360RSW05-WKB-CA10

Cisco 360 CCIE R&S Exercise Workbook Introduction

The Cisco 360 CCIE® R&S Exercise Workbook contains 20 challenging scenarios at the CCIE level that can be used for rigorous self-paced practice.

Each lab provides an extensive answer key, Mentor Guide support, and verification tables and is designed to maximize learning by providing practical experience. Also, self-paced learning resources such as the Cisco 360 CCIE R&S Reference Library and Cisco 360 CCIE R&S lessons supplement the Exercise Workbook scenarios.

Cisco 360 CCIE R&S Exercise Workbook Lab 10 Configuration Section Answer Key

Table of Contents

Answer Key Structure

Section One

The answer key PDF document is downloadable from the web portal.

Section Two

To obtain a comprehensive view of the configuration for a specific section, access the Mentor Guide engine in the web portal.

Exercise Workbook Lab 10 Configuration Section Answer Key

Note	Regardless of any configuration you perform in this lab, it is very important that you conform to the general guidelines that are provided in the "Restrictions and Goals" section. If you do not conform to the guidelines, you could have a significant deduction of points in your final score.

Grading and Duration

- Configuration lab duration: 6 hours
- Configuration lab maximum score: 76 points

Note	You can assess your progress on the self-paced labs in this workbook by adding up the points that are assigned to sections and tasks. Consider taking the full Assessment Labs to assess your readiness level.

Difficulty Level

- Difficulty: Intermediate

Restrictions and Goals

Note	Read this section carefully.

- To receive any credit for a subsection, you must fully complete the subsection as per requirements. You will *not* receive partial credit for partially completed subsections.
- IP version 4 (IPv4) subnets displayed in the scenario diagram belong to network 151.10.0.0/16.
- *Points will be deducted from multiple sections for failing to assign correct IPv4 addresses.*
- Do not use any static routes.
- Advertise loopback interfaces with their original masks for IPv4 and IPv6 protocols.
- Network 0.0.0.0/0 should not appear in any routing table (**show ip route**) except on R8.
- Do not use the **ip default-network** commands.
- All IP addresses involved in this scenario must be reachable, unless explicitly specified otherwise.

- Unless explicitly specified otherwise, addresses and networks that are advertised in the "Border Gateway Protocol" (BGP) section need to be reachable by all BGP routers but do not have to be reachable by interior gateway protocol (IGP)-only routers.
- Do not create new interfaces to fulfill IGP requirements, and do not create any summaries, unless the summary is required to meet explicitly stated scenario requirements.
- Do not introduce any new IPv4 or IPv6 addresses unless the instructions explicitly specify otherwise.
- Do not modify the hostname, console, or vty configuration unless you are specifically asked to do so.
- Do not modify the initial interface or IP address numbering.

Explanation of Each of the Restrictions and Goals

IPv4 subnets displayed in the scenario "IPv4 IGP" diagram belong to network 151.10.0.0/16.

All IP addresses in this lab belong to the 151.10.0.0/16 address space, with the exception of a set of prefixes that are used in the BGP section.

Do not use any static routes.

Static routes can be used to solve a range of reachability problems. However, you cannot use them in this lab. You must rely on skillful configuration of all your unicast routing protocols.

Advertise loopback interfaces with their original masks.

The original mask is the mask configured on the loopback interface. OSPF treats loopback interfaces as host routes by default and advertises them as /32 prefixes. The requirement to advertise loopback interfaces with their original masks precludes using the default OSPF network type for the loopback interface. You need to provide a solution, such as changing the OSPF network type or summarizations.

Network 0.0.0.0/0 should not appear in any routing table (show ip route**).**

A 0.0.0.0/0 entry can be used to solve a range of reachability problems. In particular, a 0.0.0.0/0 entry can be used to set up the gateway of last resort. In this exercise, you cannot use any 0.0.0.0/0 entries. Route summarization is an alternative to using the 0.0.0.0/0 route to solve the reachability problem.

Do not use the ip default-network **command.**

This command can be used to solve reachability issues by setting the gateway of last resort. This command generates 0.0.0.0/0 in the Routing Information Protocol (RIP) environment. You cannot use it in this scenario.

All IP addresses that are involved in this scenario must be reachable.

This goal is a key goal to observe. It requires that all your IGPs and your routing policy tasks be configured properly. The key elements of your routing policy include route redistribution and the controlling of routing updates using the **distribute-list**, **route-map**, and **distance** commands. A key point to remember about this lab is that the term "redistribution" is not explicitly used. However, you must perform redistribution to ensure that all IP addresses are reachable without the use of static routes or 0.0.0.0/0 routes.

Addresses and networks that are advertised in the BGP section need to be reachable by all BGP routers but do not have to be reachable by IGP-only routers.

This statement relaxes the requirement that all IP addresses must be reachable. The BGP prefixes need only be reachable among the routers specified in the BGP section. They can be used in other unicast tables. However, BGP routers need to have the prefixes in the routing tables and need to be able to forward traffic to the addresses known via BGP.

1. Switch Configuration

General Tasks:

As with any switch configuration, you must address the following basic configuration requirements: setting the VLAN Trunking Protocol (VTP) mode, configuring trunk ports, and statically assigning ports to VLANs. For a good reference on mastering basic Cisco Catalyst 3560 Switch configuration tasks, access the full set of Catalyst video-on-demand (VoD) sessions within the "Link Layer" lesson in the Cisco 360 learning portal. These self-paced sessions provide more than 7 hours of instruction on a range of basic Catalyst switch configuration tasks.

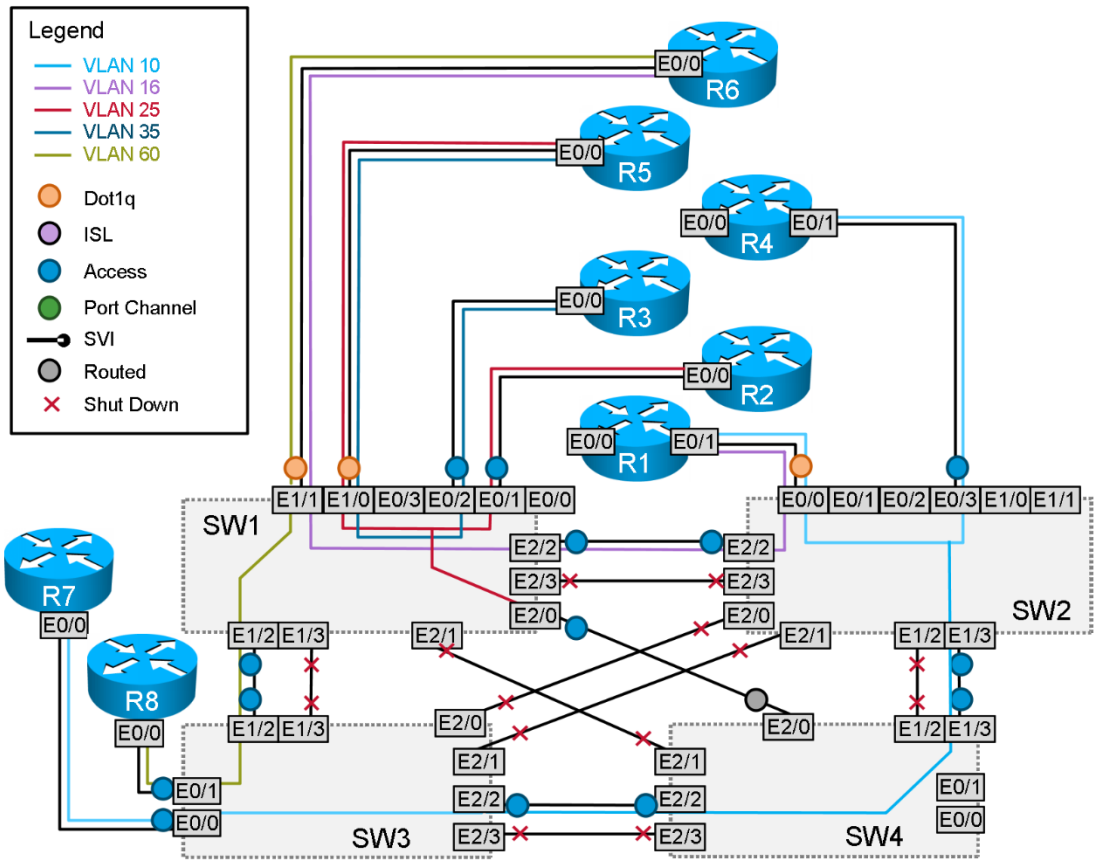
Note that not all Cisco Catalyst 3560 Switch configuration features are supported on the virtual Cisco IOS Software on UNIX.

Configure the VLANs and the VLAN names according to the scenario specifications and assign the ports of the switches to these VLANs. Make sure that the VLAN names are spelled correctly and match the letter case.

Use the "VLANs," "Switch-to-Router Connections," and "Switch-to-Switch Connections" tables to analyze the VLAN propagation in this lab.

See the following diagram for the VLAN layout.

VLAN Propagation



Carefully review the entire scenario. Closely examine the supplied diagram and any associated tables. Determine how you need to configure VTP, how to configure ports that are assigned as trunks, and how to configure ports that are assigned as static VLAN ports. Use the **switchport mode access** command to statically assign ports to a VLAN.

The fact that there is no VLAN associated with the link between SW1 and SW4 on SW4 suggests that interface 2/0 of SW4 should be configured as a routed port. Simply enter the command **no switchport** and configure the appropriate IPv6 address.

Both the VLANs table and the diagram indicate that you should have trunk links R6-SW1, R5-SW1, and R1-SW2. Minimize the VLAN traffic across trunk links using the command **switchport trunk allowed vlan**.

The scenario requires the use of access ports on switch-to-switch ports whenever possible. After you examine the switch-to-switch ports that are not specifically configured as routed ports, you can see four links:

- SW1 – SW2
- SW1 – SW3
- SW2 – SW4
- SW3 – SW4

The following links need to propagate exactly one VLAN:

— SW1 – SW2	VLAN 16	R1-R6
— SW2 – SW4	VLAN 10	R1-R4-R7
— SW3 – SW4	VLAN 10	R1-R4-R7
— SW1 – SW3	VLAN 60	R6-R8

SW4 is used as an IPv6-only router in this scenario.

Issue: Configure ports on switches SW1 and SW2 connected to routers R1, R2, R3, R4, R5, and R6 to change the spanning-tree state directly from blocking to forwarding.

Solution:

Switches support a special mode of Spanning Tree Protocol (STP) operation called PortFast, which shortcuts through normal spanning-tree states and transitions interfaces directly to a forwarding state. This mode is intended to be used on connections from switches to endstations that do not participate in STP (that is, workstations and routers).

To enable STP PortFast, issue the **spanning-tree portfast** command on access ports and the **spanning-tree portfast trunk** command on trunk ports.

Note that entering just the **spanning-tree portfast** command will not enable the PortFast feature for the ports in trunking mode.

A sample configuration for the access VLAN and trunk ports is provided for SW1 and SW2. Configure other active ports on SW1 and SW2 similarly:

SW1:

```
interface Ethernet0/1
 switchport access vlan 25
 switchport trunk encapsulation dot1q
 switchport mode access
 duplex auto
 spanning-tree portfast
```

SW2:

```
interface Ethernet0/0
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 10,16
 switchport mode trunk
 duplex auto
 spanning-tree portfast trunk
```

```
!                                               !
interface Ethernet0/2                           interface Ethernet0/3
 switchport access vlan 35                       switchport access vlan 10
 switchport mode access                          switchport mode access
 duplex auto                                     duplex auto
 spanning-tree portfast                          spanning-tree portfast
!                                               !
!
interface Ethernet1/0
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 25,35
 switchport mode trunk
 duplex auto
 spanning-tree portfast trunk
!
interface Ethernet1/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 16,60
 switchport mode trunk
 duplex auto
 spanning-tree portfast trunk
!
```

Verify the spanning tree PortFast interface configuration on SW1 and SW2:

```
SW1#show spanning-tree interface ethernet0/1 portfast
VLAN0025            enabled
SW1#show spanning-tree interface ethernet0/2 portfast
VLAN0035            enabled
SW1#show spanning-tree interface ethernete1/0 portfast
VLAN0025            enabled
VLAN0035            enabled
SW1#show spanning-tree interface ethernete1/1 portfast
VLAN0016            enabled
VLAN0060            enabled
SW1#

SW2#show spanning-tree interface ethernet0/0 portfast
VLAN0010            enabled
VLAN0016            enabled
SW2#show spanning-tree interface ethernet0/3 portfast
VLAN0010            enabled
SW2#
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

2. IP Addresses Configuration

Issue: Configure R6 to supply the IP address 151.10.16.1 to R1 via DHCP. Do not use any MAC address-based identification. Ensure that R6 only leases 151.10.16.1 to R1.

Solution:

Create a pool on R6 for the 151.10.16.0/24 subnet, and then exclude all but the one desired address.

R6:

```
no ip dhcp conflict logging
ip dhcp excluded-address 151.10.16.2 151.10.16.254
!
ip dhcp pool MYPOOL
   network 151.10.16.0 255.255.255.0
!
```

On the R1 interface e0/1.16, configure **ip address dhcp**.

R1 should receive address 151.10.16.1/24. On R6, you can use the command **debug ip dhcp server** *events*, to watch the process from the server end. You can also check the DHCP server statistics:

```
R1#show ip int e0/1.16 | i Internet address
  Internet address is 151.10.16.1/24

R1#show dhcp lease
Temp IP addr: 151.10.16.1  for peer on Interface: Ethernet0/1.16
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 151.10.16.6, state: 3 Bound
   DHCP transaction id: BD3
   Lease: 86400 secs,  Renewal: 43200 secs,  Rebind: 75600 secs
   Next timer fires after: 09:26:28
   Retry count: 0   Client-ID: cisco-0016.9d43.f7d2-Et0/1.16
   Client-ID hex dump: 636973636F2D303031362E396434332E
                       663764322D4574302F312E3136
   Hostname: R1


R6#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
151.10.16.1         0063.6973.636f.2d30.    Aug 27 2013 05:18 AM    Automatic
                    3031.362e.3964.3433.
                    2e66.3764.322d.4574.
                    302f.312e.3136
R6#
```

```
R6#sh ip dhcp server statistics
Memory usage         23105
Address pools        1
Database agents      0
Automatic bindings   1
Manual bindings      0
Expired bindings     0
Malformed messages   0
Secure arp entries   0

Message              Received
BOOTREQUEST          0
DHCPDISCOVER         1
DHCPREQUEST          1
DHCPDECLINE          0
DHCPRELEASE          0
DHCPINFORM           0

Message              Sent
BOOTREPLY            0
DHCPOFFER            1
DHCPACK              1
DHCPNAK              0
```

Issue: Configure a switch feature that will permit only R6 to be a DHCP server in VLAN 16.

Solution:

The switch DHCP Snooping feature can help protect your network against unauthorized DHCP servers. You enable the feature on the switch and VLAN, and then designate trusted ports; those ports that are connected to authorized servers. By default, all ports in the enabled VLAN are untrusted, and DHCP server packets received on them will be dropped.

Note that here, the DHCP server is connected to port 1/1 of SW1, and the only current client is connected to port 0/0 of SW2. Even though there are no potential clients currently connected to SW1, the requirement states that you should protect VLAN 16, not just the existing ports in that VLAN. So DHCP snooping protection has been configured on both switches.

Both ports 1/1 on SW1 and 2/2 on SW2 are configured as trusted ports. It was also necessary to configure the command **no ip dhcp snooping information option**. This command turns off the insertion of option 82 information into the DHCP packets. Option 82 information includes the switch MAC address and switch port, and the switches insert option 82 to help the server assign the proper addresses. In the case of this scenario, these packets were seen as malformed; they were dropped, by default, when they were received on SW1 from SW2, and they were reported as errors if they were received by the DHCP server.

If you enable the optional DHCP binding database, a number of additional security features become available through the IP Source Guard functions.

SW1:

```
ip dhcp snooping vlan 16
no ip dhcp snooping information option
ip dhcp snooping
!
interface Ethernet1/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 16,60
 switchport mode trunk
 duplex auto
 ip dhcp snooping trust
!
!
```


SW2:

```
ip dhcp snooping vlan 16
no ip dhcp snooping information option
ip dhcp snooping
!
interface Ethernet2/2
 switchport access vlan 16
 switchport mode access
 duplex auto
 ip dhcp snooping trust
!
```

Verify the DHCP snooping configuration and bindings on SW1:

```
 SW1#sh ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
16
DHCP snooping is operational on following VLANs:
16
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled
   circuit-id default format: vlan-mod-port
   remote-id: aabb.cc00.0700 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                    Trusted    Allow option    Rate limit (pps)
-----------------------      -------    ------------    ----------------
Ethernet1/1                  yes        yes             unlimited
  Custom circuit-ids:
SW1#
SW1#sh ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  -------------
-------
00:16:9D:43:F7:D2   151.10.16.1      81506       dhcp-snooping   16    Ethernet2/2
Total number of bindings: 1

SW1#
```

Verify the DHCP snooping configuration and bindings on SW2:

```
SW2#sh ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
16
DHCP snooping is operational on following VLANs:
16
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is disabled
   circuit-id default format: vlan-mod-port
   remote-id: aabb.cc00.0800 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                    Trusted    Allow option    Rate limit (pps)
-----------------------      -------    ------------    ----------------
```

```
Ethernet2/2                yes         yes             unlimited
  Custom circuit-ids:
SW2#

SW2#sh ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  -------------
-------
00:16:9D:43:F7:D2   151.10.16.1      81492       dhcp-snooping   16    Ethernet0/0
Total number of bindings: 1

SW2#
```

Issue: Configure all VLAN 25 and VLAN 35 interfaces for a bandwidth of 100 Mb/s. Configure all VLAN 25 interfaces for a delay of 100 microseconds. Configure the R2 and R3 interfaces on subnet 151.10.23.0/24 to have the same bandwidth and delay as the R2 interface on subnet 151.10.25.0/24.

Solution:

Configure and verify the specified bandwidth on the Ethernet interfaces of R2, R3, and R5. Note that Cisco IOS Software uses two different standards for delay. Input measurements in tens of microseconds will display in command output in microseconds.

R2:

```
interface Ethernet0/0
 bandwidth 100000

R2#show int e0/0 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 1000 usec,
R2#
```

R3:

```
interface Ethernet0/0
 bandwidth 100000

R3#show int e0/0 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 1000 usec,
R3#
```

R5:

```
interface Ethernet0/0.25
 bandwidth 100000
 !
interface Ethernet0/0.35
 bandwidth 100000
 !
R5#show int e0/0.25 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 1000 usec,
R5#show int e0/0.35 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 1000 usec,
R5#
```

Configure and verify the specified delay on the Ethernet interfaces of R2 and R5 on VLAN 25.

R2:

```
interface Ethernet0/0
 bandwidth 100000
delay 10
```



```
!
R2#show int e0/0 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
R2#
```

R5:

```
interface Ethernet0/0.25
 bandwidth 100000
 delay 10


 !
R5#show int e0/0.25 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
R5#
```

Configure and verify the bandwidth and delay on the S1/0 interfaces of R2 and R3:

R2:

```
interface Serial1/0
 bandwidth 100000
 delay 10


R2#show int s1/0 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
R2#
```

R3:

```
interface Serial1/0
 bandwidth 100000
 delay 10

R3#show int s1/0 | inc BW
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
R3#
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

3. IPv4 OSPF

Note	All OSPF routers must be configured with only one OSPF Process ID (PID). Use your IGP diagram to help guide configuration.

Issue: Do not form an OSPF neighbor relationship between R1 and R4 in OSPF Area 15. Host routes should be generated for the interfaces that are connected to OSPF Area 15.

Solution:

R1 and R4 should not be able to form adjacencies. Could you make R7 the designated router (DR) and use the broadcast or non-broadcast network types? Adjacencies would form, but the host entries for the OSPF Area 15 interfaces would not be generated.

OSPF can work around these issues by using the point-to-multipoint non-broadcast network type. This network types treats the VLAN as a collection of point-to-point links. The point-to-multipoint non-broadcast network type will turn VLAN 10 into a logical point to point link between R1 and R7 and between R4 and R7. By using this network type, R1, R4, and R7 will generate explicit /32 host routes to their OSPF Area 15 interfaces. Notice that the R1 and R4 routing tables never see the other router as a next hop. You can configure the neighbor statements only on R7.

R1:

```
interface Ethernet0/1.10
 encapsulation dot1Q 10
 ip address 151.10.15.1 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 100
 router-id 151.10.101.1
 area 15 range 151.10.15.0 255.255.255.0
 area 15 virtual-link 151.10.107.1
 network 151.10.15.0 0.0.0.255 area 15
!
```

R4:

```
interface Ethernet0/1
 ip address 151.10.15.4 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 100
 area 15 range 151.10.15.0 255.255.255.0
 area 15 virtual-link 151.10.107.1
 network 151.10.15.0 0.0.0.255 area 15
!
```

R7:

```
interface Ethernet0/0
 ip address 151.10.15.7 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 100
 area 15 range 151.10.15.0 255.255.255.0
 area 15 virtual-link 151.10.104.1
 area 15 virtual-link 151.10.101.1
 network 151.10.15.0 0.0.0.255 area 15
 neighbor 151.10.15.1
 neighbor 151.10.15.4
!
```

Verification:

```
R1#show ip route ospf | inc 151.10.15.
O        151.10.15.4/32 [110/20] via 151.10.15.7, 00:13:22, Ethernet0/1.10
O        151.10.15.7/32 [110/10] via 151.10.15.7, 00:13:22, Ethernet0/1.10
O E1     151.10.30.0/24 [110/104] via 151.10.15.7, 00:13:17, Ethernet0/1.10
O IA     151.10.43.0/24 [110/84] via 151.10.15.7, 00:13:17, Ethernet0/1.10
O E1     151.10.60.0/24 [110/104] via 151.10.15.7, 00:13:17, Ethernet0/1.10
O IA     151.10.104.0/24 [110/21] via 151.10.15.7, 00:13:17, Ethernet0/1.10
O IA     151.10.107.0/24 [110/11] via 151.10.15.7, 00:13:17, Ethernet0/1.10
O E1     151.10.110.0/24 [110/104] via 151.10.15.7, 00:13:17, Ethernet0/1.10
O E1     151.10.120.0/24 [110/104] via 151.10.15.7, 00:13:17, Ethernet0/1.10
<skipped>
```


Issue: OSPF Areas 107, 104 and 43 are not directly connected to OSPF Area 0.

Solution:

Virtual links would be the straightforward solution. But how many do you need, and where should they go? Again, because of the lab requirement that prohibits any type of OSPF neighbor relationship between R1 and R4, including the virtual link neighbor relationship, you will need to go through R7. The solution is to configure a virtual link between R1 and R7, and another between R7 and R4.

R1:

```
interface Ethernet0/1.10
 encapsulation dot1Q 10
 ip address 151.10.15.1 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 100
 router-id 151.10.101.1
 area 15 range 151.10.15.0 255.255.255.0
 area 15 virtual-link 151.10.107.1
 network 151.10.15.0 0.0.0.255 area 15
!
```

R4:

```
interface Ethernet0/1
ip address 151.10.15.4 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 100
 area 15 range 151.10.15.0 255.255.255.0
 area 15 virtual-link 151.10.107.1
 network 151.10.15.0 0.0.0.255 area 15
!
```

R7:

```
interface Ethernet0/0
ip address 151.10.15.7 255.255.255.0
 ip ospf network point-to-multipoint non-broadcast
!
router ospf 100
 area 15 range 151.10.15.0 255.255.255.0
 area 15 virtual-link 151.10.104.1
 area 15 virtual-link 151.10.101.1
 network 151.10.15.0 0.0.0.255 area 15
 neighbor 151.10.15.1
 neighbor 151.10.15.4
!
```

Note that when you issue the command **show ip ospf** on R7, you will see that Area 0 now exists on this router. The following output is the result of entering **show ip ospf virtual-links**.

```
R7# show ip ospf virtual-links
Virtual Link OSPF_VL1 to router 151.10.104.1 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 15, via interface Ethernet0/0
 Topology-MTID    Cost    Disabled     Shutdown      Topology Name
        0          10        no          no            Base
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
    Adjacency State FULL (Hello suppressed)
    Index 2/4, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
```

```
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
Virtual Link OSPF_VL0 to router 151.10.101.1 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 15, via interface Ethernet0/0
 Topology-MTID    Cost    Disabled     Shutdown      Topology Name
        0          10        no          no            Base
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
    Adjacency State FULL (Hello suppressed)
    Index 1/2, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
R7#
```

To meet the requirement that host routes not be advertised to other areas, enter the command **area 15 range 151.10.15.0 255.255.255.0** on *all* the Area 15 Area Border Routers (ABRs). R7 is included, because without this command, the /32 routes will be in summary link-state advertisements (LSAs) in the Area 107 database.

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

4. IPv4 RIP

Issue: Configure RIP version 2 (RIPv2) on R3, R6, R8, and R9. RIP updates must be sent only on the segments between these routers.

Solution:

Use passive interfaces to make sure that RIP updates are sent only over the required interfaces. Some find it a good general practice to issue the command **passive-interface default** and then **no passive** on the necessary interfaces. This process makes the debugs easier to read—just remember to enter **no passive** as required.

The following is partial output for **show ip protocols** on **R3**:


```
R3#show ip protocols | section "rip"
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 26 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    Ethernet0/2            2     2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    151.10.0.0
  Passive Interface(s):
    Ethernet0/0
    Ethernet0/1
    Ethernet0/3
    Serial1/0
    Serial1/1
    Serial1/2
  Passive Interface(s):
    Serial1/3
    Loopback103
    RG-AR-IF-INPUT1
    Tunnel0
    Tunnel1
    VoIP-Null0
  Routing Information Sources:
    Gateway         Distance      Last Update
    151.10.30.10         120      00:00:22
  Distance: (default is 120)
R3#
```

Issue: R6 should not receive prefixes from the other RIP speakers. Do not use passive interface, route filtering, or administrative distance manipulation to accomplish this.

Solution:

One of the options to control the routing information in distance-vector protocols is to set the metric (hop count) using the command **offset-list**. The command **offset-list 0 out VLAN 16** under the RIP process on R8 sets the metric to 16 for all routes sent toward R6. R6 will receive the update but will not use the prefixes to forward packets, because a RIP metric of 16 is infinite and unusable. As a result, the advertised prefixes will not show up in the R6 forwarding table.

Here is some output from the **debug ip rip** command to give you an idea of what is going on. None of these routes are in the R6 routing table or in the R6 RIP database.

```
R6#deb ip rip
RIP protocol debugging is on
R6#
*Aug 16 22:52:00.658: RIP: sending v2 update to 224.0.0.9 via Ethernet0/0.60
(151.10.60.6)
*Aug 16 22:52:00.659: RIP: build update entries
*Aug 16 22:52:00.659:   151.10.16.0/24 via 0.0.0.0, metric 1, tag 0
*Aug 16 22:52:00.659:   151.10.106.0/24 via 0.0.0.0, metric 1, tag 0
R6#
*Aug 16 22:52:13.212: RIP: received v2 update from 151.10.60.10 on Ethernet0/0.60
*Aug 16 22:52:13.212:      151.10.0.0/17 via 0.0.0.0 in 16 hops  (inaccessible)
*Aug 16 22:52:13.212:      151.10.1.0/24 via 0.0.0.0 in 16 hops  (inaccessible)
*Aug 16 22:52:13.212:      151.10.30.0/24 via 0.0.0.0 in 16 hops  (inaccessible)
*Aug 16 22:52:13.212:      151.10.110.0/24 via 0.0.0.0 in 16 hops  (inaccessible)
*Aug 16 22:52:13.212:      151.10.120.0/24 via 0.0.0.0 in 16 hops  (inaccessible)
```

```
R6#u all
All possible debugging has been turned off
R6#
```

Issue: R8 and R9 should have the single smallest route in the routing tables to reach destinations outside of the RIP routing domain via R3.

Solution:

This task asks you to configure the RIP summary on R3. Select the smallest covering summary, which for networks 151.10.15.0/24 and 151.10.111.0/24 is 151.10.0.0/17.

R3:

```
interface Ethernet0/2
 ip address 151.10.30.3 255.255.255.0
 ip summary-address rip 151.10.0.0 255.255.128.0
!

R3#show ip protocols | section "rip"
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 24 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Ethernet0/2           2     2
  Automatic network summarization is in effect
  Address Summarization:
    151.10.0.0/17 for Ethernet0/2
  Maximum path: 4
  Routing for Networks:
    151.10.0.0
  Passive Interface(s):
    Ethernet0/0
    Ethernet0/1
    Ethernet0/3
    Serial1/0
    Serial1/1
    Serial1/2
    Serial1/3
  Passive Interface(s):
    Loopback103
    RG-AR-IF-INPUT1
    Tunnel0
    Tunnel1
    VoIP-Null0
  Routing Information Sources:
    Gateway         Distance      Last Update
    151.10.30.10         120      00:00:27
  Distance: (default is 120)
R3#
R8#show ip route rip
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
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set
```



```
      151.10.0.0/16 is variably subnetted, 11 subnets, 3 masks
R        151.10.0.0/17 [120/2] via 151.10.1.20, 00:00:18, Ethernet0/2
R        151.10.16.0/24 [120/1] via 151.10.60.6, 00:00:14, Ethernet0/0
R        151.10.30.0/24 [120/1] via 151.10.1.20, 00:00:18, Ethernet0/2
R        151.10.106.0/24 [120/1] via 151.10.60.6, 00:00:14, Ethernet0/0
R        151.10.120.0/24 [120/1] via 151.10.1.20, 00:00:18, Ethernet0/2
R8#

R9#sh ip ro rip
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
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      151.10.0.0/16 is variably subnetted, 11 subnets, 3 masks
R        151.10.0.0/17 [120/1] via 151.10.30.3, 00:00:05, Ethernet0/1
R        151.10.16.0/24 [120/2] via 151.10.1.10, 00:00:12, Ethernet0/2
R        151.10.60.0/24 [120/1] via 151.10.1.10, 00:00:12, Ethernet0/2
R        151.10.106.0/24 [120/2] via 151.10.1.10, 00:00:12, Ethernet0/2
R        151.10.110.0/24 [120/1] via 151.10.1.10, 00:00:12, Ethernet0/2
R9#
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

5. IPv4 EIGRP

Issue: Authenticate the EIGRP adjacency over the 151.10.23.0/24 subnet using key **rs?ccie**.

Solution:

As you see in the Mentor Guide output of R2 and R3 for this scenario, EIGRP authentication is a three-step process. First, create a key chain in global configuration mode. Then, on the interface, specify the authentication method and the key chain for the process. The challenge here is to actually enter a question mark character in the key-string command without getting a help scroll. Press **Ctrl-V**, then release those keys and press the **Shift** and **?** keys to insert the question mark into the password string.

Here is a quick verification. You might also want to check that you still have a good EIGRP neighbor.

```
R2#sh ip eigrp interfaces detail | include (^[SFL])|(Auth)
Se1/0                   1        0/0       0/0           3       0/0           50
0
  Authentication mode is md5,  key-chain is "eigrp-fr"
  Authentication mode is not set
Lo102                   0        0/0       0/0           0       0/0            0
0
  Authentication mode is not set
R2#

R2#show key chain eigrp-fr
```

```
Key-chain eigrp-fr:
    key 1 -- text "rs?ccie"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
R2#
```

Issue: On R5, traffic to native OSPF domain prefixes should be load-balanced using a 6:1 ratio favoring R2. Also, R5 should prefer the direct path across VLAN 35 to the RIP domain routes. Influence the routing decision for these prefixes by configuring R5 only.

Solution:

You can achieve load balancing across paths with unequal metrics by using the **variance** command. Eligible paths with metrics equal to or less than the lowest metric times the multiplier are eligible to be placed into the forwarding table. By default, EIGRP will distribute traffic among eligible routes in inverse proportion to their metrics—a path with half the metric of another gets twice as much of the traffic. Also by default, EIGRP calculates its metric using just the minimum bandwidth on a path and the sum of path delays.

Since the OSPF native prefixes are learned via redistribution on both R2 and R3, R5 should have two equal-cost paths for each of these EIGRP external routes. The easiest way to influence the metric is usually to increase or decrease the interface delay parameter. You need to increase the delay parameter on the R5 link to R3 so that the resulting metrics are six times the metrics of the alternate path through R2. The actual EIGRP metric formula is as follows:

Metric = 256 * (10,000,000/minimum bandwidth + sum of delays in tens of microseconds)

The default metrics on these external routes in your pod are 284,160. How was this calculated? First, let's divide it by the scaling factor, 256, to get 1,110. If you look at the output of **show int e0/0.25**, you see that the delay value on this interface is 100 microseconds, which is 10 in tens of microseconds. Subtracting 10 from 1,110 leaves 1,100. You must also subtract the delay value used in the redistribution metric, 100, to get 1,000. The bandwidth configured on E0/0.25 is 100,000 and the bandwidth used during redistribution is 10,000; the minimum bandwidth is 10,000. 1,000 is equal to 10,000,000/10,000. So, the current metric is:

```
284,160 = 256 * (10,000,000/10,000 + 100 + 10)
```

The question comes down to the algebra of determining the delay value, *X*, that will satisfy the following equation:

```
6 * 284,160 = 256 * (10,000,000/10,000 + 100 + X)
6 * 284,160 / 256 = 10,000,000/10,000 + 100 + X
6 * 284,160 / 256 - 10,000,000/10,000 - 100 = X
5,560 = X
```

And your route metric after delay modification will be:

```
1,704,960 = 256 * (10,000,000/10,000 + 100 + 5,560)
```

R5:

```
interface Ethernet0/0.35
 encapsulation dot1Q 35
 bandwidth 100000
 ip address 151.10.35.5 255.255.255.0
 delay 5560
```

To verify the configuration, look at the following example route:

```
R5#show ip route 151.10.107.0
Routing entry for 151.10.107.0/24
  Known via "eigrp 100", distance 170, metric 284160, type external
  Redistributing via eigrp 100
  Last update from 151.10.25.2 on Ethernet0/0.25, 08:59:02 ago
```



```
  Routing Descriptor Blocks:
  * 151.10.35.3, from 151.10.35.3, 08:59:02 ago, via Ethernet0/0.35
      Route metric is 1704960, traffic share count is 1
      Total delay is 56600 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    151.10.25.2, from 151.10.25.2, 08:59:02 ago, via Ethernet0/0.25
      Route metric is 284160, traffic share count is 6
      Total delay is 1100 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
R5#
```

Note that Cisco IOS Software does not use metrics directly for load balancing; instead, it calculates traffic share count (seen in the preceding output). Further, because of rounding in share count calculations, multiple EIGRP metrics may result in a 6:1 traffic share count. The traffic share count is then used by the relevant switching method, which can be verified for Cisco Express Forwarding as follows:

```
R5#show ip cef 151.10.107.0  255.255.255.0 internal
151.10.107.0/24, epoch 0, RIB[I], refcount 5, per-destination sharing
  sources: RIB
  feature space:
   IPRM: 0x00028000
  ifnums:
   Ethernet0/0.25(15): 151.10.25.2
   Ethernet0/0.35(16): 151.10.35.3
  path F09C1F18, path list EE4CDEB4, share 1/1, type attached nexthop, for IPv4
  nexthop 151.10.35.3 Ethernet0/0.35, adjacency IP adj out of Ethernet0/0.35, addr
151.10.35.3 EF7D5698
  path F0A81210, path list EE4CDEB4, share 6/6, type attached nexthop, for IPv4
  nexthop 151.10.25.2 Ethernet0/0.25, adjacency IP adj out of Ethernet0/0.25, addr
151.10.25.2 EF7D57C8
  output chain:
    loadinfo EDF09670, per-session, 2 choices, flags 0003, 14 locks
    flags: Per-session, for-rx-IPv4
    14 hash buckets
      < 0 > IP adj out of Ethernet0/0.35, addr 151.10.35.3 EF7D5698
      < 1 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 2 > IP adj out of Ethernet0/0.35, addr 151.10.35.3 EF7D5698
      < 3 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 4 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 5 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 6 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 7 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 8 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      < 9 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      <10 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      <11 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      <12 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
      <13 > IP adj out of Ethernet0/0.25, addr 151.10.25.2 EF7D57C8
    Subblocks:
     None
R5#
```

The RIP domain prefixes are received by EIGRP from both R3 (via redistribution there) and from R2 (passed on from R3). You cannot use bandwidth/delay parameters to manipulate these routes, so you should offset the scalar metric. To make sure that R5 does *not* load-balance these prefixes, but prefers the direct path over E0/0.35, you need to increase the metric for these paths via E0/0.24 so that it is

(1) MORE than 6 times the metric over E0/0.35.

```
R5#show ip route | b 151.10.110.0
D EX    151.10.110.0
```

```
          [170/1704960] via 151.10.35.3, 00:04:28, Ethernet0/0.35
          [170/286720] via 151.10.25.2, 00:04:28, Ethernet0/0.25
```

(1,704,960*6)+1 = 10,229,761

or

(2) Large enough that the R2 is not selected anymore. In other words, the composite metric needs to be equal to what the peer advertised. Unlike adjusting bandwidth/delay, using an inbound offset changes the router vision to whatever metric the peer has advertised. This means that the router will assume that it can advertise a better metric to that peer and not use the route from the peer.

```
R5#show ip eigrp topology 151.10.110.0 255.255.255.0
IP-EIGRP (AS 100): Topology entry for 151.10.110.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 286720
  Routing Descriptor Blocks:
  151.10.25.2 (Ethernet0/0.25), from 151.10.25.2, Send flag is 0x0
      Composite metric is (286720/284160), Route is External
  151.10.35.3 (Ethernet0/0.35), from 151.10.35.3, Send flag is 0x0
      Composite metric is (1704960/281600), Route is External
```

1,704,960 – 284,160 = 1,420,800

Picking the smaller of two numbers above gives us the following:

R5:

```
router eigrp 100
 variance 6
 offset-list 1 in 1420800 Ethernet0/0.25
!

R5#show ip route 151.10.110.0 | i from
  Last update from 151.10.35.3 on Ethernet0/0.35, 19:32:25 ago
  * 151.10.35.3, from 151.10.35.3, 19:32:25 ago, via Ethernet0/0.35
R5#

R5#show ip eigrp topology 151.10.110.0 255.255.255.0
EIGRP-IPv4 Topology Entry for AS(100)/ID(151.10.105.1) for 151.10.110.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1704960
  Descriptor Blocks:
  151.10.35.3 (Ethernet0/0.35), from 151.10.35.3, Send flag is 0x0
      Composite metric is (1704960/281600), route is External
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 56600 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 151.10.103.1
      External data:
        AS number of route is 0
        External protocol is RIP, external metric is 2
        Administrator tag is 0 (0x00000000)
  151.10.25.2 (Ethernet0/0.25), from 151.10.25.2, Send flag is 0x0
      Composite metric is (1707520/1704960), route is External
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 56700 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
        Originating router is 151.10.103.1
      External data:
        AS number of route is 0
        External protocol is RIP, external metric is 2
```



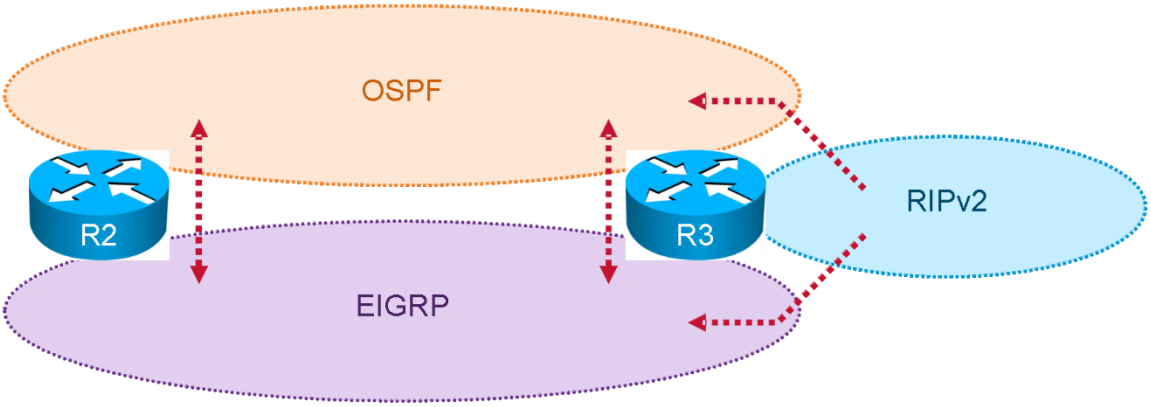
```
        Administrator tag is 0 (0x00000000)
R5#
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

6. IPv4 Route Redistribution

Before examining the specific issues related to configuring each of the IGPs involved in this scenario, survey the entire topology and determine how all the different IGPs will interoperate. Performing such a survey forces you to consider the issues related to route redistribution.

When evaluating a single internetwork topology that contains multiple routing protocols, a good starting point of analysis is to determine if there is more than one direct or indirect connecting point between two routing protocols. If there is only one connecting point between two routing protocols, providing connectivity between them is relatively simple. If there are two or more connecting points, then providing connectivity between the two routing protocols can be complex. When two or more connecting points exist, you can use them to provide redundancy as well as load balancing and optimum path selection. However, when two or more connecting points exist, you must also ensure at a minimum that no routing loops exist and, whenever possible, no suboptimal paths are selected.



The following table provides a useful summary of which prefixes were imported into a given routing protocol. If a permit column for a given routing protocol is completely empty, no prefixes were redistributed into the routing protocol. The lack of prefixes indicates that the routing protocol is involved in one-way redistribution.

Redist point	Into RIP		Into OSPF		Into EIGRP	
	PERMIT	DENY	PERMIT	DENY	PERMIT	DENY
R2			All routes from EIGRP		All routes from OSPF	
R3	Inject Default		All routes from RIP		All routes from RIP	

Issue: When redistributing from OSPF and RIP into EIGRP, use the following metrics:

- Bandwidth 10,000
- Delay 100
- Reliability 255
- Load 1
- MTU 1500

Solution:

This task prepares for EIGRP load sharing.

R3:

```
router eigrp 100
 redistribute rip metric 10000 100 255 1 1500
 redistribute ospf 100 metric 10000 100 255 1 1500
```

Issue: R2 should prefer to reach RIP destinations via the EIGRP path toward R3.

Solution:

By default, R2 would prefer OSPF routes because the default OSPF external administrative distance (110) is better than EIGRP (170) for EIGRP external prefixes. Adjust the OSPF external distance to 180 to make R2 prefer the EIGRP external prefixes:

R2:

```
router ospf 100
 distance ospf external 180
!
```

After the OSPF external distance is adjusted, R3 will have two ways to learn about RIP networks: (a) via RIP and (b) via OSPF as a result of R3 advertising the RIP network via EIGRP to R2 and R2 advertising it via OSPF to R3. The default administrative distance is better for OSPF, which may push RIP routes out of the routing table and potentially set up a routing loop. Configuring an administrative distance of external OSPF routes on R3 to be higher than RIP will prevent this from happening.

R3:

```
router ospf 100
 distance ospf external 180
!
```

Issue: Except as required by the "Multicast" section, routers in OSPF domain should prefer to reach RIP destinations via R3. Routers in the OSPF domain should prefer reaching EIGRP destinations via R2.

Solution:

This task requires configuration of appropriate OSPF metrics.

Move ahead to the "Multicast" section and look at the requirements that are there. The "Multicast" section prohibits you from using static mroutes, and permits only Protocol Independent Multicast (PIM) to be enabled on a subset of the links. When PIM is not enabled on all links, it is highly likely that a Reverse Path Forwarding (RPF) check for either a source or route processor (RP) would fail, resulting in multicast traffic being dropped. To avoid this problem, you need OSPF routers to prefer the route to RIP networks 151.10.1.0/24 (multicast source) and 151.10.103.0/24 (RP) via R2.

You can develop several schemas of OSPF metrics for external routes. In this scenario, only simple selection is needed (R2 better, or R3 better) while maintaining redundancy. So use metric type-1 as "superior" and metric type-2 as "inferior," because OSPF considers one to always be better then another when comparing external routes. Metric type-2 is a default, but is shown in the following route maps for clarity.

R2:

```
router ospf 100
 redistribute eigrp 100 subnets route-map EIGRP-->OSPF
!
ip prefix-list MULTICAST seq 5 permit 151.10.1.0/24
ip prefix-list MULTICAST seq 10 permit 151.10.103.0/24
!
route-map EIGRP-->OSPF permit 10
 description        Set preferred metric to multicast source/RP via R2 to prevent
RPF check failure
 match ip address prefix-list MULTICAST
 set metric-type type-1
!
route-map EIGRP-->OSPF permit 20
 description        Set preferred metric for EIGRP internal, so routers in OSPF
domain prefer reaching EIGRP destinations via R2
 match route-type internal
 set metric-type type-1
!
route-map EIGRP-->OSPF permit 30
 description        Set inferior metric for EIGRP external (RIP), so routers in
OSPF domain prefer reaching RIP destinations via R3
 set metric-type type-2
!
```

R3:

```
router ospf 100
 redistribute eigrp 100 subnets                ! default metric is used here
 redistribute rip subnets route-map RIP-->OSPF
!
ip prefix-list MULTICAST seq 5 permit 151.10.1.0/24
ip prefix-list MULTICAST seq 10 permit 151.10.103.0/24
!
route-map RIP-->OSPF permit 10
 description        Set inferior metric for to multicast source/RP so OSPF routers
route via R2
 match ip address prefix-list MULTICAST
 set metric-type type-2
!
```



```
route-map RIP-->OSPF permit 20
 description        Set preferred metric for routes from RIP, ), so routers in OSPF
domain prefer reaching RIP destinations via R3
 match ip address prefix-list RIP-out
 set metric-type type-1
```

Issue: When redistributing from RIP into OSPF and EIGRP, limit redistribution to the following networks; do not use access control lists (ACLs):

- 151.10.1.0/24
- 151.10.60.0/24
- 151.10.110.0/24
- 151.10.120.0/24
- Connected routes on R3 that are part of the RIP routing domain

Solution:

The following output is the basic redistribution filtering requirement using the prefix list:

R3:

```
router eigrp 100
 redistribute rip metric 10000 100 255 1 1500 route-map RIP-->EIGRP
!
router ospf 100
 redistribute rip subnets route-map RIP-->OSPF
!
ip prefix-list MULTICAST seq 5 permit 151.10.1.0/24
ip prefix-list MULTICAST seq 10 permit 151.10.103.0/24
!
ip prefix-list RIP-out seq 5 permit 151.10.1.0/24
ip prefix-list RIP-out seq 10 permit 151.10.60.0/24
ip prefix-list RIP-out seq 15 permit 151.10.120.0/24
ip prefix-list RIP-out seq 20 permit 151.10.110.0/24
ip prefix-list RIP-out seq 25 permit 151.10.30.0/24
ip prefix-list RIP-out seq 30 permit 151.10.103.0/24
!
route-map RIP-->EIGRP permit 10
 match ip address prefix-list RIP-out
!
route-map RIP-->OSPF permit 10
 description        Set inferior metric for to multicast source/RP so OSPF routers
route via R2
 match ip address prefix-list MULTICAST
 set metric-type type-2
!
route-map RIP-->OSPF permit 20
 description        Set preferred metric for routes from RIP, ), so routers in OSPF
domain prefer reaching RIP destinations via R3
 match ip address prefix-list RIP-out
 set metric-type type-1
```

Issue: Allow for full reachability (except for failed links) if either R2 or R3 loses its connection to the Serial links.

Solution:

This task requires you to allow both the EIGRP and OSPF domains to permit transit for the RIP domain prefixes. In the solution, no filtering is done on the redistributions. Route feedback is handled by relying on the lower administrative distance (AD) for internal prefixes and the higher AD for external prefixes. EIGRP has this protection built in; it was added to OSPF by using the **distance ospf external 180** command on R2 and R3.

To verify that you have met the requirements for redundant paths, shut the S1/0 and S1/1 interfaces on R2, let the paths converge, then rerun your Tool Command Language (Tcl) script. The only failures should be on the shutdown Serial links. In the following output, you see the routing table on R1 with the Serial links shut down on R2:

```
R1#sh ip route ospf
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
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      151.10.0.0/16 is variably subnetted, 24 subnets, 2 masks
O E2     151.10.1.0/24 [110/20] via 151.10.15.7, 00:01:22, Ethernet0/1.10
O        151.10.15.4/32 [110/20] via 151.10.15.7, 1w2d, Ethernet0/1.10
O        151.10.15.7/32 [110/10] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E2     151.10.25.0/24 [110/20] via 151.10.15.7, 00:01:22, Ethernet0/1.10
O E1     151.10.30.0/24 [110/104] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E2     151.10.35.0/24 [110/20] via 151.10.15.7, 00:01:22, Ethernet0/1.10
O IA     151.10.43.0/24 [110/84] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E1     151.10.60.0/24 [110/104] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E2     151.10.102.0/24 [110/20] via 151.10.15.7, 00:01:22, Ethernet0/1.10
O E2     151.10.103.0/24 [110/20] via 151.10.15.7, 00:01:22, Ethernet0/1.10
O IA     151.10.104.0/24 [110/21] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E2     151.10.105.0/24 [110/20] via 151.10.15.7, 00:01:22, Ethernet0/1.10
O        151.10.106.0/24 [110/11] via 151.10.16.6, 1w2d, Ethernet0/1.16
O IA     151.10.107.0/24 [110/11] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E1     151.10.110.0/24 [110/104] via 151.10.15.7, 1w2d, Ethernet0/1.10
O E1     151.10.120.0/24 [110/104] via 151.10.15.7, 1w2d, Ethernet0/1.10
R1#
```

Note that the R1 shows that all RIP and EIGRP networks are now reachable via R7.

Bring the Serial links back up on R2.

You can also repeat the testing process by shutting down the Serial links on R3 and verifying the routing table on R1. Do not forget to enable the Serial links on R3 after you complete the testing.

One way to test that your redistribution satisfies the goal of universal connectivity is to run a Tcl script, like the one in the following output, on each router. Tcl scripting support is available in the Cisco IOS Software versions used here on the routers. The following script is simple and lists all the IP addresses in your pod. It can be built once in Notepad, and then pasted into each router to automate pings. A paper on Tcl scripting is available in the Cisco 360 CCIE R&S Reference Library. Some addresses are used in later tasks and may not be reachable at this point.

Run **tclsh** in privileged mode, paste in the following script, and then issue the command **tclq.**

```
tclsh
foreach address {
151.10.21.1
151.10.16.1
151.10.101.1
151.10.111.1
151.10.15.1
151.10.21.2
151.10.23.2
151.10.25.2
151.10.102.1
151.10.30.3
151.10.43.3
151.10.23.3
151.10.35.3
151.10.103.1
151.10.15.4
151.10.43.4
151.10.104.1
151.10.35.5
151.10.105.1
151.10.60.6
151.10.16.6
151.10.106.1
151.10.15.7
151.10.107.1
```



```
151.10.60.10
151.10.1.10
151.10.110.1
151.10.30.10
```

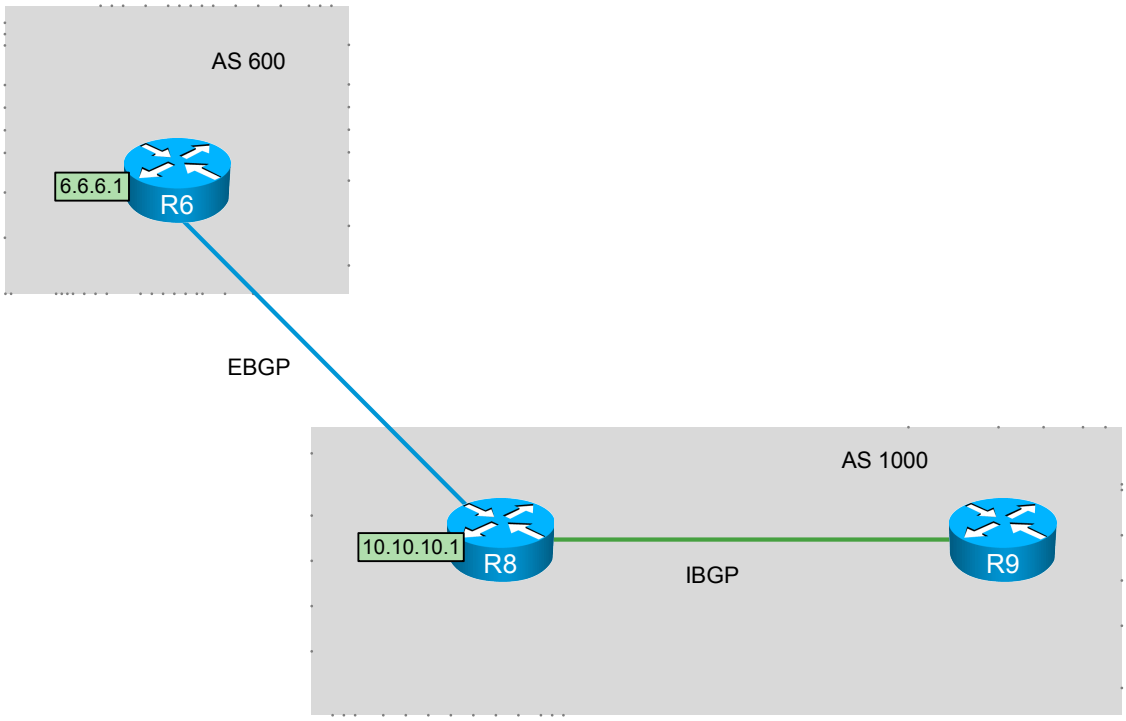
```
151.10.1.20
151.10.120.1
} {ping $address}
```

You also need to make sure that your solution is a stable one. If you have split-horizon issues or other route feedback problems, routes may continually be inserted and removed from our routing tables. You can test stability by observing the output of **debug IP routing**.

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

7. Border Gateway Protocol

Logical BGP Peering



Issue: Peer AS 600 with the Loopback interface 151.10.110.1 of R8.

Solution:

Try a trace from R6 to the R8 loopback address. Notice that the path is via R1. In addition, you are peering to a loopback address. So you will need to configure **EBGP-multihop** on both sides of the R6-R8 peering. R8 will need to peer with 151.10.16.6 (the R6 outgoing interface), and will also require an **update-source loop110** command. The logical depiction below looks deceptively simple.

A good place to start is with a check of the peering relationships with the command **show ip bgp summary**. This output gives a one-line summary for each configured peer. Look for a number under the State/PfxRcd column. An indication of "Active" or "Idle" indicates a problem in forming a TCP session with the peer. Once good peer relationships are verified, check that networks are properly advertised and received with the command **show ip bgp**.

```
R8#sh ip bgp
   Network          Next Hop            Metric LocPrf Weight Path
*> 6.6.6.0/24       151.10.60.6              0             0 600 i
*> 10.10.10.0/24    0.0.0.0                  0         32768 i
```

The > symbol indicates that the prefix is eligible to be placed into the local forwarding table and advertised to peers. If no **>** symbol is indicated for a prefix, get the details for the route with the command **show ip bgp <prefix>**. You will see that the neighbor relationships are good and the BGP routes are good, but can you ping 6.6.6.1 from R8?

The underlying IGP connectivity between AS 600 and AS 1000 is via R3, R4, R7, and R1. Since they do not have the 6.6.6.0 and 10.10.10.0 routes, they will drop the traffic. Here are three possible solutions:

- Create an unnumbered tunnel between R6 and R8 and run External Border Gateway Protocol (EBGP) over it.
- Redistribute BGP into the IGPs on R6 and R9.
- Modify BGP next-hop values with route maps to avoid forwarding via R3, R4, R7, and R1.

The configurations in the Mentor Guide and in this Answer Key use the third option. The basic BGP diagram for this lab looks simple, but beware the interactions between BGP next hops and the underlying IGP reachability.

R8:

```
router bgp 1000
 no synchronization
 bgp log-neighbor-changes
 network 10.10.10.0 mask 255.255.255.0
 neighbor 151.10.1.20 remote-as 1000
 neighbor 151.10.16.6 remote-as 600
 neighbor 151.10.16.6 ebgp-multihop 255
 neighbor 151.10.16.6 update-source Loopback110
 neighbor 151.10.16.6 route-map BGP-next-hop-in in
 neighbor 151.10.16.6 route-map BGP-next-hop-out out
!
route-map BGP-next-hop-in permit 10
 set ip next-hop 151.10.60.6
!
route-map BGP-next-hop-out permit 10
 set ip next-hop 151.10.60.10
!

R6#ping 10.10.10.1 source 6.6.6.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
Packet sent with a source address of 6.6.6.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
R6#
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

8. Router Maintenance

Issue: Capture data on IP flows arriving on the E0/0.16 interface of R6. Send the flow statistics to a management workstation on 151.10.60.100, port 9992. Sample just one Internet Control Message Protocol (ICMP) packet out of every one hundred ICMP packets.

Solution:

The Random Sampled NetFlow feature allows you to reduce demands on the router by randomly processing a sample of the packets. It is configured by applying a NetFlow sampler map to a policy map. For detailed information on configuration, refer to the configuration section “Using NetFlow Filtering or Sampling to Select the Network Traffic to Track” of the *Cisco IOS NetFlow Configuration Guide* at Cisco.com.

First, create a class map that defines the traffic of interest. Then, create a flow map to define the sampling rate.

R6:

```
access-list 101 permit icmp any any
!
flow-sampler-map SAMPLE-RATE
 mode random one-out-of 100
!
flow-sampler-map SAMPLE-RATE-FULL
 mode random one-out-of 1
!
class-map match-all ICMP
 match access-group 101
!
```

Next comes the policy map, which contains all the previous elements:

```
policy-map NETFLOW
 class ICMP
   netflow-sampler SAMPLE-RATE
 class class-default
   netflow-sampler SAMPLE-RATE-FULL
```

Finally, apply the policy on E0/0.16 with the command **service-policy input NETFLOW**.

```
interface Ethernet0/0.16
 service-policy input NETFLOW
!
ip flow-export destination 151.10.60.100 9992
```

Here is the output from the basic **show** command for this feature before a test ping is sent:

```
R6#sh flow-sampler

Sampler : SAMPLE-RATE, id : 1, packets matched : 0, mode : random sampling mode
  sampling interval is : 100

 Sampler : SAMPLE-RATE-FULL, id : 2, packets matched : 0, mode : random sampling
mode
  sampling interval is : 1
```

Then, **debug flow-sampler match** was enabled and a ping was sent from R1 to 151.10.106.1 with a repeat count of 1200. Here is the **show** command output after the test:


```
R6#debug flow-sampler match
Flow sampler match debugging is on
R6#
*Aug  16  23:33:00.956:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:00.960:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:00.992:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.020:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.057:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.066:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.118:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
R6#
*Aug  16  23:33:01.128:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.164:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.182:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.228:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
*Aug  16  23:33:01.235:  Flow:  packet  matched  sampler  SAMPLE-RATE  on  interface
Ethernet0/0.16
R6#sh flow-sampler

 Sampler : SAMPLE-RATE, id : 1, packets matched : 12, mode : random sampling mode
  sampling interval is : 100

 Sampler : SAMPLE-RATE-FULL, id : 2, packets matched : 3, mode : random sampling
mode
  sampling interval is : 1
R6#
```

The log of the debug also showed 12 hits. The output from **show policy-map interface** verifies that the interface did in fact receive all 1200 packets:

```
R6#show policy-map interface
 Ethernet0/0.16

  Service-policy input: NETFLOW

    Class-map: ICMP (match-all)
      1200 packets, 141600 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group 101
      netflow-sampler: SAMPLE-RATE

    Class-map: class-default (match-any)
      8409 packets, 814024 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
      netflow-sampler: SAMPLE-RATE-FULL
R6#
```

If you use Telnet from R1 to 151.10.106.1, you will see the increase in non-ICMP packet count:

```
R6#sh flow-sampler

 Sampler : SAMPLE-RATE, id : 1, packets matched : 12, mode : random sampling mode
  sampling interval is : 100
```

```
 Sampler : SAMPLE-RATE-FULL, id : 2, packets matched : 15, mode : random sampling
mode
  sampling interval is : 1
R6#
```


Issue: All packets originating from 151.10.111.1 and destined to 151.10.105.1 must be forwarded to R6. On R6, all traffic coming from 151.10.111.1 and destined to 151.10.105.1 must be forwarded to R8 with the type of service (ToS) field set to Network Control.

Solution:

This issue clearly suggests using policy routing, since you are routing on a basis other than destination address. The 151.10.111.1 address is a loopback on R1, and 151.10.105.1 is a loopback on R5. The normal path from R1 to R5 is via R2.

R1:

```
R1#sh ip route 151.10.105.1
Routing entry for 151.10.105.0/24
  Known via "ospf 100", distance 110, metric 84, type extern 1
  Last update from 151.10.21.2 on Serial1/0, 00:05:40 ago
  Routing Descriptor Blocks:
  * 151.10.21.2, from 151.10.102.1, 00:05:40 ago, via Serial1/0
      Route metric is 84, traffic share count is 1
R1#
```

The route to 151.10.105.1 on R6 is back through R1:

R6:

```
R6#sh ip route 151.10.105.1
Routing entry for 151.10.105.0/24
  Known via "ospf 100", distance 110, metric 94, type extern 1
  Last update from 151.10.16.1 on Ethernet0/0.16, 00:10:09 ago
  Routing Descriptor Blocks:
  * 151.10.16.1, from 151.10.102.1, 00:10:09 ago, via Ethernet0/0.16
      Route metric is 94, traffic share count is 1
R6#
```

First, write an access list on R1 that defines the traffic. Then, create a route map and reference the access list. In the following output, you see the access list and route map configured on R1. Since this traffic is locally originated, apply the route map in an **ip local policy** statement in global configuration mode.

R1:

```
access-list 180 permit ip host 151.10.111.1 host 151.10.105.1
!
route-map send-111-to-R6 permit 10
 match ip address 180
 set ip next-hop 151.10.16.6
!
ip local policy route-map send-111-to-R6
```

The configuration on R6 is similar. Add a statement to the route map setting the IP precedence to meet the ToS condition. The route map is applied inbound to the 151.10.60.10 interface of R6, rather than in global configuration mode, because the traffic is not locally originated, but is arriving on this interface.

R6:

```
access-list 180 permit ip host 151.10.111.1 host 151.10.105.1
!
route-map send-to-R8 permit 10
 match ip address 180
 set ip precedence network
 set ip next-hop 151.10.60.10
```

```
!
!
interface Ethernet0/0.16
 ip policy route-map send-to-R8
!
```

Use the **trace** command to verify that policy routing is active. You can also use **debug ip policy** on R1 and R6.

When you do a source ping or trace from R1, you see the following outputs from your debug on R1 and R6:

R1:

```
R1#trace 151.10.105.1 source  151.10.111.1
Type escape sequence to abort.
Tracing the route to 151.10.105.1
VRF info: (vrf in name/id, vrf out name/id)
  1 151.10.16.6 0 msec 1 msec 0 msec
  2 151.10.60.10 5 msec 5 msec 5 msec
  3 151.10.1.20 5 msec 5 msec 5 msec
  4 151.10.30.3 5 msec 5 msec 5 msec
  5 151.10.23.2 5 msec 4 msec 5 msec
  6 151.10.25.5 5 msec *  6 msec
R1#
*Aug 26 01:23:40.639: IP: s=151.10.111.1 (local), d=151.10.105.1, len 28, policy match
*Aug 26 01:23:40.639: IP: route map send-111-to-R6, item 10, permit
*Aug 26 01:23:40.639: IP: s=151.10.111.1 (local), d=151.10.105.1 (Ethernet0/1.16), len 28,
policy routed
*Aug 26 01:23:40.639: IP: local to Ethernet0/1.16 151.10.16.6
```

R6:

```
*Aug 26 01:23:40.639: IP: s=151.10.111.1 (Ethernet0/0.16), d=151.10.105.1, len 28, policy
match
*Aug 26 01:23:40.639: IP: route map send-to-R8, item 10, permit
*Aug 26 01:23:40.639: IP: s=151.10.111.1 (Ethernet0/0.16), d=151.10.105.1 (Ethernet0/0.60),
len 28, policy routed
*Aug 26 01:23:40.639: IP: Ethernet0/0.16 to Ethernet0/0.60 151.10.60.10
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

9. IPv6 Routing

Issue: Configure IPv6 addresses on R2, R3, R4, R7, and SW4 according to lab requirements.

Solution:

On each router, enter the **IPv6 unicast-routing** command in global configuration mode. Then, assign the addresses as directed. See the Mentor Guide engine for specific configuration.

Issue: Complete IPv6 OSPF tuning. Use the same OSPF network type on all interfaces.

Solution:

IPv6 OSPF exhibits behavior similar to IPv4: the loopback interfaces are advertised as host routes. Because the scenario requires advertisement of loopback interfaces with the original mask, you need to do one of the following:

- Redistribute the loopbacks into OSPF (redistribution does not change the mask).
- Advertise the loopbacks into a different area and perform summarization.
- Change the default IPv6 OSPF network type on loopback interfaces.

Of these three possible solutions, only the last one is applicable to this scenario.

OSPFv3 supports only point-to-point network type on the Loopback interfaces. Here is an example from R2:

```
R2(config-if)#ipv6 ospf network ?
  broadcast            Specify OSPF broadcast multi-access network
  manet                Specify MANET OSPF interface type
  non-broadcast        Specify OSPF NBMA network
  point-to-multipoint  Specify OSPF point-to-multipoint network
  point-to-point       Specify OSPF point-to-point network
R2(config-if)#ipv6 ospf network broadcast
% OSPFv3: Invalid type for interface
R2(config-if)#ipv6 ospf network manet
% OSPFv3: Invalid type for interface
R2(config-if)#ipv6 ospf network point-to-multipoint
% OSPFv3: Invalid type for interface
R2(config-if)#ipv6 ospf network non-broadcast
% OSPFv3: Invalid type for interface
R2(config-if)#ipv6 ospf network point-to-point
R2(config-if)#
```

Note that you can only change the Loopback interface OSPFv3 network type to point-to-point.

You also need to configure the same network type on all other interfaces. Manual configuration of the point-to-point OSPF network type on the Ethernet and Loopback interfaces resolves this issue.

R2 (example):

```
interface Loopback102
 ip address 151.10.102.1 255.255.255.0
 ipv6 address 2001:102::1/64
 ipv6 ospf network point-to-point
 ipv6 ospf 1 area 0
!
interface Ethernet0/0
  ipv6 address 2001:25::2/64
 ipv6 ospf 1 area 0
 ipv6 ospf network point-to-point
```

Issue: Complete IPv6 OSPF tuning. Use Secure Hash Algorithm (SHA) to authenticate OSPF packets between R2 and R3. Use a key constructed as a repeating pattern of 204 (decimal).

Solution:

Authentication and encryption for IPv6 OSPF utilizes IPsec, as described in the configuration section "Configuring IPsec on OSPF for IPv6" of the *Cisco IOS IPv6 Configuration Guide* at Cisco.com.

There are two modes of operation—authentication-only and encryption—and they can be enabled on the interface or in the area. This scenario asks for authentication on the interface. The authentication key is a string of 40 hexadecimal digits. The scenario asks for a "repeating pattern of 204 (decimal)." When you convert 204 into hexadecimal, you get 0xCC.

R2 (example):

```
interface Serial1/0
 ipv6 address FE80::2 link-local
 ipv6 address 2001:23::2/64
 ipv6 ospf 1 area 0
 ipv6 ospf authentication ipsec spi 300 sha1
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!


R2#show ipv6 ospf interface s1/0
Serial1/0 is up, line protocol is up
  Link Local Address FE80::2, Interface ID 7
  Area 0, Process ID 1, Instance ID 0, Router ID 151.10.102.1
  Network Type POINT_TO_POINT, Cost: 1
  SHA-1 authentication SPI 300, secure socket UP (errors: 0)
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:04
  Graceful restart helper support enabled
  Index 1/3/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 3, maximum is 3
  Last flood scan time is 1 msec, maximum is 1 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 151.10.103.1
  Suppress hello for 0 neighbor(s)
R2#
```

Issue: Configure the IPv6 EIGRP AS 100 between routers R3 and R4. Introduce the IPv6 addresses of the loopback interface of R3 and R4 into IPv6 EIGRP AS 100.

Solution:

Enable IPv6 EIGRP AS 100 on the Serial1/1 and Loopback103 interfaces of R3 and on the Serial0/1 and Loopback104 interfaces of R4:

R3:

```
!
interface Loopback103
ipv6 address 2001:103::1/64
 ipv6 eigrp 100
!

interface Serial1/1
 ipv6 address 2001:43::3/64
 ipv6 eigrp 100
```



```
!
```

R4:

```
!
interface Loopback104
ipv6 address 2001:104::1/64
 ipv6 eigrp 100
!

interface Serial1/0
 ipv6 address 2001:43::4/64
 ipv6 eigrp 100
!
```

Verify the IPv6 EIGRP AS 100 neighbor relationship between R3 and R4 and the routing table. Here is an example from R3:

R3:

```
!
R3#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   Link-local address:     Se1/1                    13 12:28:49   14   100  0  12
    FE80::A8BB:CCFF:FE00:400
R3#
R3#show ipv6 route eigrp | begin 2001
D   2001:104::/64 [90/2297856]
     via FE80::A8BB:CCFF:FE00:400, Serial1/1
R3#!
```

Issue: Perform mutual redistribution between IPv6 EIGRP AS 100 and OSPFv3 on R3. Redistribute the VLAN 10 IPv6 network into IPv6 EIGRP AS 100 as connected. Configure the static IPv6 default route on R7.

Solution:

Perform the required IPv6 route redistribution configuration on R3 and R4:

R3:

```
ipv6 router eigrp 100
 redistribute ospf 1
 redistribute connected
 default-metric 1 1 1 1 1
!

ipv6 router ospf 1
 redistribute connected
 redistribute eigrp 100
!
```

Note that you can choose different redistribution metrics.

R4:

```
!
ipv6 router eigrp 100
 redistribute connected metric 1 1 1 1 1
!
```

Verify the IPv6 connectivity. You can use a Tcl script similar to the following:

R3:

```
tclsh
foreach address {
2001:15::7
2001:43::3
2001:103::1
2001:23::3
2001:23::2
2001:102::1
2001:25::2
2001:15::4
2001:104::1
2001:43::4
2001:25::40
2001:140::1
} {ping $address}
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

10. Security

Issue: Permit the required types of traffic in the extended numbered access list on R5. Your solution should not disturb any other IPv4 traffic. Apply your access list on all Ethernet interfaces on R5.

Solution:

Configure the extended numbered access list on R5 and apply to all Ethernet interfaces:

R5:

```
access-list 101 permit ip any any precedence network
access-list 101 permit ip any any precedence critical
access-list 101 permit ip any any dscp 25
access-list 101 permit ip any any dscp af22
access-list 101 permit ip any any

interface Ethernet0/0.25
 ip address 151.10.25.5 255.255.255.0
 ip access-group 101 in
!
interface Ethernet0/0.35
 ip address 151.10.35.5 255.255.255.0
 ip access-group 101 in
!
```

Note that the Cisco IOS Software automatically translates the numbers to names where applicable. DSCP af22 is equal to DSCP decimal value 20.

Also note that the last statement of the access list permits all IP traffic to meet the lab requirement for not disturbing any other types of IP traffic.

This access list can be used for QoS section configuration testing.

Issue: Permit the required types of traffic in the extended named access list on R4. Your solution should not disturb any other IPv6 traffic. Apply your access list on all Ethernet interfaces on R5.

Solution:

Configure the extended named access list on R4 and apply it to the Serial1/0 interface:

R4:

```
ipv6 access-list Lo102_to_Lo104
 permit tcp host 2001:102::1 host 2001:104::1 eq telnet
 deny tcp any host 2001:104::1 eq telnet
 deny tcp any host 2001:43::4 eq telnet
 deny tcp any host 2001:15::4 eq telnet
 permit ipv6 any any
!

interface Serial1/0
 ipv6 address 2001:43::4/64
 ipv6 eigrp 100
 ipv6 traffic-filter Lo102_to_Lo104 in
!
```

Telnet from the Loopback102 interface of R2 to the Loopback104 interface of R4:

R2:

```
R2#telnet 2001:104::1 /source-interface Loopback102
Trying 2001:104::1 ... Open

------------------------------------------------
Cisco 360 R&S Exercise Workbook
Product, POD location:  cierswbv5-ce-lab10-sc, SJ
Device:                 R4
------------------------------------------------

R4#exit

[Connection to 2001:104::1 closed by foreign host]
R2#
```

Note that the Telnet session is successful.

Now try to Telnet from any other interface of R2 to the Loopback104 interface of R4:

R2:

```
R2#telnet 2001:104::1
Trying 2001:104::1 ...
% Destination unreachable; gateway or host down

R2#
```

Note that the Telnet session is not successful.

Now try to Telnet and ping from any other interface of R2 to the Serial1/0 and Ethernet0/1 interface of R4:

R2:

```
R2#telnet 2001:104::1
Trying 2001:104::1 ...
```

```
% Destination unreachable; gateway or host down

R2#telnet 2001:43::4
Trying 2001:43::4 ...
% Destination unreachable; gateway or host down

R2#ping 2001:43::4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:43::4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/13 ms
R2#telnet 2001:15::4
Trying 2001:15::4 ...
% Destination unreachable; gateway or host down

R2#ping 2001:15::4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:15::4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms
R2#
```

Note that the Telnet sessions are not successful, but pings are successful.

Verify the IPv6 access list counters on R4:

```
R4#show ipv6 access-list
IPv6 access list Lo102_to_Lo104
    permit tcp host 2001:102::1 host 2001:104::1 eq telnet (21 matches) sequence 10
    deny tcp any host 2001:104::1 eq telnet (26 matches) sequence 20
    deny tcp any host 2001:43::4 eq telnet (1 match) sequence 30
    deny tcp any host 2001:15::4 eq telnet (1 match) sequence 40
    permit ipv6 any any (81 matches) sequence 50
R4#
```

Note that the access list matches the types of traffic according to the requirements of the lab. Your counters may differ from the counters in the access list.

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

11. Quality of Service

Issue: Mark and re-mark the traffic.

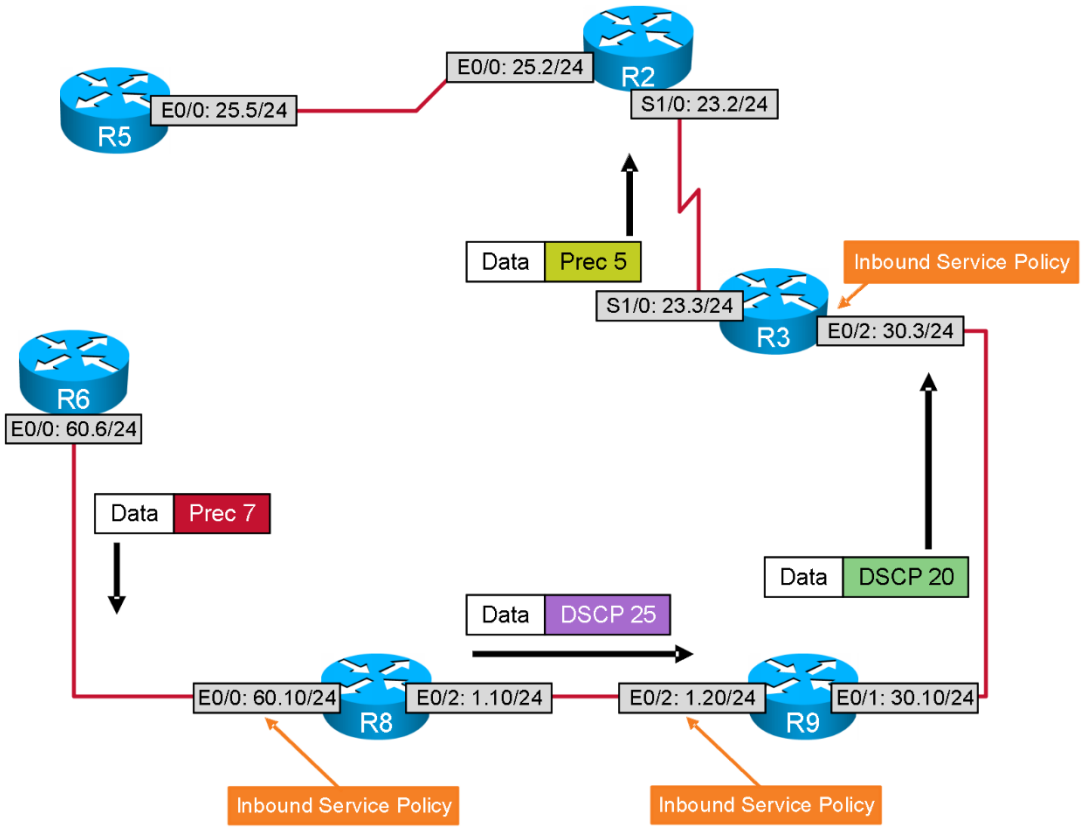
Solution:

In the policy routing section of the lab, you marked traffic sourced from 151.10.111.1 and destined to 151.10.105.1 with IP precedence **network**, which is precedence level 7. Because that traffic arrives at R8, you want to identify and match it in a class map, and then set the differentiated services code point (DSCP) value to 25. DSCP values, defined in RFC 2474, represent a new way to define the ToS byte in the IP header, and range from 0 to 63. When that traffic arrives at R9, it should be re-marked with DSCP value 20, and when the traffic with DSCP value 20 arrives at R3, it should be marked with the IP precedence value **critical** (5). R3 forwards traffic marked with IP precedence **critical** to R5.

The basic process is one of matching traffic with a class map, setting the desired marking in a policy map, and then applying the policy inbound to the required interface. If you perform a

source trace to 151.10.105.1 on R1, you will see that the incoming interface on R5 is the interface with address 151.10.25.5. You can test the effects of your marking efforts by observing the counts on the **show access-list** output. The access list is configured in the "Security" section of this lab.

Marking Precedence and DSCP Values



Ping from R1 and verify the access list counters on R5:

```
R1#ping 151.10.105.1 source 151.10.111.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.10.105.1, timeout is 2 seconds:
Packet sent with a source address of 151.10.111.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/28 ms
R1#

R5#sh access-lists 101
Extended IP access list 101
    10 permit ip any any precedence network (5 matches)
    20 permit ip any any precedence critical
    30 permit ip any any dscp 25
    40 permit ip any any dscp af22
```

```
    50 permit ip any any (12 matches)
R5#
```

Create your class map and policy map on R8, and apply them inbound to the Ethernet0/0 interface of R8. Clear your access-list counters on R5, and source ping again from R1. You should be getting hits against the line matching on DSCP 25.

R8:

```
class-map match-all in-from-R6
  match ip precedence 7
!
!
policy-map in-from-R6
  class in-from-R6
   set dscp 25
!
interface Ethernet0/0
 service-policy input in-from-R6
```

R5:

```
R5#clear access-list counters
R5#
```

R1:

```
R1#ping 151.10.105.1 source 151.10.111.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.10.105.1, timeout is 2 seconds:
Packet sent with a source address of 151.10.111.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/28 ms
R1#
```

R5:

```
R5#show access-lists 101
Extended IP access list 101
    10 permit ip any any precedence network
    20 permit ip any any precedence critical
    30 permit ip any any dscp 25 (5 matches)
    40 permit ip any any dscp af22
    50 permit ip any any (7 matches)
R5#
```

In order to re-mark DSCP on R9, you will have to create a class map that matches on DSCP 25 and then set DSCP to 20 under that class in a policy map. Apply the policy map inbound using the **service-policy** command on R9's Ethernet0/2 interface. You should now get hits against the af22 line on your R5 access list.

R9:

```
class-map match-all in-from-R8
  match ip dscp 25
!
policy-map in-from-R8
  class in-from-R8
   set dscp af22
!
interface Ethernet0/2
```



```
 service-policy input in-from-R8
```

R5:

```
R5#clear access-list counters
R5#
```

R1:

```
R1#ping 151.10.105.1 source 151.10.111.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.10.105.1, timeout is 2 seconds:
Packet sent with a source address of 151.10.111.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/28 ms
R1#
```

R5:

```
R5#show access-lists 101
Extended IP access list 101
    10 permit ip any any precedence network
    20 permit ip any any precedence critical
    30 permit ip any any dscp 25
    40 permit ip any any dscp af22 (5 matches)
    50 permit ip any any (6 matches)
R5#
```

Finally, create and apply a class map and policy map on R3 that matches on DSCP 20 and sets IP precedence to **critical**. To verify that this task is complete, repeat your source pings from R1 and make sure that you are getting hits against the access list entry matching on IP precedence **critical** on R5.

R3:

```
class-map match-all in-from-R9
 match ip dscp af22
!
class-map match-all in-from-R9
 match ip dscp af22
!
!
policy-map in-from-R9
 class in-from-R9
  set precedence 5
!
Interface Ethernet0/2
```

R5:

```
R5#clear access-list counters
R5#
```

R1:

```
R1#ping 151.10.105.1 source 151.10.111.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 151.10.105.1, timeout is 2 seconds:
Packet sent with a source address of 151.10.111.1
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/26/28 ms
R1#
```

R5:

```
R5#show access-lists 101
Extended IP access list 101
    10 permit ip any any precedence network
    20 permit ip any any precedence critical (5 matches)
    30 permit ip any any dscp 25
    40 permit ip any any dscp af22
    50 permit ip any any (3 matches)
R5#
```

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.

12. Multicast

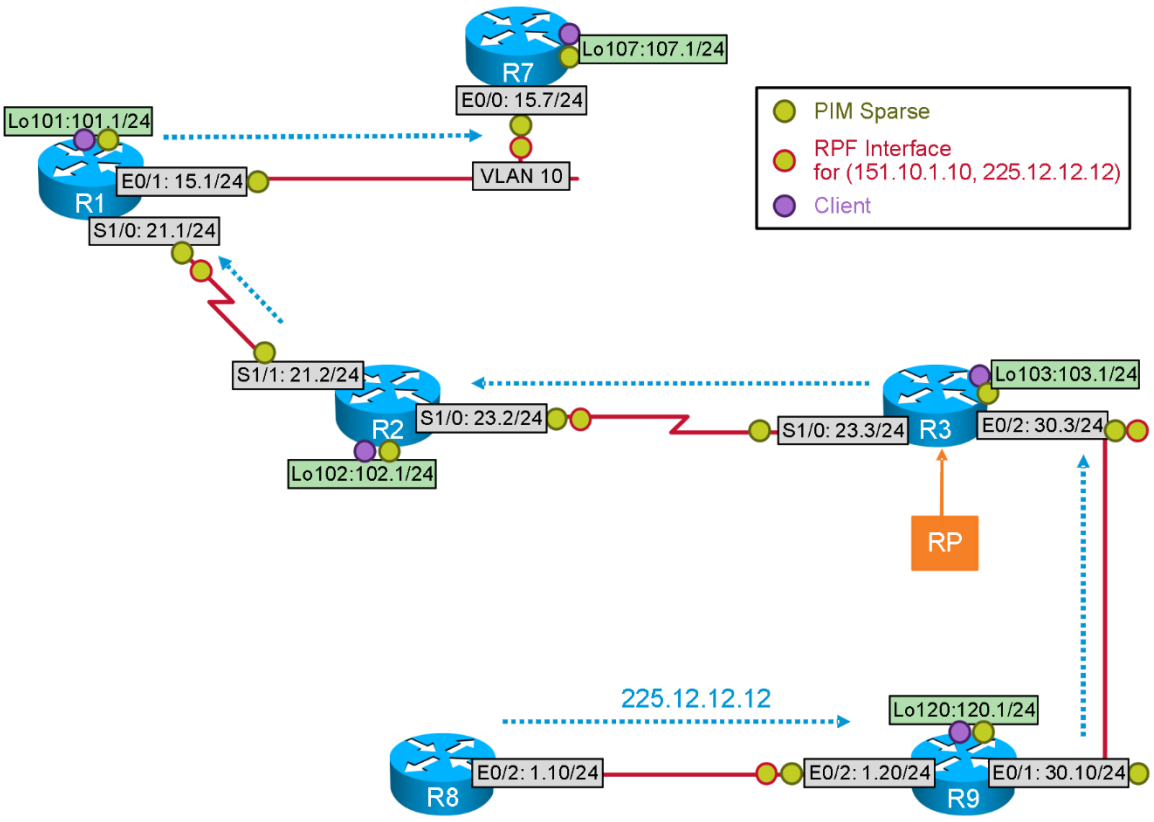
Issue: 1. Configure the specified neighbor relationships.
2. Statically configure a rendezvous point on R3.
3. Join one of the loopback interfaces of each multicast router to group 225.12.12.12.
4. Source multicast traffic from R8.
5. Make sure that all joined interfaces reply.

Solution:

Translating the requirements into a diagram is a good place to start. It is helpful to identify the locations of sources and clients and to clearly represent any multiaccess network segments.

To perform the basic configuration:

- Enter **ip multicast-routing** in global configuration mode on the multicast routers (R9, R3, R2, R1, and R7).
- Enter the command **ip pim sparse-mode** on each of the interfaces shown in green on the diagram.
- Indicate the RP by entering the command **ip pim rp-address 151.10.103.1** on each multicast router (including R3).
- Create multicast clients with the command **ip igmp join-group 225.12.12.12** on each of the indicated loopback interfaces.



To verify that you have completed the basic configuration, you might want to start with the command **show ip pim neighbor** on each multicast router. As shown in the following output, R1 has two PIM neighbors. Is that correct?

```
R1#sh ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
151.10.15.7       Ethernet0/1.10           1w2d/00:01:18     v2    1 / DR S P G
151.10.21.2       Serial1/0                1w2d/00:01:20     v2    1 / S P G
R1#
```

Since this is sparse mode, your next verification task might be to make sure that all multicast routers know about the RP. On each multicast router, enter **show ip pim rp mapping**. In the following output, you see the result for R1. It is not surprising that there has been a successful result, since you configured the RP address statically, but this step could be a major troubleshooting step if you were using Auto-RP.

```
R1#sh ip pim rp mapping
PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 151.10.103.1 (?)
R1#
```

Finally, try an extended ping to the group address from R8:

```
R8#ping 225.12.12.12 repeat 999
```

It is suggested that you use a high repeat count to avoid a multicast stream timeout while you are trying to troubleshoot.

Issue: Make sure that all joined interfaces reply.

Solution:

Assuming no basic configuration errors and successful distribution of RP information, troubleshooting this problem usually comes down to fixing RPF errors and outgoing interface list (OIL) errors. These problems can make it impossible for the router to form shared and source distribution trees.

One way to fix RPF errors is to use static mroutes, but static mroutes are not possible here. Instead, the redistribution was designed to make sure that that R1 and R7 used the multicast-capable path for the reverse paths to 151.10.103.0 and 151.10.1.0. See the IPv4 “Redistribution” section for the details of this tuning.

```
R1#show ip route 151.10.103.0
Routing entry for 151.10.103.0/24
  Known via "ospf 100", distance 110, metric 84, type extern 1
  Last update from 151.10.21.2 on Serial1/0, 1w2d ago
  Routing Descriptor Blocks:
  * 151.10.21.2, from 151.10.102.1, 1w2d ago, via Serial1/0
      Route metric is 84, traffic share count is 1
R1#show ip route 151.10.1.0
Routing entry for 151.10.1.0/24
  Known via "ospf 100", distance 110, metric 84, type extern 1
  Last update from 151.10.21.2 on Serial1/0, 1w2d ago
  Routing Descriptor Blocks:
  * 151.10.21.2, from 151.10.102.1, 1w2d ago, via Serial1/0
      Route metric is 84, traffic share count is 1
R1#

R7#show ip route 151.10.103.0
Routing entry for 151.10.103.0/24
  Known via "ospf 100", distance 110, metric 94, type extern 1
  Last update from 151.10.15.1 on Ethernet0/0, 1w2d ago
  Routing Descriptor Blocks:
  * 151.10.15.1, from 151.10.102.1, 1w2d ago, via Ethernet0/0
      Route metric is 94, traffic share count is 1
R7#show ip route 151.10.1.0
Routing entry for 151.10.1.0/24
  Known via "ospf 100", distance 110, metric 94, type extern 1
  Last update from 151.10.15.1 on Ethernet0/0, 1w2d ago
  Routing Descriptor Blocks:
  * 151.10.15.1, from 151.10.102.1, 1w2d ago, via Ethernet0/0
      Route metric is 94, traffic share count is 1
R7#


R8#ping 225.12.12.12 repeat 20 timeout 1
Sending 20, 100-byte ICMP Echos to 225.12.12.12, timeout is 1 seconds:

Reply to request 52 from 151.10.120.1, 1 ms
Reply to request 52 from 151.10.107.1, 71 ms
Reply to request 52 from 151.10.101.1, 52 ms
Reply to request 52 from 151.10.102.1, 26 ms
Reply to request 52 from 151.10.103.1, 5 ms
. . . . .
Reply to request 53 from 151.10.120.1, 1 ms
Reply to request 53 from 151.10.107.1, 22 ms
Reply to request 53 from 151.10.101.1, 22 ms
```



```
Reply to request 53 from 151.10.102.1, 13 ms
Reply to request 53 from 151.10.103.1, 13 ms
```

Note that there is a response from the loopback interface of each router configured with a client.

Note	To obtain a comprehensive view of the configuration tasks in this section, access the Mentor Guide engine. With the Mentor Guide engine, you can enter more than 1000 Cisco IOS Software commands, as well as a collection of proprietary commands such as **show all**.