanging VPNv4 Routes*

MPLS Layer 3 VPN Label Stack

For the MPLS domain to forward traffic, a label stack is required. Specifically, two labels are required for traffic to be successfully forwarded through the MPLS domain. The first label that is attached to the packet is a VPN label, and the second label that is attached is the LDP label, as shown in Figure 18-16.

- When the IP packet arrives at the ingress PE router, the PE router attaches both labels.
- The egress router uses the VPN label to determine customer specifics about the packet and what should be done with it.
- The LDP label is used for label switching from PE to PE in the MPLS domain.
- VPN labels are learned from PE routers over the MP-IBGP peering, and the LDP labels are learned using the methods we explored in the “An Introduction to MPLS Operations” section of this chapter.

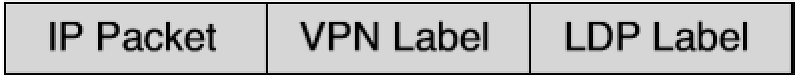
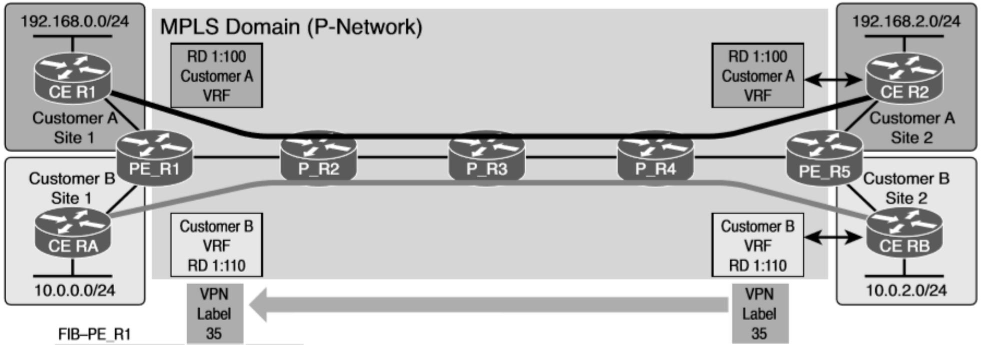


Figure 18-16 *Sample Label Stack for MPLS L3 VPNs*

VPN Label Assignment

When PE_R5 learns of 10.0.2.0/24 from CE_RB, it places it in the Customer B VRF instance. It then redistributes it into MP-BGP and thus creates a VPNv4 route of 1:110:10.0.2.0/24. This VPNv4 route needs a VPN label created for it so that forwarding will be successful. In this case, PE_R5 assigns it the label 35. This label is shared with PE_R1 over the MP-IBGP peering they have. Any time PE_R1 receives an IP packet that is destined for 10.0.2.0/24, it knows to attach the label 35 so the packet can be forwarded. However, this label is known only by the PE routers. Therefore, if PE_R1 forwards this VPN packet to P_R2, it will be dropped because it has no idea what the VPN label 35 means. Therefore, the LDP label is needed to forward the packet from PE_R1 and PE_R5.



FIB–PE_R1

DST IP In	Label Out	Via
1:110:10.0.2.0/24	35	PE_R5

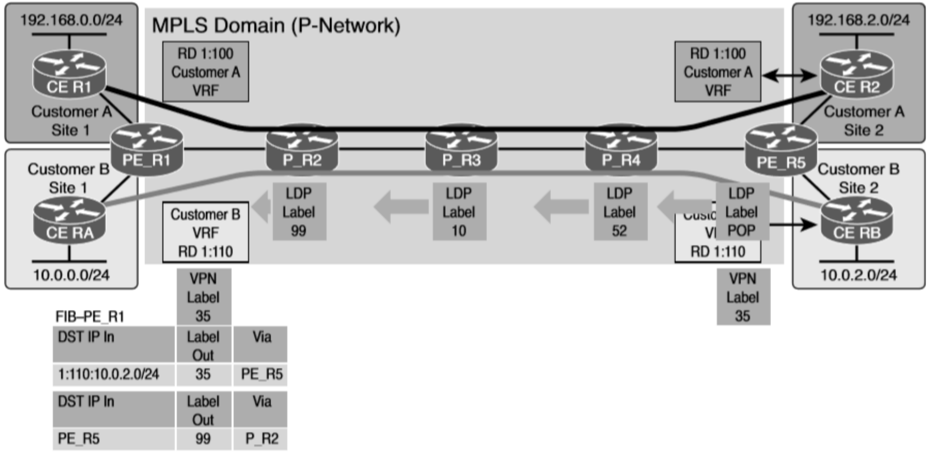
Figure 18-17 *VPN Label Assigned by PE_R5 and Shared with PE_R1*

An Introduction to MPLS Layer 3 VPNs

Label Switched Path

Figure 18-18 shows how LDP is used to exchange labels that have been generated by the PE routers (ingress and egress LSRs) and the P routers (intermediate LSRs). PE_R5 tells P_R4 to pop the label. P_R4 tells P_R3 to use the label 52. P_R3 tells P_R2 to use the label 10. P_R2 tells PE_R1 to use the label 99.

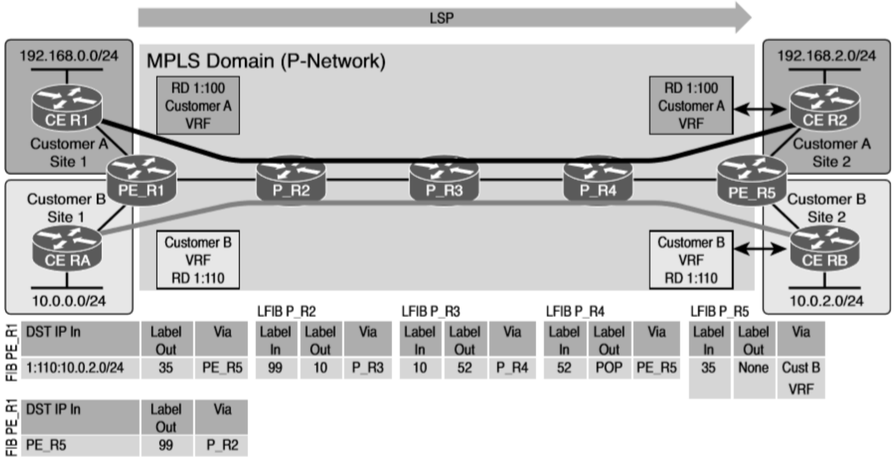
In Figure 18-19, the complete LSP is now ready to label switch the VPN packet from PE_R1 to PE_R5.



DST IP In	Label Out	Via
1:110:10.0.2.0/24	35	PE_R5

DST IP In	Label Out	Via
PE_R5	99	P_R2

Figure 18-18 *LDP Label Assignment*



FIB PE_R1

DST IP In	Label Out	Via
1:110:10.0.2.0/24	35	PE_R5

LFIB P_R2

Label In	Label Out	Via
99	10	P_R3

LFIB P_R3

Label In	Label Out	Via
10	52	P_R4

LFIB P_R4

Label In	Label Out	Via
52	POP	PE_R5

LFIB P_R5

Label In	Label Out	Via
35	None	Cust B VRF

FIB PE_R1

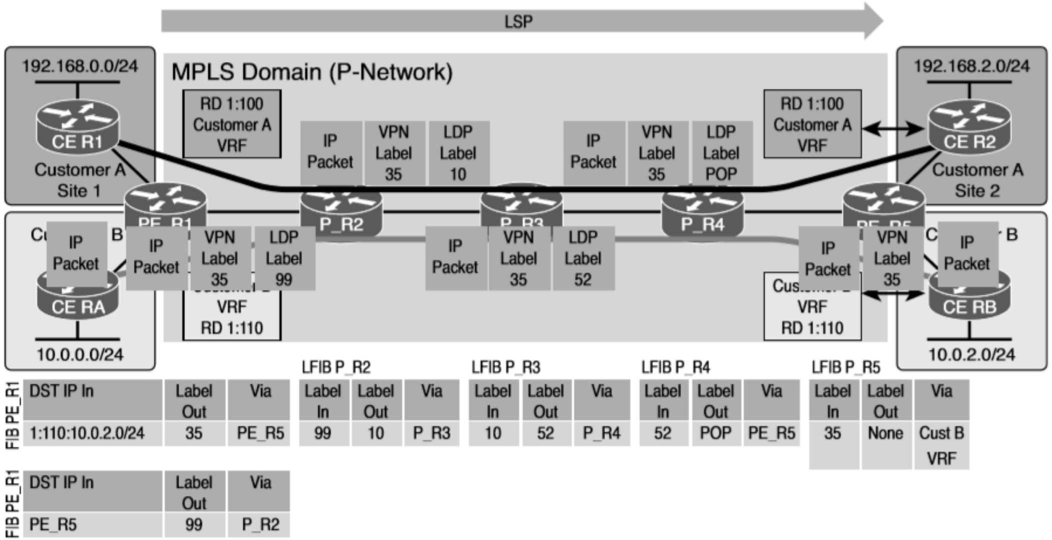
DST IP In	Label Out	Via
PE_R5	99	P_R2

Figure 18-19 *MPLS L3 VPN Label-Switched Path*

An Introduction to MPLS Layer 3 VPNs

Label Switched Path (Cont.)

When an IP packet destined to 10.0.2.0/24 arrives at PE_R1 from CE_RA, PE_R1 determines that the packet needs a VPN label of 35. Now, PE_R5 will know what to do with the VPN packet and an LDP label of 99 so that the VPN packet can be label switched through the MPLS domain. When the label stack is complete, PE_R1 sends the labelstacked packet to P_R2. When P_R2 receives it, it only examines the LDP label. Based on the LFIB, it states that the label 99 needs to be swapped to 10 and forwarded to P_R3. When P_R3 receives it, it only examines the LDP label. Based on the LFIB, it states that the label 10 needs to be swapped to 52 and forwarded to P_R4. When P_R4 receives it, it only examines the LDP label. Based on the LFIB, it states that the label 52 needs to be popped and forwarded to P_R5. Now PE_R5 only needs to read the VPN label, which is 35. Based on the scenario, the label is removed, and the VRF instance for Customer B is used to forward the IP packet to the CE_RB router.



FIB PE_R1

DST IP In	Label Out	Via
1:110:10.0.2.0/24	35	PE_R5

LFIB P_R2

Label In	Label Out	Via
99	10	P_R3

LFIB P_R3

Label In	Label Out	Via
10	52	P_R4

LFIB P_R4

Label In	Label Out	Via
52	POP	PE_R5

LFIB P_R5

Label In	Label Out	Via
35	None	Cust B VRF

FIB PE_R1

DST IP In	Label Out	Via
PE_R5	99	P_R2

Figure 18-20 *Forwarding Through the MPLS L3 VPN Domain*

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 18

Description
The purpose of VRF
Configuring VRF on R1 with the **ip vrf** command
Assigning interfaces to the VRF instances with the **ip vrf forwarding** command
Creating subinterfaces on R1 and assigning them to the correct VRF instances
Verifying the interface IP address, VRF, and protocol configurations
Verifying the VRF routing tables

Prepare for the Exam

Key Topics for Chapter 18 (Cont.)

Description	
MPLS LIB and LFIB	The benefit of PHP
Label switching routers	MPLS Layer 3 VPNs
Label-switched path	The Layer 3 VPNv4 address and route distinguishers
Labels	How routes are learned by CE routers over an MPLS Layer 3 VPN
Label Distribution Protocol	MPLS Layer 3 VPN label stack

Prepare for the Exam

Key Terms for Chapter 18

Term		
VRF	intermediate LSR	MPLS Layer 3 VPN
VRF-Lite	egress LSR	P-network
MPLS	ingress LSR	C-network
LIB	LSP	CE router
LFIB	LDP	PE router
RIB	Label	P router
FIB	LDP label	VPNv4 address
LSR	VPN label	label stack
edge LSR	PHP	

Prepare for the Exam

Command Reference for Chapter 18

Task	Command Syntax
Define a VRF instance and enter VRF configuration mode for the instance (in global configuration mode)	**ip vrf** *vrf-name*
Associate an interface or a subinterface with a VRF instance (in interface configuration mode)	**ip vrf forwarding** *vrf-name*
Display configured VRF instances and associated router interfaces	**show ip vrf**
Display the routes in the global routing table	**show ip route**
Display the routes in the routing table for the VRF specified in the command	**show ip route vrf** *vrf-name*
Display all VRF-enabled interfaces on the router, including their IP addresses and whether the protocol is up or down	**show ip vrf interfaces**
Test IP connectivity for a specific VRF	**ping vrf** *vrf-name ipv4-address*

cisco