360RSW05-LAB-CA02

Cisco Expert-Level Training for CCIE Routing and Switching Assessment Lab 2 (CIERS1) Configuration Section

Cisco Expert-Level Training for CCIE® Routing and Switching (R&S) Advanced Workshop 1 introduces a blended learning approach for CCIE preparation.

This six-hour online lab assesses your grasp of the technology and configuration skills needed to pass the Cisco CCIE Routing and Switching lab exam. The lab provides an experience similar to performing the actual Cisco CCIE lab. Like the actual CCIE lab, tasks are presented in language that forces students to carefully analyze tasks, consider all options, and spot issues. Like the actual CCIE lab, the students are under time pressure to complete as many sections of the lab as possible.

However, unlike the actual Cisco CCIE lab, this lab rates tasks as basic, intermediate, and advanced to facilitate the learning process. Furthermore, unlike the actual CCIE lab, this lab provides extensive feedback to maximize the learning experience after the lab is scored. Detailed online answer keys deliver feedback and access to Mentor Guide output that compares the state of the end-of-lab pod of the student with a Master Answer Key pod.

Cisco Expert-Level Training for CCIE Routing and Switching Assessment Lab 2 (CIERS1) Answer Key Configuration Section

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.

Cisco Expert-Level Training for CCIE Routing and Switching Assessment Lab 2 (CIERS1) Answer Key Configuration Section

Regardless of any configuration that you perform in this lab, you must conform to the general guidelines provided. If you do not conform, you can expect a significant deduction of points in your final exam score.

Grading and Duration

- Configuration lab duration: 6 hours
- Configuration lab maximum score: 76 points
- Minimum passing score: 61 point

Restrictions and Goals

Read this section carefully.

- To receive any credit for a subsection, you must complete the subsection. You will not get partial credit for partially completed subsections.
- IP subnets on the IPv4 IGP diagram belong to network 172.16.0.0/16.
- Do not introduce any new IP addresses or new tunnel links unless explicitly specified.
- Do not use any IP version 4 (IPv4) static routes, the **ip default-network** command, or **default-information originate** command.
- Advertise loopback interfaces with their original masks.
- All IP addresses involved in this scenario must be reachable, unless specified otherwise.
- Networks that are received from backbone routers, networks that are connected to the shared equipment, and networks that are involved in the Multiprotocol Label Switching (MPLS) section and Dynamic Multipoint Virtual Private Network (DMVPN) nonbroadcast multiaccess (NBMA) addressing are excluded from the reachability requirement.
- The backbone router BB1 is reachable via 170.100.10.110.

- The backbone router BB2 is reachable via 173.35.33.100.
- Prefixes that are advertised from the backbone and interfaces that are connected to shared equipment need not be reachable from within your pod.
- Do not modify the hostname, console, or vty configuration unless specified otherwise.
- Do not modify the initial interface or IP address numbering.

Explanation of Each of the Restrictions and Goals

IP subnets in the scenario diagram belong to network 172.16.0.0/16.

The third and forth octets of the IP addresses that are displayed on the diagram belong to 172.16.0.0/16.

Do not introduce any new IP addresses.

Do not use any static routes, ip default-network **command, or** default-information originate **command.**

Static routes can solve a range of reachability problems. However, you cannot use them. You must rely on skillful configuration of all your unicast routing protocols. The scenario is not concerned about the static route that is created by the Cisco IOS Software protocol or feature.

All IP addresses involved in this scenario must be reachable, unless specified otherwise.

This is a key goal and requires that all interior gateway protocols (IGPs) and routing policy tasks be configured properly. The key elements of your routing policy include route redistribution and the controlling of routing updates using distribute lists, route maps, and the **distance** command.

Although the term "redistribution" is never explicitly used in this exam, you must perform redistribution to ensure that all IP addresses are reachable without the use of static routes.

Do not modify the hostname, console and vty configuration, initial interface, and IP address numbering. Follow the numbering conventions carefully.

1. DMVPN and Serial Communications Section

Issue: Configure the multipoint Generic Routing Encapsulation (mGRE) Tunnel124 interfaces on R1, R2, and R4. Use the Ethernet0/2 interfaces on the subnet 10.1.1.0/24 for the Tunnel124 source.
Solution:

Configure the mGRE Tunnel124 on R1, R2, and R4 according to the scenario requirements:

R1:

```
interface Tunnel124
 ip address 172.16.124.1 255.255.255.0
 tunnel source Ethernet0/2
 tunnel mode gre multipoint
 tunnel key 124
```

R2:

```
interface Tunnel124
 ip address 172.16.124.2 255.255.255.0
 tunnel source Ethernet0/2
 tunnel mode gre multipoint
 tunnel key 124
```

R4:

```
interface Tunnel124
 ip address 172.16.124.4 255.255.255.0
 tunnel source Ethernet0/2
 tunnel mode gre multipoint
 tunnel key 124
```

Note that the **tunnel key 124** command is used to configure the tunnel key in this answer key. Because this lab does not specify the tunnel key value, you can use any number as long as it matches between the tunnel endpoints. Also, note that the tunnel key configuration is optional on a router with only one mGRE tunnel interface corresponding to one IP address. Here is an excerpt from the documentation at http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/DMVPN_2_Phase2.html#wp37601 :

The protocol header for an mGRE packet is four bytes larger than a p2p GRE packet. The additional four bytes constitute a tunnel key value, which is used to differentiate between different mGRE interfaces in the same router. Without a tunnel key, a router can support only one mGRE interface corresponding to one IP network. Tunnel keys allow a branch router to have a different mGRE interface corresponding to each DMVPN cloud in the network topology. A headend router can be configured as well with two mGRE interfaces pointing to each DMVPN cloud for high availability and redundancy.

Cisco IOS Software Releases 12.3(13)T, 12.3(11)T3, or later allow multiple mGRE interfaces on a single router to be configured without tunnel keys. Each mGRE interface must then reference a unique IP address as its tunnel source.

Issue: Configure R1 as a next-hop server for the Next Hop Resolution Protocol (NHRP) spokes R2 and R4. Supply the NHRP next-hop server mapping on R2 and R4. Do not configure any NHRP mapping for unicast traffic on R1. Provide mapping for the multicast and broadcast traffic between R1 and R2, and between R1 and R4.

Solution:

Configure the NHRP and DMVPN on R1, R2, and R4 according to the scenario requirements:

R1:

```
interface Tunnel124
 ip address 172.16.124.1 255.255.255.0
 ip nhrp map multicast dynamic
 ip nhrp network-id 124
 tunnel source Ethernet0/2
 tunnel mode gre multipoint
 tunnel key 124
```

R2:

```
interface Tunnel124
 ip address 172.16.124.2 255.255.255.0
 ip nhrp map 172.16.124.1 1.1.1.1
 ip nhrp map multicast 1.1.1.1
 ip nhrp network-id 124
 ip nhrp nhs 172.16.124.1
 tunnel source Ethernet0/2
 tunnel mode gre multipoint
 tunnel key 124
```

R4:

```
interface Tunnel124
 ip address 172.16.124.4 255.255.255.0
 ip nhrp map 172.16.124.1 1.1.1.1
 ip nhrp map multicast 1.1.1.1
 ip nhrp network-id 124
 ip nhrp nhs 172.16.124.1
 tunnel source Ethernet0/2
 tunnel mode gre multipoint
 tunnel key 124
```

Note that R1 is defined as a next-hop server and the NHRP mapping for next-hop server for unicast traffic is done on the NHRP spokes R2 and R4. The multicast traffic is mapped statically on R2 and R4, and dynamically on R1. Also the DMVPN network ID is defined on all DMVPN routers with the **ip nhrp network-id 124** command.

Verification:

Verify the NHRP registrations on the DMVPN hub R1:

```
R1#show ip nhrp
172.16.124.2/32 via 172.16.124.2
   Tunnel124 created 17:21:09, expire 01:58:49
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 1.1.1.2
172.16.124.4/32 via 172.16.124.4
   Tunnel124 created 17:21:09, expire 01:58:49
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 1.1.1.4
R1#
R1#show ip nhrp multicast
  I/F     NBMA address
Tunnel124  1.1.1.2         Flags: dynamic           (Enabled)
Tunnel124  1.1.1.4         Flags: dynamic           (Enabled)
R1#
```

Verify the NHRP registrations on one of the DMVPN spokes, for example on R2:

```
R2#show ip nhrp
172.16.124.1/32 via 172.16.124.1
   Tunnel124 created 17:23:41, never expire
   Type: static, Flags: used
```

```
   NBMA address: 1.1.1.1
R2#
R2#show ip nhrp multicast
  I/F     NBMA address
Tunnel124  1.1.1.1          Flags: static            (Enabled)
R2#
```

Verify the DMVPN connectivity. Here is an example on R2:

```
R2#ping 172.16.124.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.124.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
R2#ping 172.16.124.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.124.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
R2#
```

Note that the spoke R2 can ping the hub R1 and the other spoke, R4.

Issue: Configure PPP encapsulation on the serial link between R1 and R3. R1 and R3 should not advertise host routing entries for the PPP endpoints.

Solution:

The default encapsulation on the Cisco router serial interfaces is High-Level Data Link Control (HDLC). Configure the PPP encapsulation on the Serial1/0 interfaces on R1 and R3.

R1:

```
R1#show run interface s1/0
Building configuration...

Current configuration : 203 bytes
!
interface Serial1/0
 ip address 172.16.13.1 255.255.255.0
 encapsulation ppp
 no peer neighbor-route
 end

R1#
```

R3:

```
R3#show run interface s1/0
Building configuration...

Current configuration : 225 bytes
!
interface Serial1/0
 ip address 172.16.13.3 255.255.255.0
 encapsulation ppp
 no peer neighbor-route
 end

R3#
```

Verification:

Verify the PPP link between R1 and R3. Here is an example from R3:


```
R3#show users
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00   ciers1v5-ca-lab02-sc, SJ

  Interface    User               Mode         Idle     Peer Address
  Se1/0                           Sync PPP     00:00:02 172.16.13.1

R3#
```

Verify the connected PPP link entry in the routing table on R1:

```
R1#sho ip route | inc 172.16.13.*\/32
L        172.16.13.1/32 is directly connected, Serial1/0
R1#
```

Note that, because the **no peer neighbor-route** command is configured on the PPP interface, R1 does not show the host entry 172.16.13.3/32 for the other end of the PPP link that is connected to the Serial1/0 interface of R3.

Verify the connected PPP link entry in the routing table on R3:

```
R3#sho ip route | inc 172.16.13.*\/32
L        172.16.13.3/32 is directly connected, Serial1/0
R3#
```

Verify the IP connectivity over the PPP link between R1 and R3. Here is an example from R1:

```
R1#ping 172.16.13.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms
R1#
```

Note	To obtain a configuration view of the tasks in this and following sections, access the Mentor Guide engine. You can retrieve the available commands by querying the Mentor Guide engine via “Command Line" field.

2. Switch Configuration Section

Issue: Configure the VLANs for all Ethernet segments involved in this scenario. All VLANs must be configured consistently on switches SW1 and SW2. Configure only necessary VLANs on SW3 and SW4. Set the VLAN Trunking Protocol (VTP) mode to transparent and the domain name to “lab2” on all four switches.

Solution:

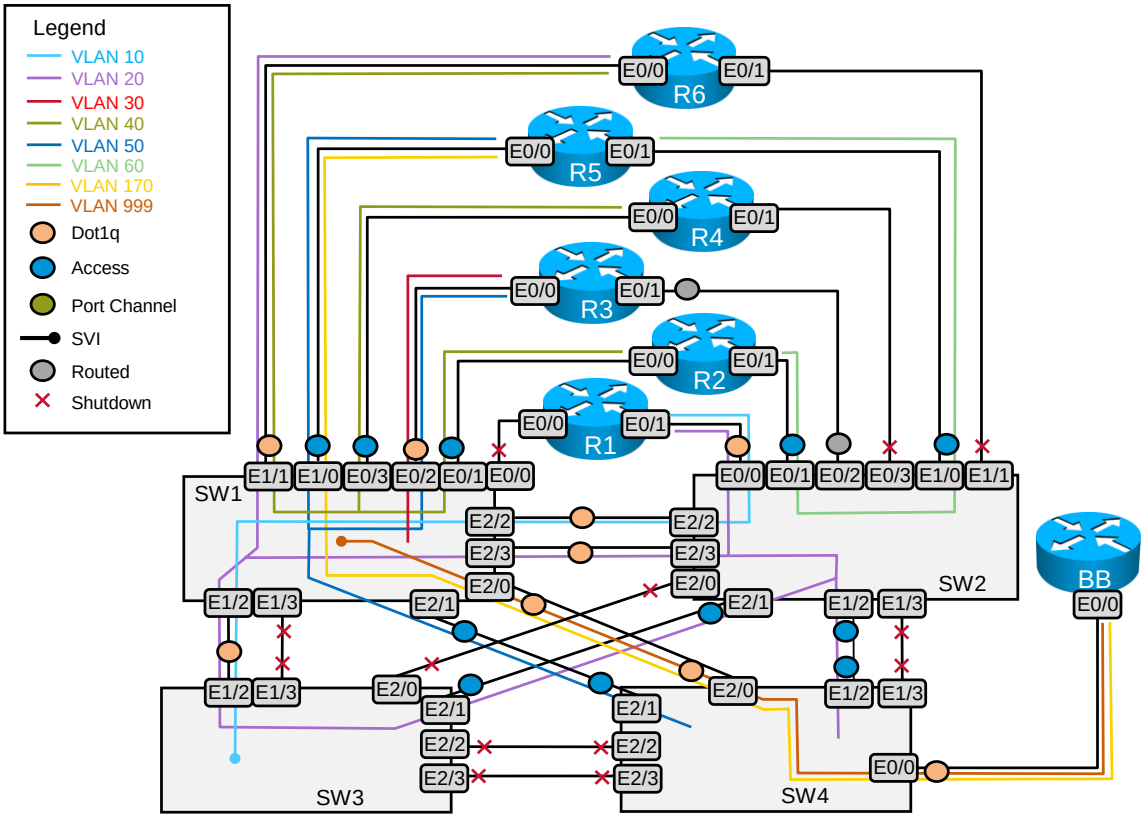
To ensure a thorough understanding of the Layer 2 topology, many candidates find it helpful to create a VLAN propagation diagram as shown in the example. Study the “VLANs” table, “Switch-to-Router Connections” table, “Switch-to-Switch Connections” table, the IGP diagrams, and the other section requirements, and then carefully document each connection on a copy of the physical layer diagram. With practice, you can create a diagram quickly and find it to be a valuable tool.

All the switches must be in transparent mode, so you must ensure VLAN consistency on SW1 and SW2 manually. You can configure the VLANs on one switch and then copy and paste them into the other. Spell the VLAN names correctly, and match the letter case. Use the VLAN Propagation diagram to determine which VLANs must be created on SW3 and SW4; only VLAN 10 and VLAN 20 must be created on SW3, and only VLANs 20, 50, 170, and 999 must be created on SW4.

Verification:

Before moving on to switching optimization techniques, complete your basic switching configuration, as required. Verify that all Layer 3 interfaces in the same VLAN can ping each other. Compare the output of various **show** commands in your VLAN Propagation diagram. You could use **show vlan brief** to verify VLAN names and access port assignments. Use **show interfaces trunk** to verify trunk port encapsulation and allowed VLANs.

VLAN Distribution



Issue: Tune Inter-Switch Links (ISLs). Allow only VLAN 20 and VLAN 10 on the link between ports 1/2 of SW1 and SW3. Allow only VLAN 170 and VLAN 999 on the link between ports 2/0 of SW1 and SW4.

Solution:

Use the command **switchport trunk allowed vlan,** to control which VLANs are allowed on these trunks. Configure the commands on both sides of each link. Verify with the command **show interfaces trunk**.

Issue: Assign the IP address 172.16.1.1/24 without assigning it to logical or physical Ethernet interfaces on R1**.**

Solution:

Fulfill this requirement by configuring integrated routing and bridging (IRB) on router R1. Assign the Ethernet subinterface on R1 that is associated with VLAN 10 to a bridge group. Then, enable IRB and create a bridge-group virtual interface (BVI) on R1 using the 172.16.1.0/24 subnet. Do not assign an IP address to the VLAN 10 Ethernet subinterface. Finally, enter the global configuration command **bridge 1 route ip** on R1 to allow bridged IP packets to be forwarded to the BVI.

Configuration and verification:

R1:

```
bridge irb
!
interface Ethernet0/1.10
 encapsulation dot1Q 10
 no ip redirects
 bridge-group 1
!
interface BVI1
 ip address 172.16.1.1 255.255.255.0
!
bridge 1 protocol ieee
bridge 1 route ip
```

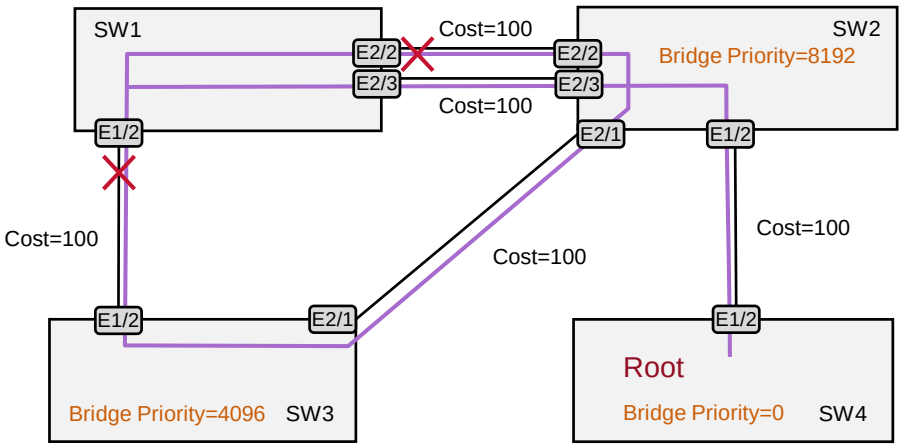
The network 172.16.1.0/24 is listed as connected via interface BVI1, which is associated with the bridge group combining the logical Ethernet interfaces.

```
R1#show ip route | inc 172.16.1.0/24
C        172.16.1.0/24 is directly connected, BVI1
R1#

R1#show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  172.16.1.10           197   000a.b7f7.7900  ARPA   BVI1
Internet  172.16.1.1              -   00d0.ba8b.0021  ARPA   BVI1
Internet  172.16.1.2            198   0010.7b3b.74de  ARPA   BVI1
R1#
```

Issue: On VLAN 20, set bridge priority to 4096 on SW3, 8192 on SW2, and 0 on SW4. SW3 must be the root bridge.

Solution:

A good first step in most spanning-tree optimization tasks is to create a diagram of the VLAN that shows the root bridge, the ports in the VLAN, and their spanning-tree states. Here is an example of a diagram after configuration of the required bridge priorities.



SW4 is going to be elected as a root bridge because it has the lowest priority; however, the task requires that SW3 be the root bridge for VLAN20. Your challenge is to find a way to override the normal root bridge election process.

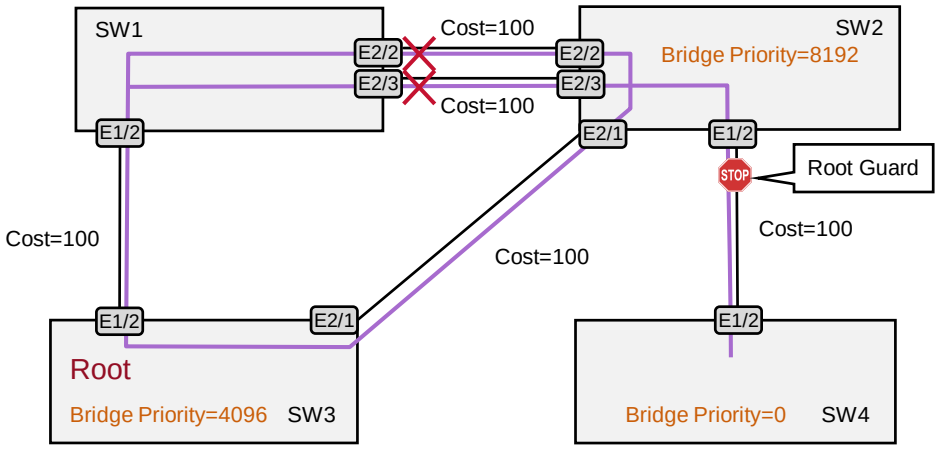
One solution is to configure root guard on the incoming interface of SW2, facing SW4:

```
SW2(config)#int E1/2
SW2(config-if)#spanning-tree guard root
SW2(config-if)#
23:43:48:  %SPANTREE-2-ROOTGUARD_CONFIG_CHANGE:  Root  guard  enabled  on  port
Ethernet1/2.
23:43:48:  %SPANTREE-2-ROOTGUARD_BLOCK:  Root  guard  blocking  port  Ethernet1/2  on
VLAN0020.

SW2#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID     Priority    4116
              Address     aabb.cc00.0900
              Cost        100
              Port        10 (Ethernet2/1)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    8212   (priority 8192 sys-id-ext 20)
              Address     aabb.cc00.0800
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
```



```
------------------- ---- --- --------- -------- --------------------------------
Et0/0               Desg FWD 100       128.1    Shr
Et1/2               Desg BKN*100       128.7    P2p *ROOT_Inc
Et2/1               Root FWD 100       128.10   P2p
Et2/2               Desg FWD 100       128.11   P2p
Et2/3               Desg FWD 100       128.12   P2p


SW2#
```

The root guard feature blocks a port rather than allowing it to become a root port. With E1/2 on SW2 blocked, SW3 will win the root bridge election because of its lowest priority. The blocking interfaces will be moved to new interfaces, as reflected in the following diagram.



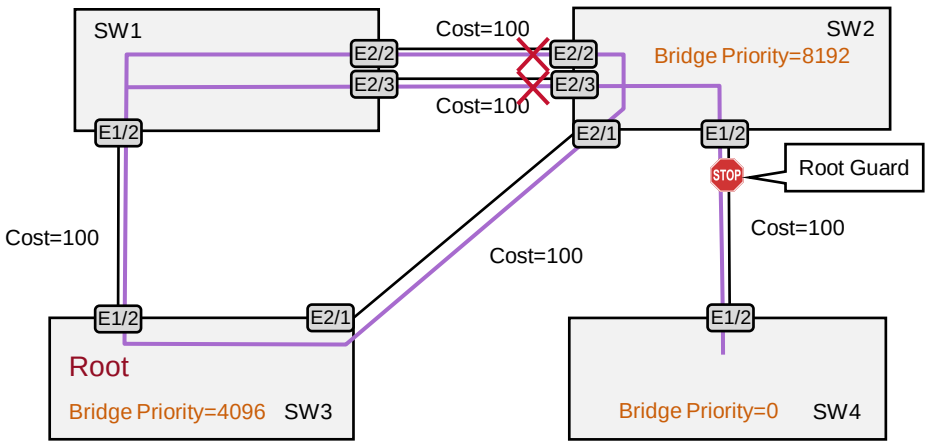

```
SW1#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID     Priority    4116
              Address     aabb.cc00.0900
              Cost        100
              Port        7 (Ethernet1/2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32788  (priority 32768 sys-id-ext 20)
              Address     aabb.cc00.0700
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time  300 sec
```

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et0/2               Desg FWD 100       128.3    Shr
Et1/0               Desg FWD 100       128.5    Shr
Et1/1               Desg FWD 100       128.6    Shr
Et1/2               Root FWD 100       128.7    P2p
Et2/2               Altn BLK 100       128.11   P2p
Et2/3               Altn BLK 100       128.12   P2p


SW1#
```

Issue: User traffic on VLAN 20 must always be forwarded via SW3. SW1 must not have any blocking interfaces on VLAN 20. You are allowed to change the cost to 101 only on one link.

Solution:

The previous diagram shows that the blocking interfaces are on SW1. We can move these blocked ports to SW2 by increasing the cost to 101 for VLAN 20 on port 2/1 of SW2. With a root path cost of 100 on SW1 and a root path cost of 101 on SW2, SW1 ports 2/2 and 2/3 will be elected designated ports, and the SW2 ports will block.




```
SW2#conf t
```



```
Enter configuration commands, one per line.  End with CNTL/Z.
SW2(config)#interface Ethernet2/1
SW2(config-if)#spanning-tree vlan 20 cost 101
SW2(config-if)#end
SW2#
SW2#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    4116
             Address     aabb.cc00.0900
             Cost        101
             Port        10 (Ethernet2/1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    8212   (priority 8192 sys-id-ext 20)
             Address     aabb.cc00.0800
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  15  sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et0/0               Desg FWD 100       128.1    Shr
Et1/2               Desg BKN*100       128.7    P2p *ROOT_Inc
Et2/1               Root FWD 101       128.10   P2p
Et2/2               Altn BLK 100       128.11   P2p
Et2/3               Altn BLK 100       128.12   P2p


SW2#

SW1#show spanning-tree vlan 20

VLAN0020
  Spanning tree enabled protocol ieee
  Root ID    Priority    4116
             Address     aabb.cc00.0900
             Cost        100
             Port        7 (Ethernet1/2)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32788  (priority 32768 sys-id-ext 20)
             Address     aabb.cc00.0700
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et0/2               Desg FWD 100       128.3    Shr
Et1/0               Desg FWD 100       128.5    Shr
Et1/1               Desg FWD 100       128.6    Shr
Et1/2               Root FWD 100       128.7    P2p
Et2/2               Desg FWD 100       128.11   P2p
Et2/3               Desg FWD 100       128.12   P2p


SW1#
```

3. IPv4 OSPF Section

Note	Configure all Open Shortest Path First (OSPF) routers for network 172.16.X.0/24 with the OSPF process ID (PID) 100. OSPF PIDs must be different for 172.16.X.0/24 and 10.X.X.0/24. Use your IGP and MPLS diagrams to help guide configuration.

Issue: The OSPF network type for the DMVPN subnet must be the broadcast type.

Solution:

The 172.16.124.0/24 subnet is configured on a DMVPN hub-and-spoke topology. Because all OSPF packets have a Time to Live (TTL)=1, OSPF spoke routers will never communicate directly with other spoke routers. Therefore, no spoke routers can become either a designated router (DR) or a backup designated router (BDR). To ensure that this never happens, set the OSPF priority to 0 on the spoke routers R2 and R4. This configuration is made at the DMVPN interface level.

Issue: The OSPF network type for the PPP subnet must be the non-broadcast type. R3 is the DR on the PPP subnet. R1 should not participate in DR or BDR election on the link between R1 and R3.

Reminder:

OSPF devices configured for the non-broadcast network type will go through the DR or BDR election procedure. Also, OSPF packets will be generated with an IP unicast destination, so neighbor statements will be required.

Solution:

Configure OSPF priority 0 under the point-to-point interface on the subnet 172.16.13.0/24 on R1. With priority 0, R1 will not have a chance to become the DR. Configure the OSPF neighbor statement on R3.

Issue: On R3, configure the subnet 172.16.50.0/24 on interface VLAN 30 advertised through OSPF without including it in any OSPF areas. It must be viewed as a type 1 route by OSPF.

Solution:

Configure **redistribute connected** on R3 to inject the VLAN 30 subnet into OSPF without using an OSPF network command. Whenever you redistribute connected prefixes, consider applying either a distribute list or route map to inject only the connected prefixes that you intend to redistribute.

External routes are redistributed into OSPF as external type 2 networks by default. You can override this default by setting type 1 in the route map referenced in the **redistribute connected** statement. Here is an example:

```
route-map Connected-->OSPF permit 10
 match ip address Connected-->OSPF-E1
 set metric-type type-1
```

Issue: Configure a Generic Routing Encapsulation (GRE) tunnel between R5 and R3, and use only two IP addresses, 172.16.35.5 and 172.16.35.3, to accomplish this task.

Solution:

The two specified IP addresses are assigned to the VLAN 50 interfaces on R3 and R5, so they will be used as the tunnel source and destination addresses. The tunnel can use these addresses as its endpoints by referencing them in the **ip unnumbered** command.

R5	R3
interface Tunnel305 ip unnumbered Ethernet0/0.50 tunnel source 172.16.35.5 tunnel destination 172.16.35.3 ! interface Ethernet0/0.50 encapsulation dot1Q 50 ip address 172.16.35.5 255.255.255.0	interface Tunnel305 ip unnumbered Ethernet0/0.50 tunnel source 172.16.35.3 tunnel destination 172.16.35.5 ! interface Ethernet0/0.50 encapsulation dot1Q 50 ip address 172.16.35.3 255.255.255.0

The tunnel mode is the GRE by default, so no specific configuration is required.

Issue: Place the tunnel link into OSPF Area 0. Form an OSPF adjacency on the tunnel link only. The **show** output on the R5 interface representing VLAN 50 should display a result similar to this:

```
R5#show ip ospf int ethernet 0/0.50
%OSPF: OSPF not enabled on Ethernet0/0.50
```

Solution:

Because the tunnel IP address is the same as the corresponding Ethernet VLAN50 IP address, the network statement **network 172.16.35.0 0.0.0.255 area** under the OSPF process will add both the Ethernet interface and the tunnel interface into the OSPF database. To add just the tunnel into the OSPF process, use the interface configuration mode command **ip ospf area.**

Configuration and verification:

R5:

```
interface Tunnel305
 ip unnumbered Ethernet0/0.50
 ip ospf 100 area 0
 tunnel source 172.16.35.5
 tunnel destination 172.16.35.3

R5#show ip ospf interface tunnel305
Tunnel305 is up, line protocol is up
  Interface is unnumbered. Using address of Ethernet0/0.50 (172.16.35.5), Area 0,
Attached via Interface Enable
  Process ID 100, Router ID 172.16.105.1, Network Type POINT_TO_POINT, Cost: 1000
  Topology-MTID    Cost    Disabled    Shutdown      Topology Name
        0           1000      no          no            Base
  Enabled by interface config, including secondary ip addresses
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 1/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 172.16.103.1
```

```
  Suppress hello for 0 neighbor(s)
R5#
```

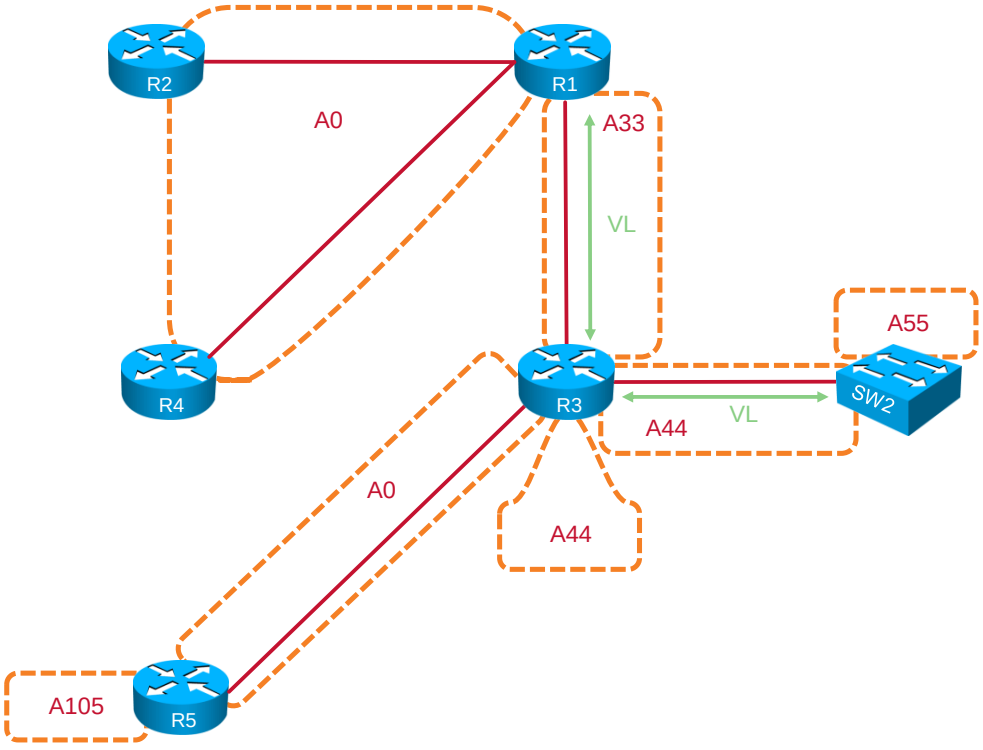
Here, you see that OSPF is not activated on the VLAN 50 interface, but only on the tunnel.

```
R5#show ip ospf interface ethernet0/0.50
%OSPF: OSPF not enabled on Ethernet0/0.50


R5#show ip ospf neigh
Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.103.1      0   FULL/  -        00:00:35    172.16.35.3     Tunnel305
R5#
```

Issue: Verify OSPF connectivity.

Solution:

With just the configuration discussed so far, you are likely to see connectivity problems with the OSPF domain. The two Areas 0 are separated by Area 33, and Area 55 is separated from the backbone, as shown. The solution in each case is a virtual link.



Issue: Loopback networks are advertised as /32 if they are advertised as OSPF interarea prefixes.

Solution:

Default OSPF behavior advertises loopback interfaces as host routes and, therefore, advertises them as /32. This is because the default network type LOOPBACK is assigned to these interfaces by OSPF. This is not inherently undesirable; however, the Restrictions and Goals section of the lab scenario specifies that loopbacks be advertised with their original masks. One solution is to configure **ip ospf network point-to-point** under the loopback interface. In other contexts, you might also consider creating an interarea summary or adding the loopback into the OSPF process as an external route.

4. IPv4 EIGRP Section

Issue: On SW1, configure the Enhanced Interior Gateway Routing Protocol (EIGRP) autonomous system (AS) 80 and AS1. Also configure an EIGRP neighbor relationship with backbone router BB2 AS9999. See the IPv4 IGP diagram.

Solution:

Configure three EIGRP routing processes: **router eigrp 80**, **router eigrp 1**, and **router eigrp 9999**. Add the network 172.16.1.0/24 in both EIGRP 80 and EIGRP 1 processes. Add the network 173.35.33.0/24 to the EIGRP 9999 process.

Issue: Configure EIGRP AS1 between SW1 and SW3. Configure EIGRP AS80 between SW1 and R1. Advertise the network 172.16.107.0/24 in EIGRP AS1 and the network 172.16.110.0/24 in EIGRP AS9999.

Solution:

Configure routing processes **router eigrp 1** on SW3 and **router eigrp 80** on R1. Add the command **network 172.16.1.0 0.0.0.255** on R1 and SW3. Add the network statements **network 172.16.110.0 0.0.0.255** and **network 172.16.107.0 0.0.0.255** under the respective EIGRP processes on SW1 (AS9999) and SW3 (AS1).

Issue: You might experience an EIGRP neighbor relationship setup problem with BB2. You check the AS numbers and IP addresses and detect no problems. You can ping the backbone router, but your EIGRP speaker does not form an adjacency with BB2.

Solution:

This is an example of a discovery task that you might need to perform in a lab. Set up a debugging session on the backbone link and try to interpret packets that you receive from the BB2 IP address, 173.35.33.100. EIGRP packets are encapsulated into IP protocol number 88 packets. Look at the destination of the packet. By default, EIGRP uses 224.0.0.10 for hello packet addressing. This default was overridden on the backbone router with the **neighbor** command. The backbone router is expecting unicast communications. You must specify the neighbor statement on your end as well:

```
SW1#deb condition int vlan 999
Condition 1 set
SW1#deb ip packet det
IP packet debugging is on (detailed)
SW1#
02:29:20: datagramsize=60, IP 0: s=173.35.33.1 (local), d=224.0.0.10 (Vlan999),
totlen 60, fragment 0, fo 0, sending broad/multicast, proto=88
02:29:22:  datagramsize=78,  IP  0:  s=173.35.33.100  (Vlan999),  d=173.35.33.1
(Vlan999), totlen 60, fragment 0, fo 0, rcvd 3, proto=88
```

The green output represents the EIGRP packet generated at your end if you do not have a neighbor statement. Notice that the destination is 224.0.0.10. The yellow output represents the EIGRP packet received from the backbone. Notice that the destination is unicast. After you apply the neighbor statement, the EIGRP speaker changes the destination to unicast.

```
SW1(config)#router eigrp 9999
SW1(config-router)#neighbor 173.35.33.100 vlan 999
SW1(config-router)#


SW1#
02:33:34: datagramsize=60, IP 0: s=173.35.33.1 (local), d=173.35.33.100 (Vlan999),
totlen 60, fragment 0, fo 0, sending, proto=88
```

And the adjacency is formed:

```
SW1#show ip eigrp 9999 neighbors
EIGRP-IPv4 Neighbors for AS(9999)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   173.35.33.100           Vl999                    13 11:39:36   17   102  0  2
SW1#
```

EIGRP neighbor relationships and interfaces verification:

SW1 should have three neighbors: R1 (172.16.1.1), SW3 (172.16.1.2), and BB2 (173.35.33.100):

```
SW1#show ip eigrp 80 neighbors
EIGRP-IPv4 Neighbors for AS(80)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   172.16.1.1              Vl110                  14 20:30:45    5   100  0  7
SW1#
SW1#show ip eigrp 9999 neighbors
EIGRP-IPv4 Neighbors for AS(9999)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   173.35.33.100           Vl999                  10 20:30:41    8   100  0  2
SW1#
SW1#show ip eigrp 1 neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   172.16.1.2              Vl110                  12 20:30:44    5   100  0  6
SW1#
```

EIGRP-aware interfaces on SW1:

```
SW1#sh ip eigrp int
EIGRP-IPv4 Interfaces for AS(80)
                              Xmit  Queue     PeerQ              Mean     Pacing  Time
Multicast    Pending
Interface               Peers  Un/Reliable  Un/Reliable  SRTT    Un/Reliable    Flow
Timer   Routes
Vl110                    1        0/0        0/0          336        0/0          1684
0
EIGRP-IPv4 Interfaces for AS(9999)
                              Xmit  Queue     PeerQ              Mean     Pacing  Time
Multicast    Pending
Interface               Peers  Un/Reliable  Un/Reliable  SRTT    Un/Reliable    Flow
Timer   Routes
Lo110                    0        0/0        0/0            0        0/0            0
0
Vl1999                   1        0/0        0/0           17        0/0           68
0
EIGRP-IPv4 Interfaces for AS(1)
```



```
                                Xmit  Queue     PeerQ              Mean     Pacing  Time
Multicast    Pending
Interface               Peers   Un/Reliable  Un/Reliable  SRTT    Un/Reliable    Flow
Timer   Routes
Vl110                    1        0/0        0/0           5       0/0            50
0
SW1#
```

R1 should have only one neighbor, SW1 (172.16.1.10):

```
R1#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(80)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   172.16.1.10             BV1                      13 07:46:25    1   100  0  10
R1#
```

Likewise, SW3 should have only one neighbor, SW1 (172.16.1.10):

```
SW3#sh ip eigrp nei
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
0   172.16.1.10             Vl10                     10 11:44:23   73   438  0  18
SW3#
SW3#sh ip eigrp int
EIGRP-IPv4 Interfaces for AS(1)
                                Xmit  Queue     PeerQ              Mean     Pacing  Time
Multicast    Pending
Interface               Peers   Un/Reliable  Un/Reliable  SRTT    Un/Reliable    Flow
Timer   Routes
Vl110                    1        0/0        0/0           73      0/0            360
0
Lo107                    0        0/0        0/0           0       0/0              0
0
SW3#
```

Issue: Allow only networks 192.168.2.0 and 192.168.6.0 to be accepted from BB2 via EIGRP AS999. Use a single entry in the access list. All other routers must possess the allowed prefixes in their respective routing tables.

Solution:

On SW1, configure an access list allowing only the two listed prefixes to be accepted by EIGRP. Apply the access list to a **distribute-list in** command referencing the VLAN 999 interface.

```
router eigrp 9999
 network 172.16.110.0 0.0.0.255
 network 173.35.33.0 0.0.0.255
 neighbor 173.35.33.100 Vlan999
 distribute-list BB2->EIGRP9999 in Vlan999
!
ip access-list standard BB2->EIGRP9999
 permit 192.168.2.0 0.0.4.255
```

Access list explanation:

1. We care about 192; therefore, the first byte of the wildcard is set to 0.
2. We care about 168; therefore, the second byte of the wildcard is set to 0.
3. The third octet is special. Here we expand 2 and 6 to binary to see what bits are common (green) and what bits can change (yellow):

```
2           =     00000010
6           =     00000110
-------------------------------------
Wildcard    =     00000100     = 4
```

We care about bits in the green zone and we do not care about yellow bits.

4. We do not care about the last octet.

These two allowed prefixes must be redistributed into EIGRP AS80 and AS1 on SW1 to propagate them to R1 and SW3. Also, EIGRP 80 needs to be redistributed into OSPF on R1 to propagate the prefixes to all other routers.

Issue: The EIGRP prefixes should not change their length while traversing through the boundaries of other major networks.

Solution:

Under the EIGRP routing processes, configure **no auto-summary** and later, under the Routing Information Protocol (RIP) process, perform the same action.

Issue: Do not advertise any prefixes to BB2. Do not use prefix-filtering techniques to accomplish this task.

Solution:

At first, this may appear to be a passive-interface configuration requirement. However, if you make an EIGRP interface passive, it will not transmit any EIGRP hellos. If no hello packets are generated, the EIGRP speaker will never form an adjacency with another EIGRP speaker. If no adjacency is formed, then the EIGRP speaker will not receive any routing updates. SW1 must receive updates from BB2. Therefore, configuring the passive-interface command is unacceptable. One potential solution is to configure a distribute list that denies all routes and apply the distribute list out through the VLAN 999 interface under the respective EIGRP process. However, this solution would be considered prefix filtering and is not permitted.

Another option to fulfill this configuration requirement is to configure **eigrp stub receive-only** under the EIGRP routing process 9999 on SW1. With this command, SW1 can be configured to silently listen to BB2 without advertising any updates to BB2. This solution is used in the master configuration of the Mentor Guide.

Issue: Perform the necessary route redistribution for this section on R1 and SW1.

Solution:

Perform the redistribution as follows:

- SW1: From EIGRP AS9999 into EIGRP AS80, propagate the AS9999 learned prefixes to R1 and beyond.
- SW1: From EIGRP AS9999 into EIGRP AS1, propagate the AS9999 learned prefixes to SW3.
- SW1: Mutual redistribution between EIGRP AS80 and EIGRP AS1 to exchange the networks between R1 and SW3.
- R1: Perform mutual redistribution between OSPF and EIGRP AS80 to exchange prefixes between the EIGRP domain and non-EIGRP domain.

5. IPv4 RIP

Issue: Do not broadcast or multicast RIP updates on VLAN40 for security reasons. Make sure each router is able to receive the RIP updates directly from the other RIP speakers.

Solution:

If you are instructed to neither broadcast nor multicast RIP updates, configure RIP to unicast its updates. To accomplish this, make the interface passive, and then configure neighbor statements for every device that needs to receive the RIP updates. To make sure that each router receives updates directly from the other RIP speakers, configure two neighbor statements on each VLAN40 RIP speaking router.

Issue: On R2 and R4, configure mutual redistribution between OSPF and RIP. Use the minimum nonzero metric in redistribution.

Solution:

You are instructed to perform mutual redistribution between RIP and OSPF at two redistribution points. Such a configuration requirement might cause much potential routing instability. When performing mutual redistribution at more than one point between two routing protocols, create a filter so that routing domains do not learn their own routes from external sources. This solution implements such a filter on R2 and R4.

First, create an access list that defines RIP domain routes:

```
ip access-list standard RIP-->OSPF
 permit 172.16.30.0
 permit 172.16.26.0
 permit 172.16.104.0
 permit 172.16.106.0
 permit 172.16.102.0
```

Next, create two route maps referencing this access list:

```
route-map RIP-->OSPF permit 10
 match ip address RIP-->OSPF
!
route-map OSPF-->RIP deny 10
 match ip address RIP-->OSPF

route-map OSPF-->RIP permit 20
```

Route map RIP: The route map named "RIP-->OSPF" is applied on the redistribution of RIP into OSPF. This route map permits only the RIP domain routes to be redistributed, implicitly denying any routes already known to OSPF.

Route map OSPF: The route map named "OSPF-->RIP" is applied on the redistribution of OSPF into RIP. The first statement denies the RIP domain routes from being redistributed back into RIP. The second statement permits everything else known by OSPF.

Configure a metric of 1 when redistributing into RIP and into OSPF, on both R2 and R4 to fulfill the minimum metric requirement. OSPF will accept routes with a metric set to zero; however, the task requires a nonzero metric.

Issue: The prefixes originated on R6 must be displayed in the routing table of R1 with the two next hops—172.16.124.2 and 172.16.124.4.

Solution:

To meet this requirement, make sure that both R2 and R4 are Autonomous System Boundary Routers (ASBRs) for the prefixes originated from R6. Therefore, both R2 and R4 must have these prefixes listed as RIP prefixes in their respective routing tables. By default, you are likely to find that only one of the routers is an OSPF ASBR; the other router shows the routes from R6 as OSPF external type 2 (OE2) routes and cannot redistribute them into OSPF.

The root of the problem is the difference in administrative distance between OSPF and RIP. If you redistribute on R2 first, for example, then R4 will have two sources for the loopback prefixes on R6: RIP and OSPF. Because the administrative distance of OSPF is lower than that of RIP, R4 will put the OSPF route in the database and will not be an ASBR for these routes. Use the **distance** command on both R2 and R4 so that they will prefer the RIP sources for routes in the RIP domain.

Here is the configuration for R2. It sets the administrative distance to 109 for routes from any RIP source that match the access list RIP distance.

```
router rip
 version 2
 redistribute ospf 100 metric 1 route-map OSPF-->RIP
 passive-interface default
 network 172.16.0.0
 neighbor 172.16.26.6
 neighbor 172.16.26.4
 distance 109 0.0.0.0 255.255.255.255 RIP-distance

ip access-list standard RIP-distance
 permit 172.16.30.0
 permit 172.16.104.0
 permit 172.16.106.0
```

A detailed paper on this topic called *A Scenario with Multiple Redistribution Points* is available on the Cisco Expert-Level Training for CCIE Routing and Switching Reference Library for further reading.

Verification:

```
R2#sh ip route rip | section (172.16.106|172.16.30)
R       172.16.30.0/24 [109/1] via 172.16.26.6, 00:00:19, Ethernet0/0
R       172.16.106.0/24 [109/1] via 172.16.26.6, 00:00:19, Ethernet0/0
R2#

R4#sh ip route rip | section (172.16.106|172.16.30)
R       172.16.30.0/24 [109/1] via 172.16.26.6, 00:00:13, Ethernet0/0
R       172.16.106.0/24 [109/1] via 172.16.26.6, 00:00:13, Ethernet0/0
…

R1#sh ip route | section (172.16.106|172.16.30)
O E2     172.16.30.0/24 [110/1] via 172.16.124.4, 11:24:52, Tunnel124
                        [110/1] via 172.16.124.2, 11:24:52, Tunnel124
O E2     172.16.106.0/24 [110/1] via 172.16.124.4, 11:24:52, Tunnel124
                         [110/1] via 172.16.124.2, 11:24:52, Tunnel124
R1#
```

Here is a Tool Command Language (Tcl) script that you can use to test for universal reachability on all devices:

```
tclsh
foreach address {
172.16.16.1
172.16.13.1
172.16.1.1
172.16.124.1
172.16.101.1
```



```
172.16.26.2
172.16.124.2
172.16.102.1

172.16.60.1
172.16.50.3
172.16.35.3
172.16.31.3
172.16.13.3
172.16.103.1

172.16.26.4
172.16.124.4
172.16.104.1

172.16.35.5
172.16.105.1

172.16.26.6
172.16.30.1
172.16.16.6
172.16.106.1

172.16.1.2
172.16.107.1

172.16.1.10
172.16.110.1

172.16.31.20
172.16.120.1
} {ping $address}
```

6. BGP Section

Issue: On R5, configure Border Gateway Protocol (BGP) AS1030 (your AS) to speak with AS9999 (backbone AS) configured on BB1. AS9999 advertises routes from 140.10.1.0/24 to 140.10.5.0/24. Allow only the networks 140.10.2.0/24, 140.10.3.0/24, 140.10.4.0/24, and 140.10.5.0/24 into your AS**.**

Solution:

You must have connectivity to BB. Configure VLAN 170, and assign the IP address 170.100.10.1 on the Ethernet interface of R5. Ping the IP address 170.100.10.110.

Configure either an access list or prefix list to allow only the four listed prefixes into AS1030. Because the range of addresses crosses a bit boundary, it requires at least two access lists or prefix list statements.

For example:

```
router bgp 1030
 bgp log-neighbor-changes
 neighbor 170.100.10.110 remote-as 9999
 neighbor 170.100.10.110 distribute-list BGP-140routes in
!
ip access-list standard BGP-140routes
 permit 140.10.2.0 0.0.1.0
 permit 140.10.4.0 0.0.1.0
```

The first permit statement allows two networks:

```
140.10.2.0
140.10.3.0
```

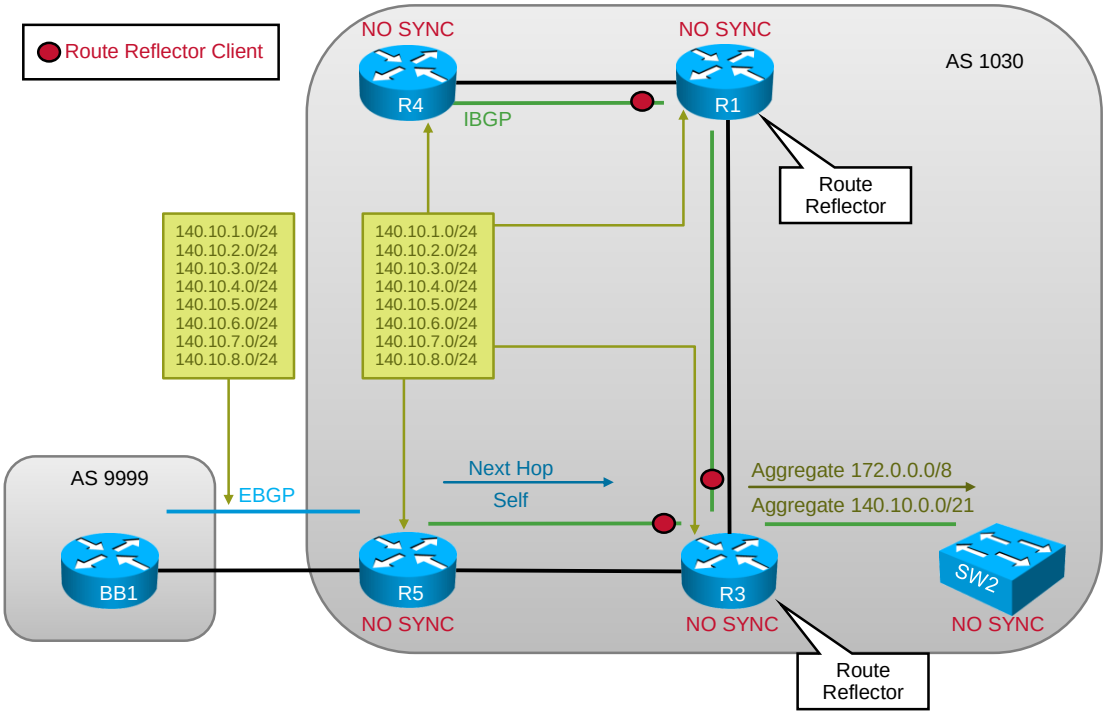
The second permit statement allows two networks:

```
140.10.4.0
140.10.5.0

R5#sh ip bgp
BGP table version is 5, local router ID is 172.16.105.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 140.10.2.0/24    170.100.10.110           0             0 9999 i
*> 140.10.3.0/24    170.100.10.110           0             0 9999 i
*> 140.10.4.0/24    170.100.10.110           0             0 9999 i
*> 140.10.5.0/24    170.100.10.110           0             0 9999 i
R5#
```

Issue: Configure the Internal Border Gateway Protocol (IBGP) peerings. Do not perform redistribution between BGP and any IGP, do not create any other IBGP peer relationships, and do not change the given BGP AS numbers. Do not introduce new BGP AS numbers.

Solution:

The following diagram represents the peer relationships required by the IPv4 portion of this scenario. Read the IP version 6 (IPv6) section of this answer key for the IPv6 BGP solutions.



Issue: Next-hop reachability and synchronization.

Solution:

When R3 receives the BGP updates from R5, they are marked as "I" BGP updates. The two most commonly encountered configuration requirements that need to be addressed when installing an IBGP learned update into a local routing table are: (1) the next-hop reachability of the IBGP-learned update, and (2) the issue of synchronization.

To fulfill this configuration requirement, remember that the link between R5 and BB1 is not included in your IGP. When R3 receives updates advertised by AS9999, it cannot reach the next-hop IP address because it is not advertised into the IGP used by R5. This is a classic example of the next-hop reachability issue. You can remedy this by configuring **next-hop-self** on the neighbor relationship configured on R5 to R3. With **next-hop-self** configured, R5 will set the next-hop IP address to itself for all BGP updates that it sends to R3.

When the next-hop reachability problem is solved, you must address the synchronization issue, which is based on the rule of synchronization, a provision to prevent traffic black-holing. The rule of synchronization states: A BGP speaker cannot advertise an IBGP-learned update to another BGP speaker until a matching entry for the IBGP learned update is in the local routing table from a source other than BGP. The rule of synchronization is commonly addressed in one of two ways: (1) Redistribute the BGP update into the IGP at the edge of the AS, or (2) Disable synchronization altogether. The scenario instructs you to not use any type of redistribution. Therefore, the best solution for addressing synchronization in this scenario is to make sure that it is disabled. You can verify the state of BGP synchronization with the **show ip protocols** command.

The requirements "do not create a full mesh" and "do not introduce and do not change the given BGP AS numbers" set you up for a route reflector solution.

Issue: Configure BGP on R3 so that you are sending an aggregate for 172.0.0.0/8 to SW2. Do not advertise any subnets configured on any interfaces into BGP. In addition, allow R3 to send to SW2 the most optimal aggregate for the networks allowed into your AS from AS9999. Suppress the longer matches to SW2. R1, R4, and R5 should not possess this aggregate but should possess its longer matches.

Solution:

Configure an aggregate for 172.0.0.0/8 on R3 with the summary-only option. To create an aggregate, you must have a longer match in the BGP table, yet you are not permitted to use any of the connected networks. You can avoid using a network statement to a connected interface by leaving **auto-summary** enabled and entering the classful network statement **network 172.16.0.0**.

The range of networks from 140.10.1.0/24 to 140.10.5.0/24 can be summarized to 140.10.0.0/21.

Proof:

Here is an example of the third byte in binary:

```
140.10.1.0/24 = 140.10.00000001.0/24
140.10.2.0/24 = 140.10.00000010.0/24
140.10.3.0/24 = 140.10.00000011.0/24
140.10.4.0/24 = 140.10.00000100.0/24
140.10.5.0/24 = 140.10.00000101.0/24
```

Yellow represents the common bits for this range of prefixes. The three right bits of the third byte can be different; therefore, 24 bits – 3 bits = 21 bits.

The following is the range of prefixes covered by 140.10.0.0/21:

```
140.10.0.0/24 = 140.10.00000000.0/21
140.10.1.0/24 = 140.10.00000001.0/21
140.10.2.0/24 = 140.10.00000010.0/21
140.10.3.0/24 = 140.10.00000011.0/21
140.10.4.0/24 = 140.10.00000100.0/21
```

```
140.10.5.0/24 = 140.10.00000101.0/21
140.10.6.0/24 = 140.10.00000110.0/21
140.10.7.0/24 = 140.10.00000111.0/21
```

Filtering strategy:

R5, R1, and R4 must possess the more specific prefixes of the 140.10 range only, and SW2 must receive only the aggregates. You could use a prefix list to allow only the aggregate routes to be advertised from R3 to R1 and R5, and use another prefix list to send only the specific routes to SW2.

```
ip prefix-list AGGREGATES seq 5 permit 140.10.0.0/21
ip prefix-list AGGREGATES seq 10 permit 172.0.0.0/8
!
ip prefix-list SPECIFIC seq 5 permit 140.10.2.0/23 ge 24 le 24
ip prefix-list SPECIFIC seq 10 permit 140.10.4.0/23 ge 24 le 24
```

Issue: On R3, associate R1 and R5 with a peer group. Use only the peer group to apply peering and filtering configurations to these routers.

Solution:

Here are the commands used in the master configuration of the Mentor Guide to configure the peer group. They are entered under the BGP process on R3:

```
neighbor ibgp-peer peer-group
neighbor 172.16.13.1 peer-group ibgp-peer
neighbor 172.16.35.5 peer-group ibgp-peer
```

When the peer group is created, you can use it to apply common peering and policy options. Here you see the **route-reflector-client** command and the filters associated with the peer group configuration on R3:

```
 neighbor ibgp-peer remote-as 1030
 neighbor ibgp-peer route-reflector-client
 neighbor ibgp-peer route-map STOP-AGGR out
```

Verification:

Here you see that SW2 has *only* the two aggregate prefixes:

```
SW2#sh ip bgp
[lines removed for brevity]

   Network          Next Hop            Metric LocPrf Weight Path
*>i 140.10.0.0/21    172.16.31.3                   100      0 i
*>i 172.0.0.0/8      172.16.31.3                   100      0 i
```

By contrast, R1, R4, and R5 have *only* the longer matches for the allowed prefixes learned from BB1:

```
R1#sh ip bgp
[lines removed for brevity]
   Network          Next Hop            Metric LocPrf Weight Path
*>i 140.10.2.0/24    172.16.35.5              0    100      0 9999 i
*>i 140.10.3.0/24    172.16.35.5              0    100      0 9999 i
*>i 140.10.4.0/24    172.16.35.5              0    100      0 9999 i
*>i 140.10.5.0/24    172.16.35.5              0    100      0 9999 i

R4#sh ip bgp
[lines removed for brevity]

   Network          Next Hop            Metric LocPrf Weight Path
*>i 140.10.2.0/24    172.16.35.5              0    100      0 9999 i
*>i 140.10.3.0/24    172.16.35.5              0    100      0 9999 i
*>i 140.10.4.0/24    172.16.35.5              0    100      0 9999 i
```



```
*>i 140.10.5.0/24    172.16.35.5              0    100      0 9999 i
R4#

R5#sh ip bgp
…
   Network          Next Hop            Metric LocPrf Weight Path
*> 140.10.2.0/24    170.100.10.110           0             0 9999 i
*> 140.10.3.0/24    170.100.10.110           0             0 9999 i
*> 140.10.4.0/24    170.100.10.110           0             0 9999 i
*> 140.10.5.0/24    170.100.10.110           0             0 9999 i
R5#
```

7. IPv6 Routing Section

Issue: Configure IPv6 addresses on the R4 and R2.

Solution:

1. Configure IPv6 addresses and IPv6 OSPF between R2 and R6, and place the IPv6 loopback networks in their respective areas:

R2	R6
interface Loopback102 ip address 172.16.102.1 255.255.255.0 ipv6 address 2102::1/122 ipv6 ospf network point-to-point ipv6 ospf 1 area 102 ! interface Ethernet0/0 ip address 172.16.26.2 255.255.255.0 ipv6 address 2026::2/122 ipv6 ospf 1 area 0 ! ipv6 router ospf 1	interface Loopback106 ip address 172.16.106.1 255.255.255.0 ipv6 address 2106::1/122 ipv6 ospf network point-to-point ipv6 ospf 1 area 106 ! interface Ethernet0/0.40 encapsulation dot1Q 40 ip address 172.16.26.6 255.255.255.0 ipv6 address 2026::6/122 ipv6 ospf 1 area 0 ! ipv6 router ospf 1

2. Verify the connectivity and the OSPF neighbors.

```
R6#ping 2026::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2026::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R6#

R6#show ipv6 ospf neighbor
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
172.16.102.1      1   FULL/BDR        00:00:34    8               Ethernet0/0.40
```

You can verify the IPv6 connectivity by using a Tcl script:

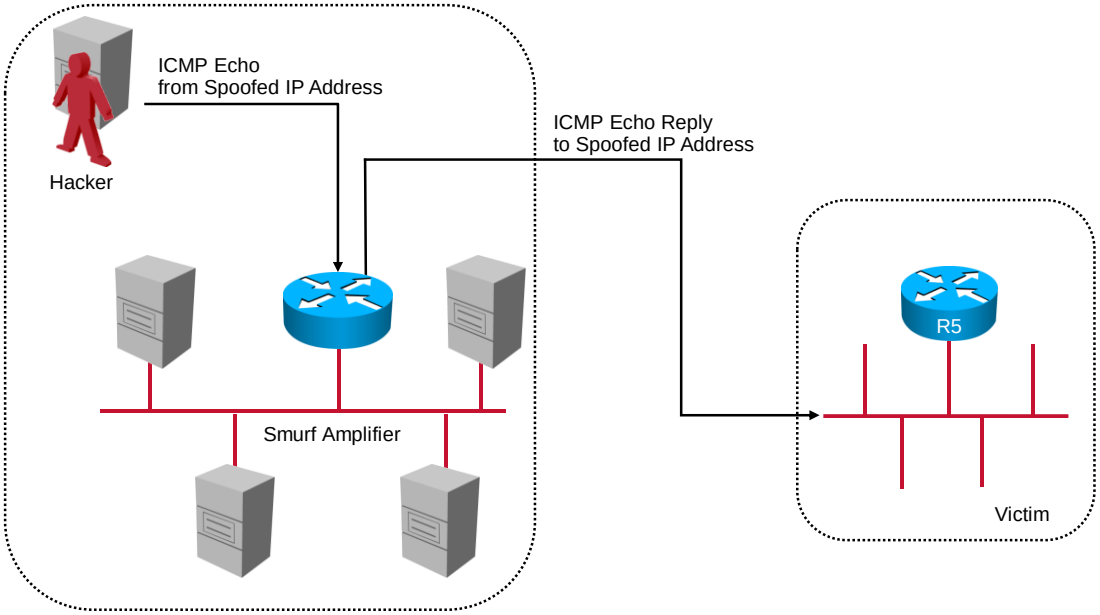
```
tclsh
foreach address {
2026::2
2102::1
2026::6
2106::1
} {ping $address}
```

8. Security Section

Issue: Your network is a victim of a smurf attack and is connected to the outside world through VLAN170. Provide a method to detect the IP addresses of amplifiers so that your network administrator can contact the ISPs who are in charge of these IP address blocks. Apply the configuration on R5.

Solution:

The following diagram shows the smurf denial of service (DoS) attack traffic flows.



A hacker uses spoofed IP addresses to generate Internet Control Message Protocol (ICMP) echo packets to a directed broadcast address. The hosts on the amplifier segment will process the broadcast ping and will send ICMP echo-reply packets to the victim network. The victim network will experience a large volume of ICMP echo-reply traffic on the ingress interface. Configure an access list to match ICMP echo-reply packets with the **log-input** keyword and apply it inbound on the ingress interface.

```
interface Ethernet0/0.170
 encapsulation dot1Q 170
 ip address 170.100.10.1 255.255.255.0
 ip access-group SMURF in
!
ip access-list extended SMURF
permit icmp any any echo-reply log-input
 permit ip any any
```

The access list should permit ICMP echo-reply traffic and all other types of IP traffic. Do not drop the ICMP echo-reply packets, because you want to use ping for maintenance and troubleshooting. Instead of dropping

packets, rate-limit the ICMP echo-reply traffic. See the quality of service (QoS) section of the answer key for more information.

The keyword **log-input** causes the router to log information about packets that match the list entry. For example:

```
1d23h: %SEC-6-IPACCESSLOGDP: list SMURF permitted icmp 140.10.2.1 (Ethernet0/0.170
0010.7b3c.47b7) -> 170.100.10.48 (0/0), 30 packets
```

The yellow field displays the source IP address of the ICMP echo-reply packet. It provides you with information about where the ICMP echo-reply packets are coming from as required.

Issue: SW1 must be able to ping all interfaces in your network. However, allow Telnet access to SW1 only from the management loopback interface of R3 (172.16.103.1). Do not apply the solution on an interface basis.

Solution:

The first part of this requirement was met in earlier sections. The scenario indirectly prohibits the **access group** interface-level command. It does not permit the VLAN map feature in the security section. To accomplish this task, create an access list, permitting only the R3 loopback interface. Then, apply the access list under the "line vty" mode with the command **access-class 1 in**.

9. QoS Section

Issue: Because of the smurf attack, your VLAN170 link is becoming saturated. Limit attack bandwidth consumption to 128 kb/s with the minimal burst and excess burst buffers. Other types of traffic should not be affected by your solution. Do not use the committed access rate (CAR) method to accomplish this task. Apply the configuration on R5 on the interface connected to the backbone.

Solution:

The CAR method is not allowed in this task. You can use the Cisco Modular QoS CLI (MQC) policing method instead, on the incoming traffic from the backbone. When configuring the MQC policing method, make sure that you use a method that explicitly specifies both the committed burst size (Bc) and excess burst (Be) values.

The following configuration will limit the incoming ICMP echo-reply flow to 128 kb/s. All other traffic types will not be affected because they will not match the classification criteria.

```
interface Ethernet0/0.170
 encapsulation dot1Q 170
 ip address 170.100.10.1 255.255.255.0
 ip access-group SMURF in
 no ip redirects
 service-policy input SMURF-RATE-LIMIT
!
class-map match-all SMURF-class
  match access-group name ECHO-REPLY
!
 policy-map SMURF-RATE-LIMIT
  class SMURF-class
   police cir 128000 bc 1000 be 1000
     conform-action transmit
     exceed-action drop
!
ip access-list extended ECHO-REPLY
 permit icmp any any echo-reply
```

Note that the Bc and Be values are set to 1000, which is the minimum value on the Cisco IOS Software used to write this answer key. These values might be different on the other Cisco IOS Software releases.

Verification:

```
R5# show policy-map interface

Ethernet0/0.170

  Service-policy input: SMURF-RATE-LIMIT

    Class-map: SMURF-class (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ECHO-REPLY
      police:
          cir 128000 bps, bc 1000 bytes, be 1000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps , violated 0000 bps


    Class-map: class-default (match-any)
      91 packets, 19745 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
R5#

R5#show access-lists ECHO-REPLY
Extended IP access list ECHO-REPLY
    10 permit icmp any any echo-reply
R5#
```

10. DHCP Section

Issue: Assign an IP address through DHCP based on a configured, sent client ID.

Solution:

The configurable DHCP client feature provides the flexibility to include various configuration options for the DHCP client. A DHCP client is defined as an Internet host using DHCP to obtain configuration parameters such as an IP address.

Issue: Only the 172.16.16.1 address should be assigned to R1. R6 should lease the IP address only if R1 is identified as “lab2.”

Solution:

Configure option 6. This option is used by DHCP clients to specify their unique identifier. DHCP servers use this value to index their database of address bindings. This value is expected to be unique for all clients in an administrative domain. Value “lab2” is used in this scenario with the command **ip dhcp client client-id.**

Issue: The lease duration of IP address 172.16.16.1 must be set to 345600 seconds.

Solution:

Configure lease duration in the DHCP pool on R6. The seconds are not an option in the **lease** command syntax; therefore, you must convert 345600 seconds to days:

345600 seconds / 60 = 5760 minutes
5760 minutes / 60 = 96 hours
96 hours / 24 = 4 days

Configuration and verification:

Configure the DHCP-configurable client on the VLAN 20 interface on R1:

```
interface Ethernet0/1.20
 encapsulation dot1Q 20
 ip dhcp client client-id ascii lab2
 ip address dhcp
```

Configure the DHCP pool on R6:

```
ip dhcp pool VLAN60
   host 172.16.16.1 255.255.255.0
   client-identifier 006c.6162.32
   lease 4
```

The DHCP pool **client-identifier** command requires a hexadecimal dump of the client ID ASCII string value. The quickest way to get this number is to use the command **debug ip dhcp server packet** on R6 after the client side is configured:

```
*Oct 28 14:08:51.729: DHCPD: DHCPREQUEST received from client 006c.6162.32.
```

Verify the lease on R1:

```
R1#sh dhcp lease
Temp IP addr: 172.16.16.1  for peer on Interface: Ethernet0/1.20
Temp  sub net mask: 255.255.255.0
   DHCP Lease server: 172.16.16.6, state: 3 Bound
   DHCP transaction id: 87
   Lease: 345600 secs,  Renewal: 172800 secs,  Rebind: 302400 secs
   Next timer fires after: 1d23h
   Retry count: 0   Client-ID: lab2
   Client-ID hex dump: 6C616232
   Hostname: R1
R1#
```

11. Address Administration Section

Issue: Configure the 1.1.1.3/24 address on the R3 Ethernet0/1 interface without changing any pre-existing IP addresses. Do not advertise 1.1.1.0/24 in any routing protocol. Use a portion of the address space of the R3 Ethernet0/1 primary subnet for the communications from network 1.1.1.0/24 to all other networks except for the networks connected to SW2.

The 1.1.1.3 address assigned to the R3 Ethernet interface will be assigned as a secondary IP address. This 1.1.1.0/24 address will be the Network Address Translation (NAT) inside address. The primary IP address of the Ethernet interface will be the NAT outside address.

Configuration:

On R3, perform the following configuration steps:

```
interface Ethernet0/1
 ip address 1.1.1.3 255.255.255.0 secondary
 ip address 172.16.31.3 255.255.255.0
 ip nat inside
!
interface Serial1/0
 ip address 172.16.13.3 255.255.255.0
 ip nat outside
!
interface Ethernet0/0.30
 encapsulation dot1Q 30
 ip address 172.16.50.3 255.255.255.0
ip nat outside
!
interface Ethernet0/0.50
 encapsulation dot1Q 50
 ip address 172.16.35.3 255.255.255.0
 ip nat outside
!
ip nat inside source list 50 interface Ethernet0/1 overload
!
access-list 50 permit 1.1.1.0 0.0.0.255
```

Verification:

Perform the extended ping from the inside network to outside, and check the NAT table. Notice that the inside (private) address is translated:

```
R3#ping 172.16.124.1 source 1.1.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.124.1, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 7/8/9 ms
R3#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 172.16.31.3:2     1.1.1.3:2          172.16.124.1:2     172.16.124.1:2
R3#
```

12. Gateway Redundancy Section

Issue: Imaginary workstations located on the VLAN 40 172.16.26.0/24 subnet are configured for the default gateway 172.16.26.1. R2 should be chosen as a preferred gateway over R6.

Solution:

This task is giving you an important piece of information, the virtual gateway IP address for the gateway redundancy protocol.

Issue: Choose the IETF standard protocol, using the multicast group address 224.0.0.18 to accomplish this section**.**

Solution:

Configure basic Virtual Router Redundancy Protocol (VRRP) on the R2 and R6 Ethernet interfaces with the command **vrrp 1 ip 172.16.26.1**., and assign the VRRP priority on R2 to a value higher than the priority value on R6. For example, configure the **vrrp 1 priority 150** command on R2:

R2:

```
interface Ethernet0/0
 ip address 172.16.26.2 255.255.255.0
 vrrp 1 ip 172.16.26.1
 vrrp 1 priority 150
```

R6:

```
interface Ethernet0/0.40
 encapsulation dot1Q 40
 ip address 172.16.26.6 255.255.255.0
 vrrp 1 ip 172.16.26.1
```

Verify the VRRP output on R2 and R6:

```
R2#show vrrp
Ethernet0/0 - Group 1
  State is Master
  Virtual IP address is 172.16.26.1
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 150
  Master Router is 172.16.26.2 (local), priority is 150
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.414 sec
R2#

R6#show vrrp
Ethernet0/0.40 - Group 1
  State is Backup
  Virtual IP address is 172.16.26.1
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 100
  Master Router is 172.16.26.2, priority is 150
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.609 sec (expires in 2.809 sec)
R6#
```

13. Multicast Configuration Section

Issue: Announce the shared root without the use of any dense groups or static configurations.

Solution:

Because a shared root is involved, this configuration requirement is a sparse mode configuration. When you configure sparse mode, a rendezvous point (RP) is involved. The challenge with sparse mode is to advertise to all sparse mode routers the location of the RP.

Three methods of advertising the RP are: (1) static configuration, (2) Auto-Rendezvous Point (Auto-RP), using Protocol Independent Multicast (PIM) sparse-dense mode, or (3) the Bootstrap Protocol (BSR).

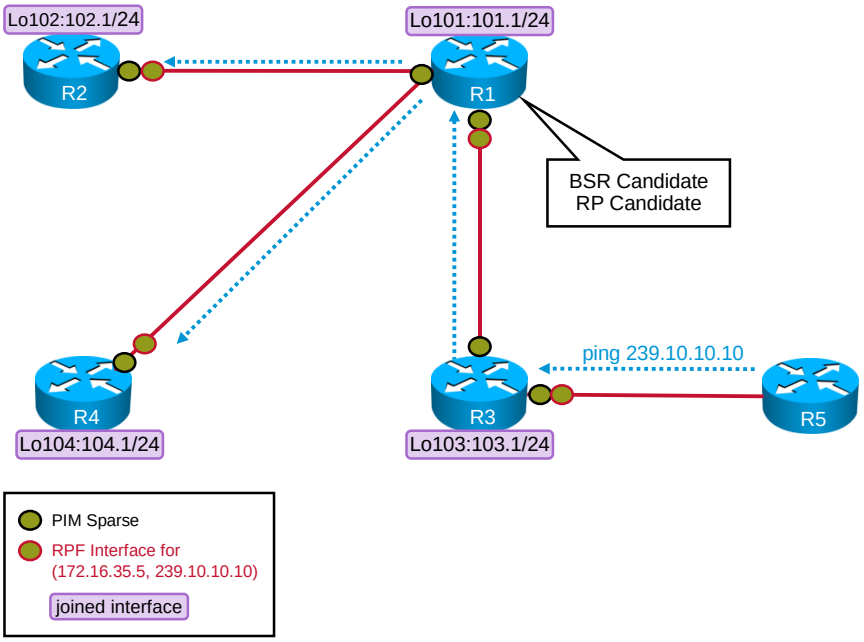
The preceding configuration requirement prohibits the use of "static configurations or dense groups." The static configuration restriction eliminates the static method of PIM sparse mode (PIM-SM) RP advertisement. The "dense group" restriction eliminates the Auto-RP method, because it requires PIM-SM or PIM dense mode (PIM-DM). Therefore, only one RP advertisement method remains—BSR. Because it is specified that R1 is to be configured as the shared root, configure R1 as the candidate RP for the 239.10.10.10 multicast group as well as the bootstrap router (BSR).

The task specified which routers should become PIM neighbors, but it did not explicitly specify all the interfaces to be configured with PIM. Interfaces facing servers and clients usually need to be configured with PIM.

Verification:

To distribute traffic in sparse mode, the router must have the address of an RP. Verify this by using the command **show ip pim rp mapping**:

```
R1#show ip pim rp mapping
PIM Group-to-RP Mappings
This system is a candidate RP (v2)
This system is the Bootstrap Router (v2)

Group(s) 224.0.0.0/4
  RP 172.16.101.1 (?), v2
    Info source: 172.16.101.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 08:37:53, expires: 00:01:34
R1#


R2#show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.101.1 (?), v2
    Info source: 172.16.101.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 08:39:24, expires: 00:01:39
R2#


R3#show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.101.1 (?), v2
    Info source: 172.16.101.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 08:36:48, expires: 00:02:11
R3#


R4#show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 172.16.101.1 (?), v2
    Info source: 172.16.101.1 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 08:40:14, expires: 00:01:45
R4#
```

Verification:

R5 should receive the replies from all routers specified in the multicast scenario.

```
R5#ping 239.10.10.10
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.10.10.10, timeout is 2 seconds:

Reply to request 0 from 172.16.103.1, 1 ms
Reply to request 0 from 172.16.104.1, 9 ms
Reply to request 0 from 172.16.102.1, 9 ms
Reply to request 0 from 172.16.101.1, 9 ms
R5#
```

Here is an example of the multicast routing table on R1:

```
R1#show ip mroute 239.10.10.10
IP Multicast Routing Table

<skipped for brevity>

(*, 239.10.10.10), 12:33:51/00:03:22, RP 172.16.101.1, flags: SJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/0, Forward/Sparse, 11:49:18/00:02:32
    Loopback101, Forward/Sparse, 12:33:50/00:02:19
    Tunnel124, Forward/Sparse, 12:32:51/00:03:22

(172.16.35.5, 239.10.10.10), 00:00:53/00:02:43, flags: LT
  Incoming interface: Serial1/0, RPF nbr 172.16.13.3
  Outgoing interface list:
    Tunnel124, Forward/Sparse, 00:00:53/00:03:22
    Loopback101, Forward/Sparse, 00:00:53/00:02:19
```

```
R1#
```

Note that R1 shows the Serial1/0 interface as incoming and the Tunnel124 and Loopback101 interfaces as outgoing.

14. MPLS Section

Note	Configure all OSPF routers for network 172.16.0.0/24 with the PID 100. OSPF PIDs must be different for 172.16.0.0/24 and 10.0.0.0/24. Use your IGP and MPLS diagrams to help guide configuration.

Issue:

Configure the MPLS forwarding on the required interfaces.

Configuration and verification:

Configure global configuration commands **ip cef** and **mpls ip** on R4 and R5.

Configure the interface with the **mpls ip** interface command:

R4:

```
interface Serial1/1
 ip address 10.10.45.4 255.255.255.0
 mpls ip
!

R4#show mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
Serial1/1              Yes (ldp)     No       No  No     Yes
R4#
```

R5:

```
interface Serial1/1
 ip address 10.10.45.5 255.255.255.0
 mpls ip
!

R5#show mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
Serial1/1              Yes (ldp)     No       No  No     Yes
R5#
```

Issue:

R4 and R5 are the label switch routers (LSRs). Label Distribution Protocol (LDP) neighbor relationship should be established between their respective IP addresses—10.1.1.4/32 and 10.1.1.5/32.

MPLS LDP enables one LSR to inform another LSR of the label bindings that it has made. Once a pair of routers communicate the LDP parameters, they establish a label-switched path (LSP). MPLS LDP enables LSRs to distribute labels along normally routed paths to support MPLS forwarding. This method of label distribution is also called hop-by-hop forwarding. With IP forwarding, when a packet arrives at a router, the router looks at the destination address in the IP header, performs a route lookup, and forwards the packet to the next hop. With MPLS forwarding, when a packet arrives at a router, the router looks at the incoming label,

looks up the label in a table, swaps (or replaces) the label in the packet with the outgoing label from the label forwarding information base (LFIB), and then forwards the packet to the next hop.

Configuration and verification:

Configure an LDP ID on R4 and R5 and the LDP neighbor relationship:

R4:

```
interface Loopback10114
 ip address 10.1.1.4 255.255.255.255

mpls ldp router-id Loopback10114
```

R5:

```
interface Loopback10115
 ip address 10.1.1.5 255.255.255.255

mpls ldp router-id Loopback10115
```

Verify LDP neighbor relationship:

R4:

```
R4#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.5:0; Local LDP Ident 10.1.1.4:0
        TCP connection: 10.1.1.5.11705 - 10.1.1.4.646
        State: Oper; Msgs sent/rcvd: 902/905; Downstream
        Up time: 12:39:56
        LDP discovery sources:
          Serial1/1, Src IP addr: 10.10.45.5
        Addresses bound to peer LDP Ident:
          172.16.35.5     170.100.10.1    10.10.25.5      10.10.45.5
          172.16.105.1    10.1.1.5
R4#
```

R5:

```
R5#show mpls ldp neighbor
    Peer LDP Ident: 10.1.1.4:0; Local LDP Ident 10.1.1.5:0
        TCP connection: 10.1.1.4.646 - 10.1.1.5.11705
        State: Oper; Msgs sent/rcvd: 906/903; Downstream
        Up time: 12:40:49
        LDP discovery sources:
          Serial1/1, Src IP addr: 10.10.45.4
        Addresses bound to peer LDP Ident:
          172.16.26.4     1.1.1.4         10.10.45.4      172.16.104.1
          10.1.1.4        172.16.124.4
R5#
```

Issue:

R4 should reject labels for all networks except for 10.2.2.0/24 advertised from R5.

Configuration and verification:

MPLS LDP supports inbound label-binding filtering. You can use the MPLS LDP feature to configure access control lists (ACLs) for controlling the label bindings that an LSR accepts from its peer LSRs.

If you check the binding on R4 before applying the inbound filter, you will notice that R5 advertises the labels for all prefixes that are in its routing table. You want to accept only the label for the 10.2.2.0/24 on R4.

R4:

```
R4#show mpls ldp bindings neighbor 10.1.1.5
  tib entry: 10.1.1.4/32, rev 12
        remote binding: lsr: 10.1.1.5:0, label: 16
  tib entry: 10.1.1.5/32, rev 14
        remote binding: lsr: 10.1.1.5:0, label: imp-null
  tib entry: 10.2.2.0/24, rev 64
        remote binding: lsr: 10.1.1.5:0, label: 40
  tib entry: 10.10.25.0/24, rev 16
        remote binding: lsr: 10.1.1.5:0, label: imp-null
  tib entry: 10.10.45.0/24, rev 8
        remote binding: lsr: 10.1.1.5:0, label: imp-null
  tib entry: 10.22.22.0/24, rev 66
        remote binding: lsr: 10.1.1.5:0, label: 41
  tib entry: 170.100.10.0/24, rev 69
        remote binding: lsr: 10.1.1.5:0, label: imp-null
  tib entry: 172.16.1.0/24, rev 43
        remote binding: lsr: 10.1.1.5:0, label: 24
  tib entry: 172.16.13.0/24, rev 44
        remote binding: lsr: 10.1.1.5:0, label: 19
  tib entry: 172.16.16.0/24, rev 18
        remote binding: lsr: 10.1.1.5:0, label: 25
  tib entry: 172.16.26.0/24, rev 2
        remote binding: lsr: 10.1.1.5:0, label: 26
  tib entry: 172.16.30.0/24, rev 20
        remote binding: lsr: 10.1.1.5:0, label: 27
  tib entry: 172.16.31.0/24, rev 45
        remote binding: lsr: 10.1.1.5:0, label: 20
  tib entry: 172.16.35.0/24, rev 46
        remote binding: lsr: 10.1.1.5:0, label: imp-null
  tib entry: 172.16.50.0/24, rev 47
        remote binding: lsr: 10.1.1.5:0, label: 28
  tib entry: 172.16.60.0/28, rev 48
        remote binding: lsr: 10.1.1.5:0, label: 21
  tib entry: 172.16.101.0/24, rev 49
        remote binding: lsr: 10.1.1.5:0, label: 29
  tib entry: 172.16.102.0/24, rev 50
        remote binding: lsr: 10.1.1.5:0, label: 30
  tib entry: 172.16.103.0/24, rev 51
        remote binding: lsr: 10.1.1.5:0, label: 22
  tib entry: 172.16.104.0/24, rev 10
        remote binding: lsr: 10.1.1.5:0, label: 31
  tib entry: 172.16.105.0/24, rev 26
        remote binding: lsr: 10.1.1.5:0, label: imp-null
  tib entry: 172.16.106.0/24, rev 22
        remote binding: lsr: 10.1.1.5:0, label: 32
  tib entry: 172.16.107.0/24, rev 52
        remote binding: lsr: 10.1.1.5:0, label: 33
  tib entry: 172.16.110.0/24, rev 53
        remote binding: lsr: 10.1.1.5:0, label: 34
  tib entry: 172.16.120.0/24, rev 54
        remote binding: lsr: 10.1.1.5:0, label: 23
  tib entry: 172.16.124.0/24, rev 6
        remote binding: lsr: 10.1.1.5:0, label: 18
  tib entry: 173.35.33.0/24, rev 59
        remote binding: lsr: 10.1.1.5:0, label: 35
  tib entry: 192.168.2.0/24, rev 57
        remote binding: lsr: 10.1.1.5:0, label: 36
  tib entry: 192.168.6.0/24, rev 58
        remote binding: lsr: 10.1.1.5:0, label: 37
R4#
```

Configure a standard access list to permit network 10.2.2.0/24, and apply it to the LDP neighbor statement on R4:

```
mpls ldp neighbor 10.1.1.5 labels accept 1

access-list 1 permit 10.2.2.0 0.0.0.255
```


Verify the binding table on R4 again:

```
R4#show mpls ldp bindings neighbor 10.1.1.5
  tib entry: 10.2.2.0/24, rev 64
        remote binding: lsr: 10.1.1.5:0, label: 40
R4#
```