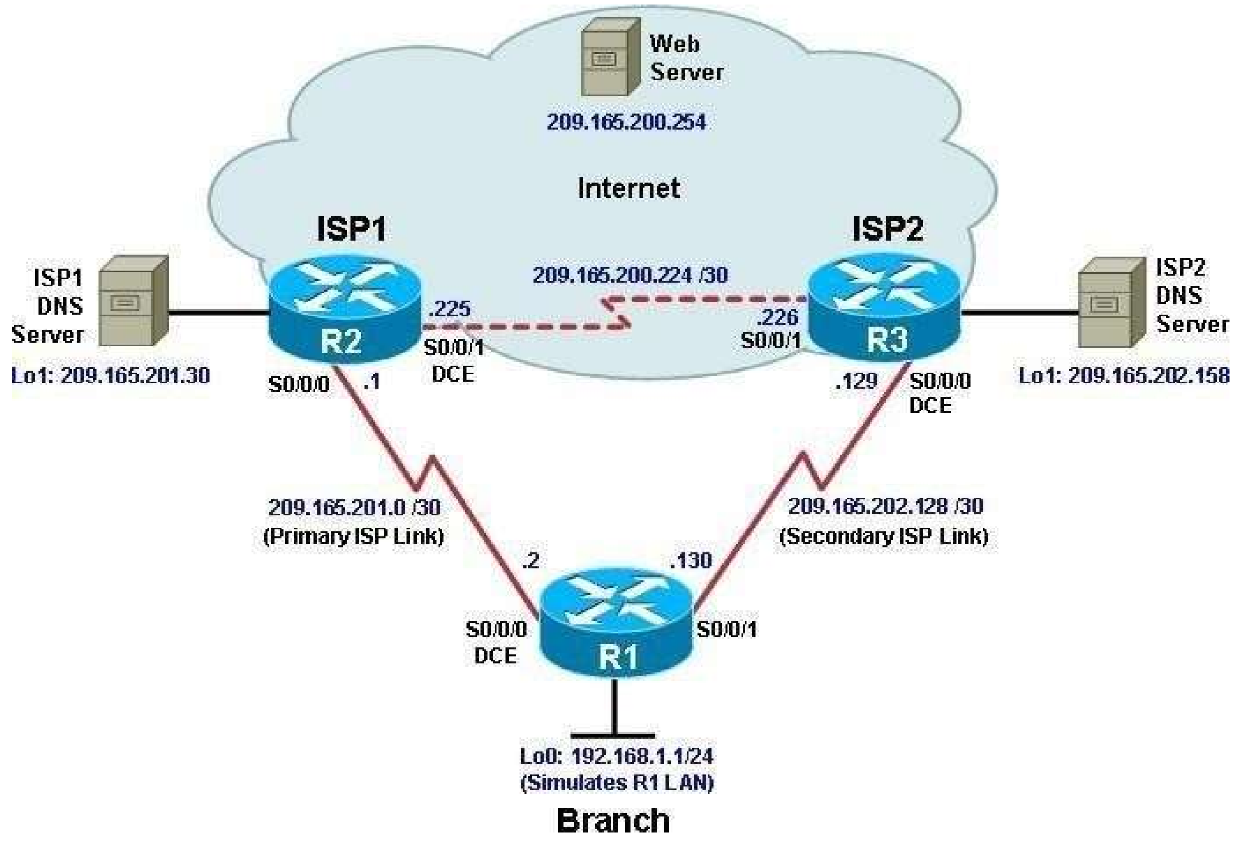
MODERN NETWORKING

INDEX

Practical No. 01

Aim: Configure IP SLA Tracking and Path Control

Topology



Objectives

- Configure and verify the IP SLA feature.
- Test the IP SLA tracking feature.
- Verify the configuration and operation using **show** and **debug** commands.

Background

You want to experiment with the Cisco IP Service Level Agreement (SLA) feature to study how it could be of value to your organization.

At times, a link to an ISP could be operational, yet users cannot connect to any other outside Internet resources. The problem might be with the ISP or downstream from them. Although policy-based routing (PBR) can be implemented to alter path control, you will implement the Cisco IOS SLA feature to monitor this behavior and intervene by injecting another default route to a backup ISP.

To test this, you have set up a three-router topology in a lab environment. Router R1 represents a branch office connected to two different ISPs. ISP1 is the preferred connection to the Internet, while ISP2 provides a backup link. ISP1 and ISP2 can also interconnect, and both can reach the web server. To monitor ISP1 for failure, you will configure IP SLA probes to track the reachability to the ISP1 DNS server. If connectivity to the ISP1 server fails, the SLA probes detect the failure and alter the default static route to point to the ISP2 server.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers and configure the router hostname and interface addresses.

a. Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear the previous configurations. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to them as well as the serial interfaces on R1, ISP1, and ISP2.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
hostname R1

interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0

interface Serial0/0/0
 description R1 --> ISP1
 ip address 209.165.201.2 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description R1 --> ISP2
 ip address 209.165.202.130 255.255.255.252
 bandwidth 128
 no shutdown
```

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255
 interface Serial0/0/0 description ISP1 --> R1
 ip address 209.165.201.1 255.255.255.252
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP1 --> ISP2
 ip address 209.165.200.225 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown
```

Router ISP2 (R3)

```
hostname ISP2

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0
 description ISP2 --> R1
 ip address 209.165.202.129 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP2 --> ISP1
 ip address 209.165.200.226 255.255.255.252
 bandwidth 128
 no shutdown
```

b. Verify the configuration by using the **show interfaces description** command. The output from router R1 is shown here as an example.

```
R1# show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Fa0/1                          admin down     down
Se0/0/0                        up             up       R1 --> ISP1
Se0/0/1                        up             up       R1 --> ISP2
Lo0                            up             up       R1 LAN
```

All three interfaces should be active. Troubleshoot if necessary.

c. The current routing policy in the topology is as follows:

- Router R1 establishes connectivity to the Internet through ISP1 using a default static route.
- ISP1 and ISP2 have dynamic routing enabled between them, advertising their respective public address pools.
- ISP1 and ISP2 both have static routes back to the ISP LAN.

Note: For the purpose of this lab, the ISPs have a static route to an RFC 1918 private network address on the branch router R1. In an actual branch implementation, Network Address Translation (NAT) would be configured for all traffic exiting the branch LAN. Therefore, the static routes on the ISP routers would be pointing to the provided public pool of the branch office. This is covered in Lab 7-1, "Configure Routing Facilities to the Branch Office."

Implement the routing policies on the respective routers. You can copy and paste the following configurations.

Router R1

```
ip route 0.0.0.0 0.0.0.0 209.165.201.1
```

Router ISP1 (R2)

```
router eigrp 1
 network 209.165.200.224 0.0.0.3
 network 209.165.201.0 0.0.0.31
 no auto-summary

ip route 192.168.1.0 255.255.255.0 209.165.201.2
```

Router ISP2 (R3)

```
router eigrp 1
 network 209.165.200.224 0.0.0.3
 network 209.165.202.128 0.0.0.31
 no auto-summary

ip route 192.168.1.0 255.255.255.0 209.165.202.130
```

EIGRP neighbor relationship messages on ISP1 and ISP2 should be generated. Troubleshoot if necessary.

```
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 209.165.200.225 (Serial0/0/1) is
up: new adjacency
```

Step 2: Verify server reachability.

The Cisco IOS IP SLA feature enables an administrator to monitor network performance between Cisco devices (switches or routers) or from a Cisco device to a remote IP device. IP SLA probes continuously check the reachability of a specific destination, such as a provider edge router interface, the DNS server of the ISP, or any other specific destination, and can conditionally announce a default route only if the connectivity is verified.

a. Before implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2 DNS server to verify connectivity. You can copy the following Tcl script and paste it intoR1.

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
ping $address source 192.168.1.1
}

R1(tcl)# foreach address {
+>(tcl)# 209.165.200.254.
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } {
+>(tcl)# ping $address source 192.168.1.1
+>(tcl)#}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.254, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.30, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.158, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms
```

b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server. You can copy the following Tcl script and paste it into R1.

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
trace $address source 192.168.1.1
}
```

```
R1(tcl)# foreach address {
+>(tcl)# 209.165.200.254
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } {
+>(tcl)# trace $address source 192.168.1.1
+>(tcl)# }

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.201.1 20 msec 8 msec *
Type escape sequence to abort.
Tracing the route to 209.165.201.30

  1 209.165.201.1 8 msec 8 msec *
Type escape sequence to abort.
Tracing the route to 209.165.202.158

  1 209.165.201.1 8 msec 8 msec 4 msec
  2 209.165.200.226 8 msec 8 msec *
```

Through which ISP is traffic flowing?

.

Step 3: Configure IP SLA probes.

When the reachability tests are successful, you can configure the Cisco IOS IP SLAs probes. Different types of probes can be created, including FTP, HTTP, and jitter probes. In this scenario, you will configure ICMP echo probes.

a. Create an ICMP echo probe on R1 to the primary DNS server on ISP1 using the **ip sla** command.

Note: With Cisco IOS Release 12.4(4)T, 12.2(33)SB, and 12.2(33)SXI, the **ip sla** command has replaced the previous **ip sla monitor** command. In addition, the **icmp-echo** command has replaced the **type echo protocol ipIcmpEcho** command.

```
R1(config)# ip sla 11
R1(config-ip-sla)# icmp-echo 209.165.201.30
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)# ip sla schedule 11 life forever start-time now
```

The operation number of 11 is only locally significant to the router. The **frequency 10** command schedules the connectivity test to repeat every 10 seconds. The probe is scheduled to start now and to run forever.

b. Verify the IP SLAs configuration of operation 11 using the **show ip sla configuration 11** command.

Note: With Cisco IOS Release 12.4(4)T, 12.2(33)SB, and 12.2(33)SXI, the **show ip sla configuration** command has replaced the **show ip sla monitor configuration** command.

```
R1# show ip sla configuration 11
IP SLAs, Infrastructure Engine-II.
Entry number: 11
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.30/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```

The output lists the details of the configuration of operation 11. The operation is an ICMP echo to 209.165.201.30, with a frequency of 10 seconds, and it has already started (the start time has already passed).

c. Issue the **show ip sla statistics** command to display the number of successes, failures, and results of the latest operations.

Note: With Cisco IOS Release 12.4(4)T, 12.2(33)SB, and 12.2(33)SXI, the **show ip sla statistics** command has replaced the **show ip sla monitor statistics** command.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11

Latest operation start time: *21:22:29.707 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 5
Number of failures: 0
Operation time to live: Forever
```

You can see that operation 11 has already succeeded five times, has had no failures, and the last operation returned an OK result.

d. Although not actually required because IP SLA session 11 alone could provide the desired fault tolerance, create a second probe, 22, to test connectivity to the second DNS server located on router ISP2. You can copy and paste the following commands on R1.

```
ip sla 22
icmp-echo 209.165.202.158
frequency 10
exit
ip sla schedule 22 life forever start-time now
```

e. Verify the new probe using the **show ip sla configuration** and **show ip sla statistics** commands.

```
R1# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1 Statistic distribution interval
   (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:

R1# show ip sla statistics 22
IPSLAs Latest Operation Statistics

IPSLA operation id: 22

Latest operation start time: *21:24:14.215 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 4
Number of failures: 0
Operation time to live: Forever
```

The output lists the details of the configuration of operation 22. The operation is an ICMP echo to 209.165.202.158, with a frequency of 10 seconds, and it has already started (the start time has already passed). The statistics also prove that operation 22 is active.

Step 4: Configure tracking options.

Although PBR could be used, you will configure a floating static route that appears or disappears depending on the success or failure of the IP SLA.

a. Remove the current default route on R1, and replace it with a floating static route having an administrative distance of 5.

```
R1(config)# no ip route 0.0.0.0 0.0.0.0 209.165.201.1
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 5
R1(config)# exit
```

b. Verify the routing table.

```
R1# show ip route
*Apr  2 20:00:37.367: %SYS-5-CONFIG_I: Configured from console by console
Codes: 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

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C   209.165.201.0  is  directly  connected,  Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C   209.165.202.128  is  directly  connected,  Serial0/0/1C
192.168.1.0/24  is  directly  connected,  FastEthernet0/0  S*
0.0.0.0/0 [5/0] via 209.165.201.1
```

Notice that the default static route is now using the route with the administrative distance of 5. The first tracking object is tied to IP SLA object 11..

c. Use the **track 1 ip sla 11 reachability** command to enter the config-track subconfiguration mode.

Note: With Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE and Cisco IOS XE Release 2.4, the **track ip sla** command has replaced the **track rtr** command.

```
R1(config)# track 1 ip sla 11 reachability
R1(config-track)#
```

d. Specify the level of sensitivity to changes of tracked objects to 10 seconds of down delay and 1 second of up delay using the **delay down 10 up 1** command. The delay helps to alleviate the effect of flapping objects—objects that are going down and up rapidly. In this situation, if the DNS server fails momentarily and comes back up within 10 seconds, there is no impact.

```
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)#
```

e. Configure the floating static route that will be implemented when tracking object 1 is active. To view routing table changes as they happen, first enable the **debug ip routing** command. Next, use the **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1** command to create a floating static default route via 209.165.201.1 (ISP1). Notice that this command references the tracking object number 1, which in turn references IP SLA operation number 11.

```
R1# debug ip routing
IP routing debugging is on
R1#
*Apr  2 21:26:46.171: RT: NET-RED 0.0.0.0/0

R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
R1(config)#
*Apr  2 21:27:02.851: RT: closer admin distance for 0.0.0.0, flushing 1
routes
*Apr  2 21:27:02.851: RT: NET-RED 0.0.0.0/0
*Apr  2 21:27:02.851: RT: add 0.0.0.0/0 via 209.165.201.1, static metric
[2/0]
*Apr  2 21:27:02.851: RT: NET-RED 0.0.0.0/0
*Apr  2 21:27:02.851: RT: default path is now 0.0.0.0 via 209.165.201.1
*Apr  2 21:27:02.855: RT: new default network 0.0.0.0
*Apr  2 21:27:02.855: RT: NET-RED 0.0.0.0/0
*Apr  2 21:27:07.851: RT: NET-RED 0.0.0.0/0
```

Notice that the default route with an administrative distance of 5 has been immediately flushed because of a route with a better admin distance. It then adds the new default route with the admin distance of 2.

f. Repeat the steps for operation 22, track number 2, and assign the static route an admin distance higher than track 1 and lower than 5. On R1, copy the following configuration, which sets an admin distance of 3.

```
track 2 ip sla 22 reachability
delay down 10 up 1
exit
ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
```

g. Verify the routing table again.

```
R1# show ip route
Codes: 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 2i - IS-IS,
        su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2ia -
        IS-IS inter area, * - candidate default, U - per-user static

        o - ODR, P - periodic downloaded static route Gateway of last

resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C    209.165.201.0    is    directly    connected,    Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C 209.165.202.128 is directly connected, Serial0/0/1C192.168.1.0/24
is  directly  connected,  FastEthernet0/0  S*  0.0.0.0/0  [2/0]  via
209.165.201.1
```

Although a new default route was entered, its administrative distance is not better than 2. Therefore, itdoes not replace the previously entered default route.

Step 5: Verify IP SLA operation.

In this step you observe and verify the dynamic operations and routing changes when tracked objects fail.The following summarizes the process:

- Disable the DNS loopback interface on ISP1 (R2).
- Observe the output of the **debug** command on R1.
- Verify the static route entries in the routing table and the IP SLA statistics of R1.
- Re-enable the loopback interface on ISP1 (R2) and again observe the operation of the IP SLAtracking feature.

```
ISP1(config)# interface loopback 1
ISP1(config-if)# shutdown ISP1(config-if)#
*Apr  2 15:53:14.307: %LINK-5-CHANGED: Interface Loopback1, changed state to
administratively down
*Apr  2 15:53:15.307: %LINEPROTO-5-UPDOWN: Line protocol on InterfaceLoopback1,
changed state to down
```

a. Shortly after the loopback interface is administratively down, observe the debug output being generatedon R1.

```
R1#
*Apr  2 21:32:33.323: %TRACKING-5-STATE: 1 ip sla 11 reachability Up->Down
*Apr  2 21:32:33.323: RT: del 0.0.0.0 via 209.165.201.1, static metric [2/0]
*Apr  2 21:32:33.323: RT: delete network route to 0.0.0.0
*Apr  2 21:32:33.323: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:33.323: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:33.323: RT: add 0.0.0.0/0 via 209.165.202.129, static metric
[3/0]
*Apr  2 21:32:33.323: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:33.323: RT: default path is now 0.0.0.0 via 209.165.202.129
*Apr  2 21:32:33.323: RT: new default network 0.0.0.0
*Apr  2 21:32:33.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:46.171: RT: NET-RED 0.0.0.0/0
```

The tracking state of track 1 changes from up to down. This is the object that tracked reachability for IPSLA object 11, with an ICMP echo to the ISP1 DNS server at 209.165.201.30.

R1 then proceeds to delete the default route with the administrative distance of 2 and installs the nexthighest default route to ISP2 with the administrative distance of 3.

```
route
```

b. Verify the routing table.

```
R1# show ip route
Codes: 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

Gateway of last resort is 209.165.202.129 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C   209.165.201.0  is  directly  connected,  Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C   209.165.202.128  is  directly  connected,  Serial0/0/1C
192.168.1.0/24  is  directly  connected,  FastEthernet0/0  S*
0.0.0.0/0 [3/0] via 209.165.202.129
```

The new static route has an administrative distance of 3 and is being forwarded to ISP2 as it should.

c. Verify the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

PSLA operation id: 11
Type of operation: icmp-echo
        Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *15:36:42.871 UTC Fri Apr 2 2010
Latest operation return code: No connection
Number of successes: 84
Number of failures: 13
Operation time to live: Forever


IPSLA operation id: 22
Type of operation: icmp-echo
        Latest RTT: 8 milliseconds
Latest operation start time: *15:36:46.335 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 81
Number of failures: 1
Operation time to live: Forever
```

Notice that the latest return code is **No connection** and there have been 12 failures on IP SLA object 11.

d. Initiate a trace to the web server from the internal LAN IP address.

```
R1# trace 209.165.200.254 source 192.168.1.1

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.202.129 8 msec 8 msec *
```

This confirms that traffic is leaving router R1 and being forwarded to the ISP2 router.

e. To examine the routing behavior when connectivity to the ISP1 DNS is restored, re-enable the DNS address on ISP1 (R2) by issuing the **no shutdown** command on the loopback 1 interface on ISP2.

```
ISP1(config-if)# no shutdown
*Apr  2 15:56:24.655: %LINK-3-UPDOWN: Interface Loopback1, changed state to
up
*Apr  2 15:56:25.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback1, changed state to up
```

Notice the output of the **debug ip routing** command on R1.

```
R1#
*Apr  2 21:35:34.327: %TRACKING-5-STATE: 1 ip sla 11 reachability Down->Up
*Apr  2 21:35:34.327: RT: closer admin distance for 0.0.0.0, flushing 1
routes
*Apr  2 21:35:34.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:34.327: RT: add 0.0.0.0/0 via 209.165.201.1, static metric
[2/0]
*Apr  2 21:35:34.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:34.327: RT: default path is now 0.0.0.0 via 209.165.201.1
*Apr  2 21:35:34.327: RT: new default network 0.0.0.0
*Apr  2 21:35:34.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:39.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:46.171: RT: NET-RED 0.0.0.0/0
```

Now the IP SLA 11 operation transitions back to an up state and reestablishes the default static route to ISP1 with an administrative distance of 2.

f. Again examine the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

Type of operation: icmp-echo
        Latest RTT: 8 milliseconds
Latest operation start time: *15:40:42.871 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 88
Number of failures: 35
Operation time to live: Forever


IPSLA operation id: 22
Type of operation:  icmp-echo
        Latest RTT: 16 milliseconds
Latest operation start time: *15:40:46.335 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 105
Number of failures: 1
Operation time to live: Forever
```

The IP SLA 11 operation is active again, as indicated by the OK return code, and the number of successes is incrementing.

g. Verify the routing table.

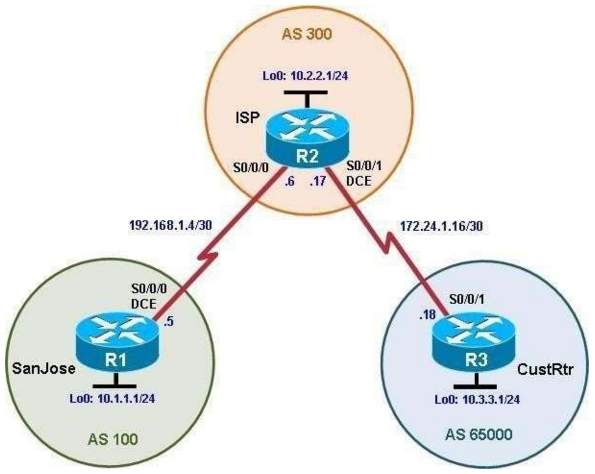
```
R1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2

i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2ia - IS-
```

```
IS inter area, * - candidate default, U - per- user static

              o - ODR, P - periodic downloaded static route
       Gateway of last resort is 209.165.201.1 to network 0.0.0.0

            209.165.201.0/30 is subnetted, 1 subnets
       C 209.165.201.0 is directly connected, Serial0/0/0
            209.165.202.0/30 is subnetted, 1 subnets
       C      209.165.202.128 is directly connected, Serial0/0/1C
       192.168.1.0/24 is directly connected, FastEthernet0/0
              S*  0.0.0.0/0 [2/0] via 209.165.201.
```

Practical No. 02

Aim: Using the AS_PATH Attribute

Topology



Objectives

- Use BGP commands to prevent private AS numbers from being advertised to the outside world.
- Use the AS_PATH attribute to filter BGP routes based on their source AS numbers.

Background

The International Travel Agency's ISP has been assigned an AS number of 300. This provider uses BGP to exchange routing information with several customer networks. Each customer network is assigned an AS number from the private range, such as AS 65000. Configure the ISP router to remove the private AS numbers from the AS Path information of CustRtr. In addition, the ISP would like to prevent its customer networks from receiving route information from International Travel Agency's AS 100. Use the AS_PATH attribute to implement this policy.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions, if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations.

Step 2: Configure the hostname and interface addresses.

a. You can copy and paste the following configurations into your routers to begin.

Router R1 (hostname SanJose)

```
hostname SanJose
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.5 255.255.255.252
 clock rate 128000
 no shutdown
```

Router R2 (hostname ISP)

```
hostname ISP
!
interface Loopback0
 ip address 10.2.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.6 255.255.255.252
 no shutdown
!
interface Serial0/0/1
 ip address 172.24.1.17 255.255.255.252
 clock rate 128000
 no shutdown
```

Router R3 (hostname CustRtr)

```
hostname CustRtr
!
interface Loopback0
 ip address 10.3.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.24.1.18 255.255.255.252
 no shutdown
```

b. Use **ping** to test the connectivity between the directly connected routers.

Note: SanJose will not be able to reach either ISP's loopback (10.2.2.1) or CustRtr's loopback (10.3.3.1), nor will it be able to reach either end of the link joining ISP to CustRtr (172.24.1.17 and 172.24.1.18).

Step 3: Configure BGP.

a. Configure BGP for normal operation. Enter the appropriate BGP commands on each router so that they identify their BGP neighbors and advertise their loopback networks.

```
SanJose(config)# router bgp 100
SanJose(config-router)# neighbor 192.168.1.6 remote-as 300
SanJose(config-router)# network 10.1.1.0 mask 255.255.255.0

ISP(config)# router bgp 300
ISP(config-router)# neighbor 192.168.1.5 remote-as 100
ISP(config-router)# neighbor 172.24.1.18 remote-as 65000
ISP(config-router)# network 10.2.2.0 mask 255.255.255.0
```

```
CustRtr(config)# router bgp 65000
CustRtr(config-router)# neighbor 172.24.1.17 remote-as 300
CustRtr(config-router)# network 10.3.3.0 mask 255.255.255.0
```

b. Verify that these routers have established the appropriate neighbor relationships by issuing the **show ipbgp neighbors** command on each router.

```
ISP# show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 65000, external link
  BGP version 4, remote router ID 10.3.3.1
  BGP state = Established, up for 00:02:05
<output omitted>

BGP neighbor is 192.168.1.5,  remote AS 100, external link
  BGP version 4, remote router ID 10.1.1.1
  BGP state = Established, up for 00:04:19
<output omitted>
```

Step 4: Remove the private AS.

a. Display the SanJose routing table using the **show ip route** command. SanJose should have a route to both 10.2.2.0 and 10.3.3.0. Troubleshoot if necessary.

```
SanJose# show ip route
Codes: 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

Gateway of last resort is not set


     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 192.168.1.6, 00:01:11
B       10.2.2.0 [20/0] via 192.168.1.6, 00:02:16
C       10.1.1.0 is directly connected,
     Loopback0192.168.1.0/30 is subnetted, 1
     subnets

C       192.168.1.4 is directly connected, Serial0/0/0
```

b. Ping the 10.3.3.1 address from

SanJose.Why does this fail?

____

____

c. Ping again, this time as an extended ping, sourcing from the Loopback0 interface address.

```
SanJose# ping
Protocol [ip]:
Target IP address: 10.3.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
```

```
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/68 ms
```

Note: You can bypass extended ping mode and specify a source address using one of these commands:

```
SanJose# ping 10.3.3.1 source 10.1.1.1
```

or

```
SanJose# ping 10.3.3.1 source Lo0
```

d. Check the BGP table from SanJose by using the **show ip bgp** command. Note the AS path for the 10.3.3.0 network. The AS 65000 should be listed in the path to 10.3.3.0.

```
SanJose# show ip bgp

BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0         0.0.0.0                  0         32768 i
*> 10.2.2.0         192.168.1.6              0             0 300 i
*> 10.3.3.0         192.168.1.6                            0 300 65000 i

Why is this a
problem?
```

____

____

e. Configure ISP to strip the private AS numbers from BGP routes exchanged with SanJose using the following commands.

```
ISP(config)# router bgp 300

ISP(config-router)# neighbor 192.168.1.5 remove-private-as
```

f. After issuing these commands, use the **clear ip bgp *** command on ISP to reestablish the BGP relationship between the three routers. Wait several seconds and then return to SanJose to check its routing table.

Note: The **clear ip bgp * soft** command can also be used to force each router to resend its BGP table.

Does SanJose still have a route to 10.3.3.0?

____

SanJose should be able to ping 10.3.3.1 using its loopback 0 interface as the source of the ping.

```
SanJose# ping 10.3.3.1 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
```

```
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

g. Now check the BGP table on SanJose. The AS_ PATH to the 10.3.3.0 network should be AS 300. It no longer has the private AS in the path.

```
SanJose# show ip bgp

BGP table version is 8, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0         0.0.0.0                  0         32768 i
*> 10.2.2.0         192.168.1.6              0             0 300 i
*> 10.3.3.0         192.168.1.6                            0 300 i
```

Step 5: Use the AS_PATH attribute to filter routes.

As a final configuration, use the AS_PATH attribute to filter routes based on their origin. In a complex environment, you can use this attribute to enforce routing policy. In this case, the provider router, ISP, must be configured so that it does not propagate routes that originate from AS 100 to the customer router CustRtr.

AS-path access lists are read like regular access lists. The statements are read sequentially, and there is an implicit deny at the end. Rather than matching an address in each statement like a conventional access list, AS path access lists match on something called a regular expression. Regular expressions are a way of matching text patterns and have many uses. In this case, you will be using them in the AS path access list to match text patterns in AS paths.

a. Configure a special kind of access list to match BGP routes with an AS_PATH attribute that both begins and ends with the number 100. Enter the following commands on ISP.

```
ISP(config)# ip as-path access-list 1 deny ^100$
ISP(config)# ip as-path access-list 1 permit .*
```

The first command uses the **^** character to indicate that the AS path must begin with the given number 100. The **$** character indicates that the AS_PATH attribute must also end with 100. Essentially, this statement matches only paths that are sourced from AS 100. Other paths, which might include AS 100 along the way, will not match this list.

In the second statement, the **.** (period) is a wildcard, and the ***** (asterisk) stands for a repetition of the wildcard. Together, **.*** matches any value of the AS_PATH attribute, which in effect permits any update that has not been denied by the previous **access-list** statement.

For more details on configuring regular expressions on Cisco routers, see:

http://www.cisco.com/en/US/docs/ios/12_2/termserv/configuration/guide/tcfaapre_ps1835_TSD_Products_Configuration_Guide_Chapter.html

b. Apply the configured access list using the **neighbor** command with the **filter-list** option.

```
ISP(config)# router bgp 300
ISP(config-router)# neighbor 172.24.1.18 filter-list 1 out
```

The **out** keyword specifies that the list is applied to routing information sent to this neighbor.

c. Use the **clear ip bgp *** command to reset the routing information. Wait several seconds and then check the routing table for ISP. The route to 10.1.1.0 should be in the routing table.

Note: To force the local router to resend its BGP table, a less disruptive option is to use the **clear ip bgp**

*** out** or **clear ip bgp * soft** command (the second command performs both outgoing and incoming route resync).

```
ISP# show ip route
<output omitted>

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected,
     Serial0/0/110.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 172.24.1.18, 00:07:34
C       10.2.2.0 is directly connected,
Loopback0B 10.1.1.0 [20/0] via 192.168.1.5,
00:10:53
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
```

d. Check the routing table for CustRtr. It should not have a route to 10.1.1.0 in its routing table.

```
CustRtr# show ip route
<output omitted>

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected,
     Serial0/0/110.0.0.0/24 is subnetted, 2 subnets
C       10.3.3.0 is directly connected,
Loopback0B 10.2.2.0 [20/0] via 172.24.1.17,
00:11:57
```

e. Return to ISP and verify that the filter is working as intended. Issue the **show ip bgp regexp ^100$** command.

```
ISP# show ip bgp regexp ^100$
BGP table version is 4, local router ID is 10.2.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0        192.168.1.5              0              0 100 i
```

The output of this command shows all matches for the regular expressions that were used in the access list. The path to 10.1.1.0 matches the access list and is filtered from updates to CustRtr.

f. Run the following Tcl script on all routers to verify whether there is connectivity. All pings from ISP should be successful. SanJose should not be able to ping the CustRtr loopback 10.3.3.1 or the WAN link

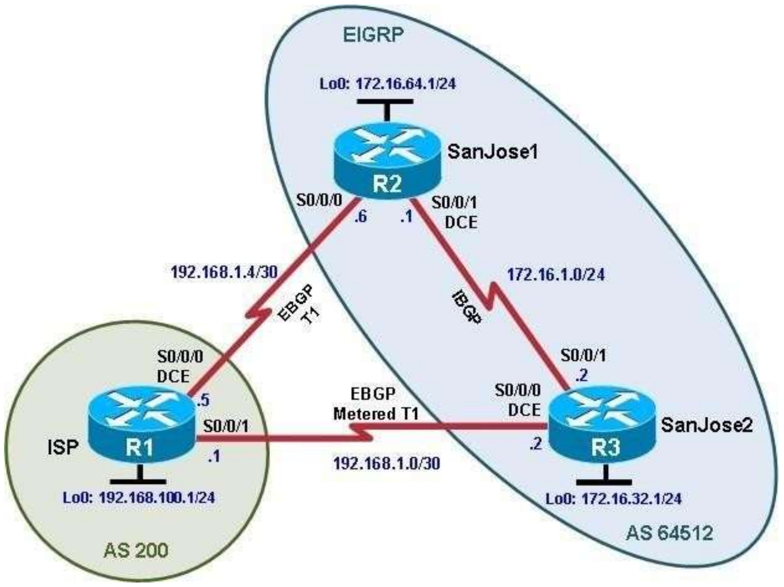
172.24.1.16/30. CustRtr should not be able to ping the SanJose loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

```
ISP# tclsh

foreach address {
10.1.1.1
10.2.2.1
10.3.3.1
192.168.1.5
192.168.1.6
172.24.1.17
172.24.1.18
} {
ping $address }
```

Practical No. 03

Aim: Configuring IBGP and EBGP Sessions, Local Preference, and MED

Topology



Objectives

- For IBGP peers to correctly exchange routing information, use the **next-hop-self** command with the **Local-Preference** and **MED** attributes.
- Ensure that the flat-rate, unlimited-use T1 link is used for sending and receiving data to and from the AS 200 on ISP and that the metered T1 only be used in the event that the primary T1 link has failed.

Background

The International Travel Agency runs BGP on its SanJose1 and SanJose2 routers externally with the ISP router in AS 200. IBGP is run internally between SanJose1 and SanJose2. Your job is to configure both EBGP and IBGP for this internetwork to allow for redundancy. The metered T1 should only be used in the event that the primary T1 link has failed. Traffic sent across the metered T1 link offers the same bandwidth of the primary link but at a huge expense. Ensure that this link is not used unnecessarily.

Note: This lab uses Cisco 1941 routers with Cisco IOS Release 15.4 with IP Base. The switches are Cisco WS-C2960-24TT-L with Fast Ethernet interfaces, therefore the router will use routing metrics associated with a 100 Mb/s interface. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco IOS Release 15.2 or comparable)
- Serial and Ethernet cables

Step 0: Suggested starting configurations.

a. Apply the following configuration to each router along with the appropriate **hostname**. The **exec-timeout 0 0** command should only be used in a lab environment.

```
Router(config)# no ip domain-lookup
Router(config)# line con 0
Router(config-line)# logging synchronous
Router(config-line)# exec-timeout 0 0
```

Step 1: Configure interface addresses.

a. Using the addressing scheme in the diagram, create the loopback interfaces and apply IPv4 addresses to these and the serial interfaces on ISP (R1), SanJose1 (R2), and SanJose2 (R3).

Router R1 (hostname ISP)

```
ISP(config)# interface Loopback0
ISP(config-if)# ip address 192.168.100.1 255.255.255.0
ISP(config-if)# exit
ISP(config)# interface Serial0/0/0
ISP(config-if)# ip address 192.168.1.5 255.255.255.252
ISP(config-if)# clock rate 128000
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface Serial0/0/1
ISP(config-if)# ip address 192.168.1.1 255.255.255.252
ISP(config-if)# no shutdown
ISP(config-if)# end
ISP#
```

Router R2 (hostname SanJose1)

```
SanJose1(config)# interface Loopback0
SanJose1(config-if)# ip address 172.16.64.1 255.255.255.0
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/0
SanJose1(config-if)# ip address 192.168.1.6 255.255.255.252
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/1
SanJose1(config-if)# ip address 172.16.1.1 255.255.255.0
SanJose1(config-if)# clock rate 128000
SanJose1(config-if)# no shutdown
SanJose1(config-if)# end
SanJose1#
Router R3 (hostname SanJose2)
SanJose2(config)# interface Loopback0
SanJose2(config-if)# ip address 172.16.32.1 255.255.255.0
SanJose2(config-if)# exit
SanJose2(config)# interface Serial0/0/0
SanJose2(config-if)# ip address 192.168.1.2 255.255.255.252
SanJose2(config-if)# clock rate 128000
SanJose2(config-if)# no shutdown
SanJose2(config-if)# exit
SanJose2(config)# interface Serial0/0/1
SanJose2(config-if)# ip address 172.16.1.2 255.255.255.0
SanJose2(config-if)# no shutdown
SanJose2(config-if)# end
SanJose2#
```

b. Use **ping** to test the connectivity between the directly connected routers. Both SanJose routers should be able to ping each other and their local ISP serial link IP address. The ISP router cannot reach the segment between SanJose1 and SanJose2.

Step 2: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 routers. (Note: If using an IOS prior to 15.0, use the no auto-summary router configuration command to disable automatic summarization. This command is the default beginning with IOS 15.)

```
SanJose1(config)# router eigrp 1
SanJose1(config-router)# network 172.16.0.0

SanJose2(config)# router eigrp 1
SanJose2(config-router)# network 172.16.0.0
```

Step 3: Configure IBGP and verify BGP neighbors.

a. Configure IBGP between the SanJose1 and SanJose2 routers. On the SanJose1 router, enter the following configuration.

```
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 172.16.32.1 remote-as 64512
SanJose1(config-router)# neighbor 172.16.32.1 update-source lo0
```

If multiple pathways to the BGP neighbor exist, the router can use multiple IP interfaces to communicate with the neighbor. The source IP address therefore depends on the outgoing interface. The **update-source lo0** command instructs the router to use the IP address of the interface Loopback0 as the source IP address for all BGP messages sent to that neighbor.

b. Complete the IBGP configuration on SanJose2 using the following commands.

```
SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 172.16.64.1 remote-as 64512
SanJose2(config-router)# neighbor 172.16.64.1 update-source lo0
```

c. Verify that SanJose1 and SanJose2 become BGP neighbors by issuing the **show ip bgp neighbors** command on SanJose1. View the following partial output. If the BGP state is not established, troubleshoot the connection.

```
SanJose2# show ip bgp neighbors
BGP neighbor is 172.16.64.1, remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.64.1
  BGP state = Established, up for 00:00:22
  Last read 00:00:22, last write 00:00:22, hold time is 180, keepalive interval
is 60 seconds
<output omitted>
```

The link between SanJose1 and SanJose2 should be identified as an internal link indicating an IBGP peering relationship, as shown in the output.

Step 4: Configure EBGP and verify BGP neighbors.

a. Configure ISP to run EBGP with SanJose1 and SanJose2. Enter the following commands on ISP.

```
ISP(config)# router bgp 200
ISP(config-router)# neighbor 192.168.1.6 remote-as 64512
ISP(config-router)# neighbor 192.168.1.2 remote-as 64512
ISP(config-router)# network 192.168.100.0
```

Because EBGP sessions are almost always established over point-to-point links, there is no reason to use the **update-source** keyword in this configuration. Only one path exists between the peers. If this path goes down, alternative paths are not available.

b. Configure a discard static route for the 172.16.0.0/16 network. Any packets that do not have a more specific match (longer match) for a 172.16.0.0 subnet will be dropped instead of sent to the ISP. Later in this lab we will configure a default route to the ISP.

```
SanJose1(config)# ip route 172.16.0.0 255.255.0.0 null0
```

c. Configure SanJose1 as an EBGP peer to ISP.

```
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 192.168.1.5 remote-as 200
SanJose1(config-router)# network 172.16.0.0
```

d. Use the **show ip bgp neighbors** command to verify that SanJose1 and ISP have reached the established state. Troubleshoot if necessary.

```
SanJose1# show ip bgp neighbors
BGP neighbor is 172.16.32.1, remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.32.1
  BGP state = Established, up for 00:12:43
<output omitted>

BGP neighbor is 192.168.1.5,  remote AS 200, external link
  BGP version 4, remote router ID 192.168.100.1
  BGP state = Established, up for 00:06:49
  Last read 00:00:42, last write 00:00:45, hold time is 180, keepalive interval
is 60 seconds
<output omitted>
```

Notice that the “external link” indicates that an EBGP peering session has been established. You should also see an informational message indicating the establishment of the BGP neighbor relationship.

```
*Sep  8 21:09:59.699: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
```

e. Configure a discard static route for 172.16.0.0/16 on SanJose2 and as an EBGP peer to ISP.

```
SanJose2(config)# ip route 172.16.0.0 255.255.0.0 null0
SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 192.168.1.1 remote-as 200
SanJose2(config-router)# network 172.16.0.0
```

Step 5: View BGP summary output.

In Step 4, the **show ip bgp neighbors** command was used to verify that SanJose1 and ISP had reached the established state. A useful alternative command is **show ip bgp summary**. The output should be similar to the following.

```
SanJose2# show ip bgp summary
BGP router identifier 172.16.32.1, local AS number 64512
BGP table version is 6, main routing table version 6
2 network entries using 288 bytes of memory
4 path entries using 320 bytes of memory
4/2 BGP path/bestpath attribute entries using 640 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1272 total bytes of memory
BGP activity 2/0 prefixes, 4/0 paths, scan interval 60 secs
```

```
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
172.16.64.1     4        64512      27      26        6    0    0 00:18:15
2
192.168.1.1     4          200      10       7        6    0    0 00:01:42
1
SanJose2#
```

Step 6: Verify which path the traffic takes.

f. Clear the IP BGP conversation with the **clear ip bgp *** command on ISP. Wait for the conversations to reestablish with each SanJose router.

```
ISP# clear ip bgp *
ISP#
*Nov  9 22:05:32.427: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Down User reset
*Nov  9 22:05:32.427: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.2 IPv4
Unicast topology base removed from session  User reset
*Nov  9 22:05:32.427: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Down User reset
*Nov  9 22:05:32.427: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.6 IPv4
Unicast topology base removed from session  User reset
*Nov  9 22:05:32.851: %BGP-5-ADJCHANGE: neighbor 192.168.1.2 Up
*Nov  9 22:05:32.851: %BGP-
ISP#5-ADJCHANGE: neighbor 192.168.1.6 Up
ISP#
```

g. Test whether ISP can ping the loopback 0 address of 172.16.64.1 on SanJose1 and the serial link between SanJose1 and SanJose2, 172.16.1.1.

```
ISP# ping 172.16.64.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
ISP#
ISP# ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
ISP#
```

h. Now ping from ISP to the loopback 0 address of 172.16.32.1 on SanJose2 and the serial link between SanJose1 and SanJose2, 172.16.1.2.

```
ISP# ping 172.16.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
ISP# ping 172.16.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/13/16 msISP#
```

You should see successful pings to each IP address on SanJose2 router. Ping attempts to 172.16.64.1 and 172.16.1.1 should fail. Why does this happen?

____

____

i. Issue the **show ip bgp** command on ISP to verify BGP routes and metrics.

```
ISP# show ip bgp
BGP table version is 3, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *   172.16.0.0       192.168.1.6              0             0 64512 i
 *>                   192.168.1.2              0             0 64512 i
 *>  192.168.100.0    0.0.0.0                  0         32768 i
ISP#
ISP# show ip bgp
```

Notice that ISP has two valid routes to the 172.16.0.0 network, as indicated by the **.** However, the link to SanJose2 has been selected as the best path, indicated by the inclusion of the ">". Why did the ISP prefer the link to SanJose2 over SanJose1?

____

____

____

Would changing the bandwidth metric on each link help to correct this issue? Explain.

____

____

BGP operates differently than all other protocols. Unlike other routing protocols that use complex algorithms involving factors such as bandwidth, delay, reliability, and load to formulate a metric, BGP is policy-based. BGP determines the best path based on variables, such as AS path, weight, local preference, MED, and so on. If all things are equal, BGP prefers the route leading to the BGP speaker with the lowest BGP router ID. The SanJose2 router with BGP router ID 172.16.32.1 was preferred to the higher BGP router ID of the SanJose1 router (172.16.64.1).

j. At this point, the ISP router should be able to get to each network connected to SanJose1 and SanJose2 from the loopback address 192.168.100.1. Use the extended **ping** command and specify the source address of ISP Lo0 to test.

```
ISP# ping 172.16.1.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms
```

```
ISP# ping 172.16.32.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

ISP# ping 172.16.1.2 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
ISP#

ISP# ping 172.16.64.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms
```

You can also use the extended ping dialogue to specify the source address, as shown in this example.

```
ISP# ping
Protocol [ip]:
Target IP address: 172.16.64.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.100.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/24 ms
ISP#
```

Complete reachability has been demonstrated between the ISP router and both SanJose1 and SanJose2.

Step 7: Configure the BGP next-hop-self feature.

SanJose1 is unaware of the link between ISP and SanJose2, and SanJose2 is unaware of the link between ISP and SanJose1. Before ISP can successfully ping all the internal serial interfaces of AS 64512, these serial links should be advertised via BGP on the ISP router. This can also be resolved via EIGRP on each SanJose router. One method is for ISP to advertise these links.

a. Issue the following commands on the ISP router.

```
ISP(config)# router bgp 200
ISP(config-router)# network 192.168.1.0 mask 255.255.255.252
ISP(config-router)# network 192.168.1.4 mask 255.255.255.252
```

b. Issue the **show ip bgp** command to verify that the ISP is correctly injecting its own WAN links into BGP.

```
ISP# show ip bgp
BGP table version is 5, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *   172.16.0.0       192.168.1.6              0             0 64512 i
 *>                   192.168.1.2              0             0 64512 i
 *>  192.168.1.0/30   0.0.0.0                  0         32768 i
 *>  192.168.1.4/30   0.0.0.0                  0         32768 i
 *>  192.168.100.0    0.0.0.0                  0         32768 i
ISP#
```

c. Verify on SanJose1 and SanJose2 that the opposite WAN link is included in the routing table. The output from SanJose2 is as follows.

```
SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:52:03, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
B        192.168.1.4/30 [20/0] via 192.168.1.1, 00:01:03
B     192.168.100.0/24 [20/0] via 192.168.1.1, 00:25:20
SanJose2#
```

The next issue to consider is BGP policy routing between autonomous systems. The next-hop attribute of a route in a different AS is set to the IP address of the border router in the next AS toward the destination, and this attribute is not modified by default when advertising this route through IBGP. Therefore, for all IBGP peers, it is either necessary to know the route to that border router (in a different neighboring AS), or our own border router needs to advertise the foreign routes using the next-hop-self feature, overriding the next-hop address with its own IP address. The SanJose2 router is passing a policy to SanJose1 and vice versa. The policy for routing from AS 64512 to AS 200 is to forward packets to the 192.168.1.1 interface. SanJose1 has a similar yet opposite policy: it forwards requests to the 192.168.1.5 interface. If either WAN link fails, it is critical that the opposite router become a valid gateway. This is achieved if the **next-hop-self** command is configured on SanJose1 and SanJose2.

d. To better understand the **next-hop-self** command we will remove ISP advertising its two WAN links and shutdown the WAN link between ISP and SanJose2. The only possible path from SanJose2 to ISP's 192.168.100.0/24 is through SanJose1.

```
ISP(config)# router bgp 200
ISP(config-router)# no network 192.168.1.0 mask 255.255.255.252
ISP(config-router)# no network 192.168.1.4 mask 255.255.255.252
ISP(config-router)# exit
ISP(config)# interface serial 0/0/1
ISP(config-if)# shutdown
ISP(config-if)#
```

e. Display SanJose2's BGP table using the **show ip bgp** command and the IPv4 routing table with **show ip route**.

```
SanJose2# show ip bgp
BGP table version is 1, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 * i 192.168.100.0    192.168.1.5              0    100      0 200 i
SanJose2#


SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 02:41:46, Serial0/0/1
SanJose2#
```

Notice that SanJose2 has 192.168.100.0 in it's BGP table but not in its routing table. The BGP table shows the next hop to 192.168.100.0 as 192.168.1.5. Because SanJose2 does not have a route to this next hop address of 192.168.1.5 in its routing table, it will not install the 192.168.100.0 network into the routing table. It won't install a route if it doesn't know how to get to the next hop.

EBGP next hop addresses are carried into IBGP unchanged. As we saw previously, we could advertise the WAN link using BGP, but this is not always desirable. It means advertising additional routes when we are usually trying to minimize the size of the routing table. Another option is to have the routers within the IGP domain advertise themselves as the next hop router using the **next-hop-self** command.

f. Issue the **next-hop-self** command on SanJose1 and SanJose2 to advertise themselves as the next hop to their IBGP peer.

```
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 172.16.32.1 next-hop-self

SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 172.16.64.1 next-hop-self
```

g. Reset BGP operation on either router with the **clear ip bgp *** command.

```
SanJose1# clear ip bgp *
SanJose1#

SanJose2# clear ip bgp *
SanJose2#
```

h. After the routers have returned to established BGP speakers, issue the **show ip bgp** command on SanJose2 and notice that the next hop is now SanJose1 instead of ISP.

```
SanJose2# show ip bgp
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  172.16.0.0       0.0.0.0                  0         32768 i
 * i                  172.16.64.1              0    100      0 i
 *>i 192.168.100.0    172.16.64.1              0    100      0 200 i
SanJose2#
```

i. The **show ip route** command on SanJose2 now displays the 192.168.100.0/24 network because SanJose1 is the next hop, 172.16.64.1, which is reachable from SanJose2.

```
SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
```

```
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 04:27:19, Serial0/0/1
B     192.168.100.0/24 [200/0] via 172.16.64.1, 00:00:46
SanJose2#
```

j. Before configuring the next BGP attribute, restore the WAN link between ISP and SanJose3. This will change the BGP table and routing table on both routers. For example, SanJose2's routing table shows 192.168.100.0/24 will now have a better path through ISP.

```
ISP(config)# interface serial 0/0/1
ISP(config-if)# no shutdown
ISP(config-if)#

SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 04:37:34, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
B     192.168.100.0/24 [20/0] via 192.168.1.1, 00:01:35
SanJose2#
```

Step 8: Set BGP local preference.

At this point, everything looks good, with the exception of default routes, the outbound flow of data, and inbound packet flow.

a. Because the local preference value is shared between IBGP neighbors, configure a simple route map that references the local preference value on SanJose1 and SanJose2. This policy adjusts outbound traffic to prefer the link off the SanJose1 router instead of the metered T1 off SanJose2.

```
SanJose1(config)# route-map PRIMARY_T1_IN permit 10
SanJose1(config-route-map)# set local-preference 150
SanJose1(config-route-map)# exit
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 192.168.1.5 route-map PRIMARY_T1_IN in
```

```
SanJose2(config)# route-map SECONDARY_T1_IN permit 10
SanJose2(config-route-map)# set local-preference 125
```

```
SanJose1(config-route-map)# exit
SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
```

b. Use the **clear ip bgp * soft** command after configuring this new policy. When the conversations have been reestablished, issue the **show ip bgp** command on SanJose1 and SanJose2.

```
SanJose1# clear ip bgp * soft
SanJose2# clear ip bgp * soft

SanJose1# show ip bgp
BGP table version is 3, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.32.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>  192.168.100.0    192.168.1.5              0    150      0 200 i
SanJose1#

SanJose2# show ip bgp
BGP table version is 7, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 192.168.100.0    172.16.64.1              0    150      0 200 i
 *                    192.168.1.1              0    125      0 200 i
SanJose2#
```

This now indicates that routing to the loopback segment for ISP 192.168.100.0 /24 can be reached only through the link common to SanJose1 and ISP. SanJose2's next hop to 192.168.100.0/24 is SanJose1 because both routers have been configured using the **next-hop-self** command.

Step 9: Set BGP MED.

a. In the previous step we saw that SanJose1 and SanJose2 will route traffic for 192.168.100.0/24 using the link between SanJose1 and ISP. Examine what the return path ISP takes to reach AS 64512. Notice that the return path is different from the original path. This is known as asymmetric routing and is not necessarily an unwanted trait.

```
ISP# show ip bgp
BGP table version is 22, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *   172.16.0.0       192.168.1.6              0             0 64512 i
 *>                   192.168.1.2              0             0 64512 i
 *>  192.168.100.0    0.0.0.0                  0         32768 i
ISP# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

B      172.16.0.0/16 [20/0] via 192.168.1.2, 00:12:45
      192.168.1.0/24 is variably subnetted, 4 subnets, 2 masks
C  192.168.1.0/30 is directly connected, Serial0/0/1
L        192.168.1.1/32 is directly connected, Serial0/0/1
C        192.168.1.4/30 is directly connected, Serial0/0/0
L  192.168.1.5/32 is directly connected, Serial0/0/0
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C  192.168.100.0/24 is directly connected, Loopback0
L      192.168.100.1/32  is  directly  connected,  Loopback0
ISP#
```

How will traffic from network 192.168.100.0 /24 on ISP return to SanJose1 or SanJose2? Will it be routed through SanJose1 or SanJose2?

____

____

____

To verify this, the simplest solution is to issue the **show ip bgp** command on the ISP router as was done above. What if access was not given to the ISP router? Traffic returning from the Internet should not be passed across the metered T1. Is there a simple way to verify before receiving the monthly bill? How can it be checked instantly?

____

____

____

a. Use an extended **ping** command to verify this situation. Specify the **record** option and compare your output to the following. Notice the return path using the exit interface 192.168.1.1 to SanJose2.

```
SanJose2# ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
```

```
Datagram size [100]:
```

```
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options: Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)

Reply to request 0 (20 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 1 (20 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 2 (20 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 3 (24 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 4 (20 ms). Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/24 ms
SanJose2#
```

If you are unfamiliar with the **record** option, the important thing to note is that each IP address in brackets is an outgoing interface. The output can be interpreted as follows:

1. A ping that is sourced from 172.16.32.1 exits SanJose2 through s0/0/1, 172.16.1.2. It then arrives at the s0/0/1 interface for SanJose1.
2. SanJose1 S0/0/0, 192.168.1.6, routes the packet out to arrive at the S0/0/0 interface of ISP.
3. The target of 192.168.100.1 is reached: 192.168.100.1.
4. The packet is next forwarded out the S0/0/1, 192.168.1.1 interface for ISP and arrives at the S0/0/0 interface for SanJose2.
5. SanJose2 then forwards the packet out the last interface, loopback 0, 172.16.32.1.

Although the unlimited use of the T1 from SanJose1 is preferred here, ISP currently takes the link from SanJose2 for all return traffic.

b. Utilize the MED (metric) that is shared between EBGP neighbors.
Create the MED policy for this ISP router to return traffic via SanJose1. Create a second route map

```
SanJose1(config)#route-map PRIMARY_T1_MED_OUT permit 10
SanJose1(config-route-map)#set Metric 50
SanJose1(config-route-map)#exit
SanJose1(config)#router bgp 64512
SanJose1(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out

SanJose2(config)#route-map SECONDARY_T1_MED_OUT permit 10
SanJose2(config-route-map)#set Metric 75
SanJose2(config-route-map)#exit
SanJose2(config)#router bgp 64512
SanJose2(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
```

c. Use the **clear ip bgp * soft** command after issuing this new policy. Issuing the **show ip bgp** command as follows on SanJose1 or SanJose2 does not indicate anything about this newly defined policy.

```
SanJose1# clear ip bgp * soft
SanJose2# clear ip bgp * soft

SanJose1# show ip bgp
BGP table version is 4, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.32.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>  192.168.100.0    192.168.1.5              0    150      0 200 i
SanJose1#

SanJose2# show ip bgp
BGP table version is 8, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 192.168.100.0    172.16.64.1              0    150      0 200 i
 *                    192.168.1.1              0    125      0 200 i
SanJose2#
```

d. Reissue an extended **ping** command with the **record** command. Notice the change in return path using the exit interface 192.168.1.5 to SanJose1.

```
SanJose2# ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
```

```
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)

Reply to request 0 (28 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 1 (28 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 2 (28 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
```

```
      (172.16.32.1) <*>

      (0.0.0.0)
      (0.0.0.0)
      (0.0.0.0)
    End of list

  Reply to request 3 (28 ms).  Received packet has options
   Total option bytes= 40, padded length=40
   Record route:
      (172.16.1.2)
      (192.168.1.6)
      (192.168.100.1)
      (192.168.1.5)
      (172.16.1.1)
      (172.16.32.1) <*>
      (0.0.0.0)
      (0.0.0.0)
      (0.0.0.0)
   End of list

  Reply to request 4 (28 ms).  Received packet has options
   Total option bytes= 40, padded length=40
   Record route:
      (172.16.1.2)
      (192.168.1.6)
      (192.168.100.1)
      (192.168.1.5)
      (172.16.1.1)
      (172.16.32.1) <*>
      (0.0.0.0)
      (0.0.0.0)
      (0.0.0.0)
   End of list

  Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
  SanJose2#
```

Does the output look correct? Does the 192.168.1.5 above mean that the ISP now prefers SanJose1 for return traffic

The newly configured policy MED shows that the lower MED value is considered best. The ISP now prefers the route with the lower MED value of 50 to AS 64512. This is just opposite from the **local-preference** command configured earlier.

```
ISP# show ip bgp
BGP table version is 24, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  172.16.0.0       192.168.1.6             50             0 64512 i
 *                    192.168.1.2             75             0 64512 i
 *>  192.168.100.0    0.0.0.0                  0         32768 i
```

```
ISP#
```

Step 10: Establish a default route.

The final step is to establish a default route that uses a policy statement that adjusts to changes in the network.

a. Configure ISP to inject a default route to both SanJose1 and SanJose2 using BGP using the **default-originate** command. This command does not require the presence of 0.0.0.0 in the ISP router. Configure the 10.0.0.0/8 network which will not be advertised using BGP. This network will be used to test the default route on SanJose1 and SanJose2.

```
ISP(config)# router bgp 200
ISP(config-router)# neighbor 192.168.1.6 default-originate
ISP(config-router)# neighbor 192.168.1.2 default-originate
ISP(config-router)# exit
ISP(config)# interface loopback 10
ISP(config-if)# ip address 10.0.0.1 255.255.255.0
ISP(config-if)#
```

b. Verify that both routers have received the default route by examining the routing tables on SanJose1 and SanJose2. Notice that both routers prefer the route between SanJose1 and ISP.

```
SanJose1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

B*    0.0.0.0/0 [20/0] via 192.168.1.5, 00:00:36
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.1/32 is directly connected, Serial0/0/1
D        172.16.32.0/24 [90/2297856] via 172.16.1.2, 05:47:24, Serial0/0/1
C        172.16.64.0/24 is directly connected, Loopback0
L        172.16.64.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.4/30 is directly connected, Serial0/0/0
L        192.168.1.6/32 is directly connected, Serial0/0/0
SanJose1#

SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override
```

```
Gateway of last resort is 172.16.64.1 to network 0.0.0.0

B*    0.0.0.0/0 [200/0] via 172.16.64.1, 00:00:45
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 05:47:33, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
SanJose2#
```

c. The preferred default route is by way of SanJose1 because of the higher local preference attribute configured on SanJose1 earlier.

```
SanJose2# show ip bgp
BGP table version is 38, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 0.0.0.0          172.16.64.1              0    150      0 200 i
 *                    192.168.1.1                   125      0 200 i
 * i 172.16.0.0       172.16.64.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
 *>i 192.168.100.0    172.16.64.1              0    150      0 200 i
 *                    192.168.1.1              0    125      0 200 i
SanJose2#
```

d. Using the traceroute command verify that packets to 10.0.0.1 is using the default route through SanJose1.

```
SanJose2# traceroute 10.0.0.1
Type escape sequence to abort.
Tracing the route to 10.0.0.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.1.1 8 msec 4 msec 8 msec
  2 192.168.1.5 [AS 200] 12 msec *  12 msec
SanJose2#
```

e. Next, test how BGP adapts to using a different default route when the path between SanJose1 and ISP goes down.

```
ISP(config)# interface serial 0/0/0
ISP(config-if)# shutdown
ISP(config-if)#
```

f. Verify that both routers are modified their routing tables with the default route using the path between SanJose2 and ISP.

```
SanJose1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

B*    0.0.0.0/0 [200/0] via 172.16.32.1, 00:00:06
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.1/32 is directly connected, Serial0/0/1
D        172.16.32.0/24 [90/2297856] via 172.16.1.2, 05:49:25, Serial0/0/1
C        172.16.64.0/24 is directly connected, Loopback0
L        172.16.64.1/32 is directly connected, Loopback0
B  192.168.100.0/24 [200/0] via 172.16.32.1, 00:00:06
SanJose1#

SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

B*    0.0.0.0/0 [20/0] via 192.168.1.1, 00:00:30
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.32.0/24 is directly connected, Loopback0
L        172.16.32.1/32 is directly connected, Loopback0
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 05:49:49, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
B     192.168.100.0/24 [20/0] via 192.168.1.1, 00:00:30
SanJose2#
```

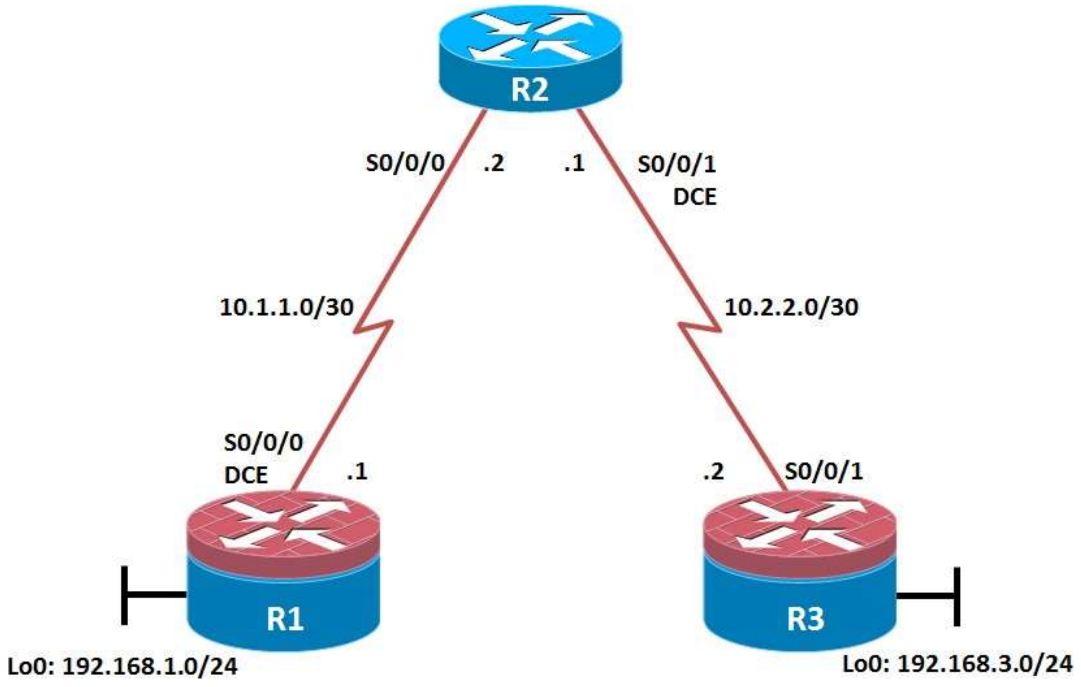
g. Verify the new path using the traceroute command to 10.0.0.1 from SanJose1. Notice the default route is now through SanJose2.

```
SanJose1# trace 10.0.0.1
Type escape sequence to abort.
Tracing the route to 10.0.0.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.1.2 8 msec 8 msec 8 msec
  2 192.168.1.1 [AS 200] 12 msec *  12 msec
SanJose1#
```

Practical No.04

Aim:Secure the Management Plane

Topology



Objectives

- Secure management access.
- Configure enhanced username password security.
- Enable AAA RADIUS authentication.
- Enable secure remote management.

Background

The management plane of any infrastructure device should be protected as much as possible. Controlling access to routers and enabling reporting on routers are critical to network security and should be part of a comprehensive security policy.

In this lab, you build a multi-router network and secure the management plane of routers R1 and R3.

Note: This lab uses Cisco 1941 routers with Cisco IOS Release 15.2 with IP Base. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco IOS Release 15.2 or comparable)
- Serial and Ethernet cables

Step 1: Configure loopbacks and assign addresses.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations. Using the addressing scheme in the diagram, apply the IP addresses to the interfaces on the R1, R2, and R3 routers.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter the designations accordingly.

R1

```
hostname R1

interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0
exit
!
interface Serial0/0/0
 description R1 --> R2
 ip address 10.1.1.1 255.255.255.252
 clock rate 128000
 no shutdown
exit
!
end
```

R2

```
hostname R2
!
interface Serial0/0/0
 description R2 --> R1
 ip address 10.1.1.2 255.255.255.252
 no shutdown
exit

interface Serial0/0/1
 description R2 --> R3
 ip address 10.2.2.1 255.255.255.252
 clock rate 128000
 no shutdown
exit
!
end
```

R3

```
hostname R3
!
interface Loopback0
 description R3 LAN
 ip address 192.168.3.1 255.255.255.0
exit


interface Serial0/0/1
 description R3 --> R2
 ip address 10.2.2.2 255.255.255.252
 no shutdown
exit
!
end
```

Step 2: Configure static routes.

a. On R1, configure a default static route to ISP.

i. `R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.2`

b. On R3, configure a default static route to ISP.

i. `R3(config)# ip route 0.0.0.0 0.0.0.0 10.2.2.1`

c. On R2, configure two static routes.

i. `R2(config)# ip route 192.168.1.0 255.255.255.0 10.1.1.1`

ii. `R2(config)# ip route 192.168.3.0 255.255.255.0 10.2.2.2`

d. From the R1 router, run the following Tcl script to verify connectivity.

```
foreach address {
192.168.1.1
10.1.1.1
10.1.1.2
10.2.2.1
10.2.2.2
192.168.3.1
} { ping $address }
```

```
R1# tclsh

R1(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#10.1.1.1
+>(tcl)#10.1.1.2
+>(tcl)#10.2.2.1
+>(tcl)#10.2.2.2
+>(tcl)#192.168.3.1
+>(tcl)#} { ping $address }
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
R1(tcl)#
```

Are the pings now successful?

____

____

Yes. If not, troubleshoot.

Step 3: Secure management access.

a. On R1, use the **security passwords** command to set a minimum password length of 10 characters.

```
R1(config)# security passwords min-length 10
```

b. Configure the enable secret encrypted password on both routers.

```
R1(config)# enable secret class12345
```

How does configuring an enable secret password help protect a router from being compromised by an attack?

____

____

____

The goal is to always prevent unauthorized users from accessing a device using Telnet, SSH, or via the console. If attackers are able to penetrate this first layer of defense, using an enable secret password prevents them from being able to alter the configuration of the device. Unless the enable secret password is known, a user cannot go into privileged EXEC mode where they can display the running config and enter various configuration commands to make changes to the router. This provides an additional layer of security.

Note: Passwords in this task are set to a minimum of 10 characters but are relatively simple for the benefit of performing the lab. More complex passwords are recommended in a production network.

c. Configure a console password and enable login for routers. For additional security, the **exec-timeout** command causes the line to log out after 5 minutes of inactivity. The **logging synchronous** command prevents console messages from interrupting command entry.

Note: To avoid repetitive logins during this lab, the **exec-timeout** command can be set to 0 0, which prevents it from expiring. However, this is not considered a good security practice.

```
R1(config)# line console 0
R1(config-line)# password ciscoconpass
R1(config-line)# exec-timeout 5 0
R1(config-line)# login
R1(config-line)# logging synchronous
R1(config-line)# exit
R1(config)#
```

d. Configure the password on the vty lines for router R1.

```
R1(config)# line vty 0 4
R1(config-line)# password ciscovtypass
R1(config-line)# exec-timeout 5 0
R1(config-line)# login
R1(config-line)# exit
R1(config)#
```

e. The aux port is a legacy port used to manage a router remotely using a modem and is hardly ever used. Therefore, disable the aux port.

```
R1(config)# line aux 0
R1(config-line)# no exec
R1(config-line)#  end
R1#
```

f. Enter privileged EXEC mode and issue the **show run** command. Can you read the enable secret password? Why or why not?

____

____

No. The enable secret password is encrypted automatically using the MD5 or SHA hash algorithm. . IOS 15.0(1)S and later default to SHA256 hashing algorithm. SHA256 which is considered to be a very strong hashing algorithm and is extremely difficult to reverse. Earlier IOS versions use the weaker MD5 hashing algorithm.

Note: If the **enable secret** password command is lost or forgotten, it must be replaced using the Cisco router password recovery procedure. Refer to cisco.com for more information.

Can you read the console, aux, and vty passwords? Why or why not?

____

____

Yes. They are all in clear text.

g. Use the **service password-encryption** command to encrypt the line console and vty passwords.

```
R1(config)# service password-encryption
R1(config)#
```

Note: Password encryption is applied to all the passwords, including the **username** passwords, the authentication key passwords, the privileged command password, the console and the virtual terminal line access passwords, and the BGP neighbor passwords.

h. Issue the **show run** command. Can you read the console, aux, and vty passwords? Why or why not?

____

____

No. The passwords are now encrypted.

Note: Type 7 passwords are encrypted using a Vigenère cipher which can be easily reversed. Therefore this command primarily protects from shoulder surfing attacks.

i. Configure a warning to unauthorized users with a message-of-the-day (MOTD) banner using the **banner motd** command. When a user connects to one of the routers, the MOTD banner appears before the login prompt. In this example, the dollar sign ($) is used to start and end the message.

```
R1(config)# banner motd $Unauthorized access strictly prohibited!$
R1(config)# exit
```

j. Issue the **show run** command. What does the $ convert to in the output?

____

____

The $ is converted to ^C when the running-config is displayed.

k. Exit privileged EXEC mode using the **disable** or **exit** command and press **Enter** to get started. Does the MOTD banner look like what you created with the **banner motd** command? If the MOTD banner is not as you wanted it, recreate it using the **banner motd** command.

h.

l. Repeat the configuration portion of steps 3a through 3k on router R3.

Step 4: Configure enhanced username password security.

To increase the encryption level of console and VTY lines, it is recommended to enable authentication using the local database. The local database consists of usernames and password combinations that are created locally on each device. The local and VTY lines are configured to refer to the local database when authenticating a user.

a. To create local database entry encrypted to level 4 (SHA256), use the **username** *name* **secret** *password* global configuration command. In global configuration mode, enter the following command:

```
R1(config)# username JR-ADMIN secret class12345
R1(config)# username ADMIN secret class54321
```

Note: An older method for creating local database entries is to use the **username** *name* **password**

password command.

b. Set the console line to use the locally defined login accounts.

```
R1(config)# line console 0
R1(config-line)# login local
R1(config-line)# exit
R1(config)#
```

c. Set the vty lines to use the locally defined login accounts.

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# end
R1(config)#
```

d. Repeat the steps 4a to 4c on R3.

i.

e. To verify the configuration, telnet to R3 from R1 and login using the ADMIN local database account.

```
R1# telnet 10.2.2.2
Trying 10.2.2.2 ... Open
Unauthorized access strictly prohibited!
User Access Verification

Username: ADMIN
Password:
R3>
```

Step 5: Enabling AAA RADIUS Authentication with Local User for Backup.

Authentication, authorization, and accounting (AAA) is a standards-based framework that can be implemented to control who is permitted to access a network (authenticate), what they can do on that network (authorize), and audit what they did while accessing the network (accounting).

Users must authenticate against an authentication database which can be stored:

- **Locally**: Users are authenticated against the local device database which is created using the username secret command. Sometimes referred to self-contained AAA.
- **Centrally**: A client-server model where users are authenticated against AAA servers. This provides improved scalability, manageability and control. Communication between the device and AAA servers is secured using either the RADIUS or TACACS+ protocols.

In this step, we will configure AAA authentication to use a RADIUS server and the local database as a backup. Specifically, the authentication will be validated against one of two RADIUS servers. If the servers are not available, then authentication will be validated against the local database.

a. Always have local database accounts created before enabling AAA. Since we created two local database accounts in the previous step, then we can proceed and enable AAA on R1.

```
R1(config)# aaa new-model
```

Note: Although the following configuration refers to two RADIUS servers, the actual RADIUS server implementation is beyond the scope. Therefore, the goal of this step is to provide an example of how to configure a router to access the servers.

b. Configure the specifics for the first RADIUS server located at 192.168.1.101. Use **RADIUS-1-pa55w0rd** as the server password.

```
R1(config)# radius server RADIUS-1
R1(config-radius-server)# address ipv4 192.168.1.101
R1(config-radius-server)# key RADIUS-1-pa55w0rd
R1(config-radius-server)# exit
R1(config)#
```

c. Configure the specifics for the second RADIUS server located at 192.168.1.102. Use **RADIUS-2-pa55w0rd** as the server password.

```
R1(config)# radius server RADIUS-2
R1(config-radius-server)# address ipv4 192.168.1.102
R1(config-radius-server)# key RADIUS-2-pa55w0rd
R1(config-radius-server)# exit
R1(config)#
```

d. Assign both RADIUS servers to a server group.

```
R1(config)# aaa group server radius RADIUS-GROUP
R1(config-sg-radius)# server name RADIUS-1
R1(config-sg-radius)# server name RADIUS-2
R1(config-sg-radius)# exit
R1(config)#
```

e. Enable the default AAA authentication login to attempt to validate against the server group. If they are not available, then authentication should be validated against the local database..

```
R1(config)# aaa authentication login default group RADIUS-GROUP local
R1(config)#
```

Note: Once this command is configured, all line access methods default to the default authentication method. The **local** option enables AAA to refer to the local database. Only the password is case sensitive.

f. Enable the default AAA authentication Telnet login to attempt to validate against the server group. If they are not available, then authentication should be validated against a case sensitive local database.

```
R1(config)# aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case
R1(config)#
```

Note: Unlike the **local** option that makes the password is case sensitive, local-case makes the username and password case sensitive.

g. Alter the VTY lines to use the TELNET-LOGIN AAA authentiaito0n method.

```
R1(config)# line vty 0 4
R1(config-line)# login authentication TELNET-LOGIN
R1(config-line)# exit
R1(config)#
```

h. Repeat the steps 5a to 5g on R3.

j.

i. To verify the configuration, telnet to R3 from R1 and login using the ADMIN local database account.

```
R1# telnet 10.2.2.2
Trying 10.2.2.2 ... Open
Unauthorized access strictly prohibited!

User Access Verification

Username: admin
Password:

% Authentication failed

Username: ADMIN
Password:

R3>
```

Note: The first login attempt did not use the correct username (i.e., ADMIN) which is why it failed.

Note: The actual login time is longer since the RADIUS servers are not available.

Step 6: Enabling secure remote management using SSH.

Traditionally, remote access on routers was configured using Telnet on TCP port 23. However, Telnet was developed in the days when security was not an issue; therefore, all Telnet traffic is forwarded in plaintext.

Secure Shell (SSH) is a network protocol that establishes a secure terminal emulation connection to a router or other networking device. SSH encrypts all information that passes over the network link and provides authentication of the remote computer. SSH is rapidly replacing Telnet as the remote login tool of choice for network professionals.

Note: For a router to support SSH, it must be configured with local authentication, (AAA services, or username) or password authentication. In this task, you configure an SSH username and local authentication.

In this step, you will enable R1 and R3 to support SSH instead of Telnet.

a. SSH requires that a device name and a domain name be configured. Since the router already has a name assigned, configure the domain name.

```
R1(config)# ip domain-name ccnasecurity.com
```

b. The router uses the RSA key pair for authentication and encryption of transmitted SSH data. Although optional it may be wise to erase any existing key pairs on the router.

```
R1(config)# crypto key zeroize rsa
```

Note: If no keys exist, you might receive this message: `% No Signature RSA Keys found in configuration`.

c. Generate the RSA encryption key pair for the router. Configure the RSA keys with **1024** for the number of modulus bits. The default is 512, and the range is from 360 to 2048.

```
R1(config)# crypto key generate rsa general-keys modulus 1024
The name for the keys will be: R1.ccnasecurity.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

R1(config)#
Jan 10 13:44:44.711: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#
```

k.

d. Cisco routers support two versions of SSH:

- **SSH version 1 (SSHv1)**: Original version but has known vulnerabilities.
- **SSH version 2 (SSHv2)**: Provides better security using the Diffie-Hellman key exchange and the strong integrity-checking message authentication code (MAC).

l. The default setting for SSH is SSH version 1.99. This is also known as compatibility mode and is merely an indication that the server supports both SSH version 2 and SSH version 1. However, best practices are to enable version 2 only.

m. Configure SSH version 2 on R1.

```
R1(config)# ip ssh version 2
R1(config)#
```

n.

e. Configure the vty lines to use only SSH connections.

```
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
R1(config-line)# end
```

Note: SSH requires that the **login local** command be configured. However, in the previous step we enabled AAA authentication using the TELNET-LOGIN authentication method, therefore **login local** is not necessary.

Note: If you add the keyword **telnet** to the **transport input** command, users can log in using Telnet as well as SSH. However, the router will be less secure. If only SSH is specified, the connecting host must have an SSH client installed.

f. Verify the SSH configuration using the **show ip ssh** command.

```
R1# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQC3Lehh7ReYlgyDzls6wq+mFzxqzoaZFr9XGx+Q/yio
dFYw00hQo80tZy1W1Ff3Pz6q7Qi0y00urwddHZ0kBZceZK9EzJ6wZ+9a87KKDETCWrGSLi6c8lE/y4K+
Z/oVrMMZk7bpTM1MFdP41YgkTf35utYv+TcqbsYo++KJiYk+xw==
R1#
```

g. Repeat the steps 6a to 6f on R3.

o.

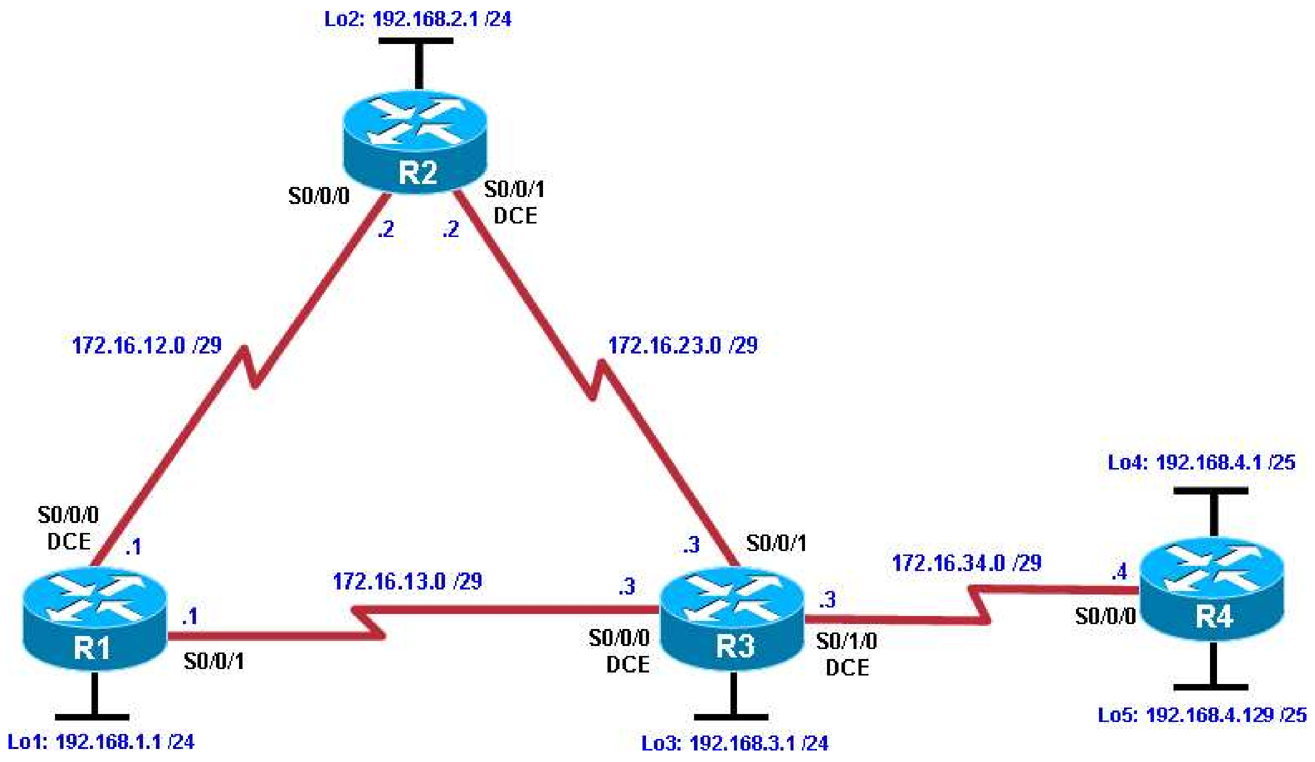
h. Although a user can SSH from a host using the SSH option of TeraTerm of PuTTY, a router can also SSH to another SSH enabled device. SSH to R3 from R1.

```
R1# ssh -l ADMIN 10.2.2.2
Password:
Unauthorized access strictly prohibited!
R3>
R3> en
Password:
R3#
```

Practical No.05

Aim: Configure and Verify Path Control

Topology



Objectives

- Configure and verify policy-based routing.
- Select the required tools and commands to configure policy-based routing operations.
- Verify the configuration and operation by using the proper **show** and **debug** commands.

Background

You want to experiment with policy-based routing (PBR) to see how it is implemented and to study how it could be of value to your organization. To this end, you have interconnected and configured a test network with four routers. All routers are exchanging routing information using EIGRP.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1, and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router and software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration, and reload each router to clear previous configurations.

Step 2: Configure router hostname and interface addresses.

a. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to these and the serial interfaces on R1, R2, R3, and R4. On the serial interfaces connecting R1 to R3 and R3 to R4, specify the bandwidth as 64 Kb/s and set a clock rate on the DCE using the **clock rate 64000** command. On the serial interfaces connecting R1 to R2 and R2 to R3, specify the bandwidth as 128 Kb/s and set a clock rate on the DCE using the **clock rate 128000** command.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
hostname R1
!
interface Lo1  description
  R1 LAN
  ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/0
  description R1 --> R2
  ip address 172.16.12.1 255.255.255.248
  clock rate 128000
  bandwidth  128no
  shutdown
!
interface Serial0/0/1
  description R1 --> R3
  ip address 172.16.13.1 255.255.255.248
  bandwidth 64no
  shutdown
!
end
```

Router R2

```
hostname R2
!
interface Lo2  description
  R2 LAN
  ip address 192.168.2.1 255.255.255.0
!
interface Serial0/0/0
  description R2 --> R1
  ip address 172.16.12.2 255.255.255.248
  bandwidth 128no
  shutdown

interface Serial0/0/1
  description R2 --> R3
  ip address 172.16.23.2 255.255.255.248
  clock rate 128000
  bandwidth 128no
  shutdown
!
end
```

Router R3

```
hostname R3
!
interface Lo3 description
  R3 LAN
  ip address 192.168.3.1 255.255.255.0
!
interface Serial0/0/0
  description R3 --> R1
  ip address 172.16.13.3 255.255.255.248
  clock rate 64000
  bandwidth  64no
  shutdown
!
interface Serial0/0/1
  description R3 --> R2
  ip address 172.16.23.3 255.255.255.248
  bandwidth 128no
  shutdown
!
interface Serial0/1/0
  description R3 --> R4
  ip address 172.16.34.3 255.255.255.248
  clock rate 64000
  bandwidth  64no
  shutdown
!
end
```

Router R4

```
hostname R4
!
interface Lo4 description R4
  LAN A
  ip address 192.168.4.1 255.255.255.128
!
interface Lo5 description R4
  LAN B
  ip address 192.168.4.129 255.255.255.128
!
interface Serial0/0/0
  description R4 --> R3
  ip address 172.16.34.4 255.255.255.248
  bandwidth 64no
  shutdown
!
end
```

b. Verify the configuration with the **show ip interface brief**, **show protocols,** and **show interfaces description** commands. The output from router R3 is shown here as an example.

```
R3# show ip interface brief
Interface          IP-Address     OK? Method Status                    Protocol
FastEthernet0/0    unassigned     YES manual administratively down down

FastEthernet0/1    unassigned     YES unset administratively down down

Serial0/0/0        172.16.13.3    YES manual up                        up

Serial0/0/1        172.16.23.3    YES manual up                        up

Serial0/1/0        172.16.34.3    YES manual up                        up

Serial0/1/1        unassigned     YES unset administratively down down

Loopback3          192.168.3.1    YES manual up                        up
```

```
R3# show protocols
Global values:
   Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down FastEthernet0/1 is
administratively down, line protocol is downSerial0/0/0 is up, line protocol is up
Internet address is 172.16.13.3/29 Serial0/0/1 is up,
line protocol is upInternet address is 172.16.23.3/29
Serial0/1/0 is up, line protocol is upInternet address
                  is 172.16.34.3/29
            Serial0/1/1 is administratively down, line protocol is down
Loopback3 is up, line protocol is upInternet
   address is 192.168.3.1/24
```

```
R3# show interfaces description
Interface                       Status          Protocol Description
Fa0/0                           admin down      down
Fa0/1                           admin down      down
Se0/0/0                         up              up       R3 --> R1
Se0/0/1                         up              up       R3 --> R2
Se0/1/0                         up              up       R3 --> R4
Se0/1/1                         admin down      down
Lo3                             up              up       R3 LAN
```

Step 3: Configure basic EIGRP.

a. Implement EIGRP AS 1 over the serial and loopback interfaces as you have configured it for the other EIGRP labs.

b. Advertise networks 172.16.12.0/29, 172.16.13.0/29, 172.16.23.0/29, 172.16.34.0/29, 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24, and 192.168.4.0/24 from their respective routers.

You can copy and paste the following configurations into your routers.

Router R1

```
router eigrp 1
 network 192.168.1.0
 network 172.16.12.0 0.0.0.7
 network 172.16.13.0 0.0.0.7
 no auto-summary
```

Router R2

```
router eigrp 1
  network 192.168.2.0
  network 172.16.12.0 0.0.0.7
  network 172.16.23.0 0.0.0.7
  no auto-summary
```

Router R3

```
router eigrp 1
  network 192.168.3.0
  network 172.16.13.0 0.0.0.7
  network 172.16.23.0 0.0.0.7
  network 172.16.34.0 0.0.0.7
  no auto-summary
```

Router R4

```
router eigrp 1
  network 192.168.4.0
  network 172.16.34.0 0.0.0.7
  no auto-summary
```

You should see EIGRP neighbor relationship messages being generated.

Step 4: Verify EIGRP connectivity.

a. Verify the configuration by using the **show ip eigrp neighbors** command to check which routers have EIGRP adjacencies.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface        Hold Uptime   SRTT   RTO  Q  Seq
                                             (sec)         (ms)       Cnt Num
1   172.16.13.3             Se0/0/1            12 00:00:58  127  2280  0  16
0   172.16.12.2             Se0/0/0            13 00:01:20    8  1140  0  17

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface        Hold Uptime   SRTT   RTO  Q  Seq
                                             (sec)         (ms)       Cnt Num
1   172.16.23.3             Se0/0/1            10 00:01:30   15  1140 0  17
0   172.16.12.1             Se0/0/0            11 00:01:43   14  1140  0  180

R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface        Hold Uptime   SRTT   RTO  Q  Seq
                                             (sec)         (ms)       Cnt Num
2   172.16.34.4             Se0/1/0            10 00:02:51   27  2280  0  3
0   172.16.13.1             Se0/0/0            12 00:03:08   45  2280  0  19
1   172.16.23.2             Se0/0/1            12 00:03:13   12  1140  0  16

R4# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface        Hold Uptime   SRTT   RTO  Q  Seq
                                             (sec)         (ms)       Cnt Num
0   172.16.34.3             Se0/0/0            13 00:03:33   40  2280  0  15
```

Did you receive the output you expected?

b. Run the following Tcl script on all routers to verify full connectivity.

```
R1# tclsh

foreach address {
172.16.12.1
172.16.12.2
172.16.13.1
172.16.13.3
172.16.23.2
172.16.23.3
172.16.34.3
172.16.34.4
192.168.1.1
192.168.2.1
192.168.3.1
192.168.4.1
192.168.4.129
} { ping $address }
```

You should get ICMP echo replies for every address pinged. Make sure to run the Tcl script on each router.

Step 5: Verify the current path.

Before you configure PBR, verify the routing table on R1.

a. On R1, use the **show ip route** command. Notice the next-hop IP address for all networks discovered by EIGRP.

```
R1# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter areaN1 - OSPF NSSA
          external type 1, N2 - OSPF NSSA external type 2E1 - 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-2ia - IS-IS inter area, * -
          candidate default, U - per-user static
route
          o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

       172.16.0.0/29 is subnetted, 4 subnets
D          172.16.34.0 [90/41024000] via 172.16.13.3, 00:05:18, Serial0/0/1D     172.16.23.0
[90/21024000] via 172.16.12.2, 00:05:18, Serial0/0/0
C          172.16.12.0 is directly connected, Serial0/0/0C
           172.16.13.0 is directly connected, Serial0/0/1
       192.168.4.0/25 is subnetted, 2 subnets
D          192.168.4.0 [90/41152000] via 172.16.13.3, 00:05:06, Serial0/0/1 D 192.168.4.128
[90/41152000] via 172.16.13.3, 00:05:06, Serial0/0/1
C      192.168.1.0/24 is directly connected, Loopback1
D      192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:05:18, Serial0/0/0D      192.168.3.0/24
[90/21152000] via 172.16.12.2, 00:05:18, Serial0/0/0
```

b. On R4, use the **traceroute** command to the R1 LAN address and source the ICMP packet from R4 LAN A and LAN B.

Note: You can specify the source as the interface address (for example 192.168.4.1) or the interface designator (for example, Fa0/0).

```
R4# traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort. Tracing the route
to 192.168.1.1

  1 172.16.34.3 12 msec 12 msec 16 msec
  2 172.16.23.2 20 msec 20 msec 20 msec
  3 172.16.12.1 28 msec 24 msec *

R4# traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort. Tracing the route
to 192.168.1.1

  1 172.16.34.3 12 msec 12 msec 16 msec
  2 172.16.23.2 20 msec 20 msec 20 msec
  3 172.16.12.1 28 msec 24 msec *
```

Notice that the path taken for the packets sourced from the R4 LANs are going through R3 --> R2 --> R1.

Why are the R4 interfaces not using the R3 --> R1 path?

c. On R3, use the **show ip route** command and note that the preferred route from R3 to R1 LAN 192.168.1.0/24 is via R2 using the R3 exit interface S0/0/1.

```
R3# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
              D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter areaN1 - OSPF NSSA
              external type 1, N2 - OSPF NSSA external type 2E1 - 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-2ia - IS-IS inter area, * -
              candidate default, U - per-user static
route
              o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

         172.16.0.0/29 is subnetted, 4 subnets
C            172.16.34.0 is directly connected, Serial0/1/0C
             172.16.23.0 is directly connected, Serial0/0/1
D            172.16.12.0 [90/21024000] via 172.16.23.2, 00:15:07, Serial0/0/1
C            172.16.13.0 is directly connected, Serial0/0/0192.168.4.0/25 is
         subnetted, 2 subnets
D            192.168.4.0 [90/40640000] via 172.16.34.4, 00:14:55, Serial0/1/0
D            192.168.4.128 [90/40640000] via 172.16.34.4, 00:14:55, Serial0/1/0D192.168.1.0/24
[90/21152000] via 172.16.23.2, 00:15:07, Serial0/0/1 D          192.168.2.0/24 [90/20640000] via
172.16.23.2, 00:15:07, Serial0/0/1
C        192.168.3.0/24 is directly connected, Loopback3
```

d. On R3, use the **show interfaces serial 0/0/0** and **show interfaces s0/0/1** commands.

```
R3# show interfaces s0/0/0
Serial0/0/0 is up, line protocol is upHardware is
   GT96K Serial Description: R3 --> R1
   Internet address is 172.16.13.3/29
   MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec, reliability
        255/255, txload 1/255, rxload 1/255
   Encapsulation HDLC, loopback not setKeepalive set
   (10 sec)
   CRC checking enabled
   Last input 00:00:00, output 00:00:00, output hang neverLast clearing of
   "show interface" counters never
   Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0Queueing strategy: weighted
   fair
   Output queue: 0/1000/64/0 (size/max total/threshold/drops)Conversations
        0/1/256 (active/max active/max total) Reserved Conversations 0/0
        (allocated/max allocated) Available Bandwidth 48 kilobits/sec
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec771 packets input,
        53728 bytes, 0 no buffer
        Received 489 broadcasts, 0 runts, 0 giants, 0 throttles
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort768 packets output,
        54404 bytes, 0 underruns
        0 output errors, 0 collisions, 6 interface resets
        0 unknown protocol drops
        0 output buffer failures, 0 output buffers swapped out
        1 carrier transitions
        DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

R3# show interfaces s0/0/1
Serial0/0/1 is up, line protocol is upHardware is
   GT96K Serial Description: R3 --> R2
   Internet address is 172.16.23.3/29
   MTU 1500 bytes, BW 128 Kbit/sec, DLY 20000 usec, reliability 255/255, txload
        1/255, rxload 1/255
   Encapsulation HDLC, loopback not setKeepalive set
   (10 sec)
   CRC checking enabled
   Last input 00:00:00, output 00:00:01, output hang neverLast clearing of
   "show interface" counters never
   Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0Queueing strategy: weighted
   fair
   Output queue: 0/1000/64/0 (size/max total/threshold/drops)Conversations
        0/1/256 (active/max active/max total) Reserved Conversations 0/0
        (allocated/max allocated) Available Bandwidth 1158 kilobits/sec
   5 minute input rate 0 bits/sec, 0 packets/sec
   5 minute output rate 0 bits/sec, 0 packets/sec894 packets input,
        65653 bytes, 0 no buffer
        Received 488 broadcasts, 0 runts, 0 giants, 0 throttles
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort895 packets output,
        66785 bytes, 0 underruns
        0 output errors, 0 collisions, 6 interface resets
        0 unknown protocol drops
```

```
0 output buffer failures, 0 output buffers swapped out
1 carrier transitions
DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

Notice that the bandwidth of the serial link between R3 and R1 (S0/0/0) is set to 64 Kb/s, while the bandwidth of the serial link between R3 and R2 (S0/0/1) is set to 128 Kb/s.

e. Confirm that R3 has a valid route to reach R1 from its serial 0/0/0 interface using the **show ip eigrp topology 192.168.1.0** command.

```
R3# show ip eigrp topology 192.168.1.0
IP-EIGRP (AS 1): Topology entry for 192.168.1.0/24
   State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000Routing Descriptor
   Blocks:
   172.16.23.2 (Serial0/0/1), from 172.16.23.2, Send flag is 0x0Composite metric is
        (21152000/20640000), Route is InternalVector metric:
           Minimum bandwidth is 128 Kbit Total delay is
           45000 microsecondsReliability is 255/255
           Load is 1/255 Minimum
           MTU is 1500Hop count is
           2
   172.16.13.1 (Serial0/0/0), from 172.16.13.1, Send flag is 0x0Composite metric is
        (40640000/128256), Route is InternalVector metric:
           Minimum bandwidth is 64 Kbit Total delay is
           25000 microsecondsReliability is 255/255
           Load is 1/255 Minimum
           MTU is 1500Hop count is
           1
```

As indicated, R4 has two routes to reach 192.168.1.0. However, the metric for the route to R1 (172.16.13.1) is much higher (40640000) than the metric of the route to R2 (21152000), making the route through R2 the successor route.

Step 6: Configure PBR to provide path control.

Now you will deploy source-based IP routing by using PBR. You will change a default IP routing decision based on the EIGRP-acquired routing information for selected IP source-to-destination flows and apply a different next-hop router.

Recall that routers normally forward packets to destination addresses based on information in their routing table. By using PBR, you can implement policies that selectively cause packets to take different paths based on source address, protocol type, or application type. Therefore, PBR overrides the router's normal routing behavior.

Configuring PBR involves configuring a route map with **match** and **set** commands and then applying the route map to the interface.

The steps required to implement path control include the following:

- Choose the path control tool to use. Path control tools manipulate or bypass the IP routing table. For PBR, **route-map** commands are used.
- Implement the traffic-matching configuration, specifying which traffic will be manipulated. The **match** commands are used within route maps.
- Define the action for the matched traffic using **set** commands within route maps.

- Apply the route map to incoming traffic.

As a test, you will configure the following policy on router R3:

- All traffic sourced from R4 LAN A must take the R3 --> R2 --> R1 path.
- All traffic sourced from R4 LAN B must take the R3 --> R1 path.

a. On router R3, create a standard access list called **PBR-ACL** to identify the R4 LAN B network.

```
R3(config)# ip access-list standard PBR-ACL
R3(config-std-nacl)# remark ACL matches R4 LAN B trafficR3(config-std-
nacl)# permit 192.168.4.128 0.0.0.127 R3(config-std-nacl)# exit
```

b. Create a route map called **R3-to-R1** that matches PBR-ACL and sets the next-hop interface to the R1 serial 0/0/1 interface.

```
R3(config)# route-map R3-to-R1 permit
R3(config-route-map)# match ip address PBR-ACL R3(config-
route-map)# set ip next-hop 172.16.13.1R3(config-route-map)#
exit
```

c. Apply the R3-to-R1 route map to the serial interface on R3 that receives the traffic from R4. Use the **ip policy route-map** command on interface S0/1/0.

```
R3(config)# interface s0/1/0
R3(config-if)# ip policy route-map R3-to-R1
R3(config-if)# end
```

d. On R3, display the policy and matches using the **show route-map** command.

```
R3# show route-map
route-map R3-to-R1, permit, sequence 10Match clauses:
     ip address (access-lists): PBR-ACLSet
  clauses:
     ip next-hop 172.16.13.1
  Policy routing matches: 0 packets, 0 bytes
```

Note: There are currently no matches because no packets matching the ACL have passed through R3 S0/1/0.

Step 7: Test the policy.

Now you are ready to test the policy configured on R3. Enable the **debug ip policy** command on R3 so that you can observe the policy decision-making in action. To help filter the traffic, first create a standard ACL that identifies all traffic from the R4 LANs.

a. On R3, create a standard ACL which identifies all of the R4 LANs.

```
R3# conf t
Enter configuration commands, one per line. End with CNTL/Z.R3(config)# access-
list 1 permit 192.168.4.0 0.0.0.255 R3(config)# exit
```

b. Enable PBR debugging only for traffic that matches the R4 LANs.

```
R3# debug ip policy ?
  <1-199> Access list
  dynamic  dynamic PBR
  <cr>
```

```
R3# debug ip policy 1
Policy routing debugging is on for access list 1
```

c. Test the policy from R4 with the **traceroute** command, using R4 LAN A as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort. Tracing the route
to 192.168.1.1

  1 172.16.34.3 0 msec 0 msec 4 msec
  2 172.16.23.2 0 msec 0 msec 4 msec
  3 172.16.12.1 4 msec 0 msec *
```

Notice the path taken for the packet sourced from R4 LAN A is still going through R3 --> R2 --> R1.

As the traceroute was being executed, router R3 should be generating the following debug output.

```
R3#
*Feb 23 06:59:20.931: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, policy rejected -- normal forwarding
*Feb 23 06:59:29.935: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, policy rejected -- normal forwarding
*Feb 23 06:59:29.939: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, policy rejected -- normal forwarding
*Feb 23 06:59:29.939: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len28,
*Feb 23 06:59:38.943: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len28,
     FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:38.947: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len28,
     FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:38.947: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len28,
     FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:47.951: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len28,
     FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:47.955: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len28,
     FIB policy rejected(no match) - normal forwarding
     FIB policy rejected(no match) - normal forwarding
```

Why is the traceroute traffic not using the R3 --> R1 path as specified in the R3-to-R1 policy?

d. Test the policy from R4 with the **traceroute** command, using R4 LAN B as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort. Tracing the route
to 192.168.1.1

  1 172.16.34.3 12 msec 12 msec 16 msec
  2 172.16.13.1 28 msec 28 msec *
```

Now the path taken for the packet sourced from R4 LAN B is R3 --> R1, as expected.

The debug output on R3 also confirms that the traffic meets the criteria of the R3-to-R1 policy.

```
R3#
*Feb 23 07:07:46.467: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, policy match
*Feb 23 07:07:46.467: IP: route map R3-to-R1, item 10, permit
```

```
*Feb 23 07:07:46.467: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1 (Se
rial0/0/0), len 28, policy routed
*Feb 23 07:07:46.467: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
*Feb 23 07:07:55.471: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, policy match
*Feb 23 07:07:55.471: IP: route map R3-to-R1, item 10, permit
*Feb 23 07:07:55.471: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1 (Se
rial0/0/0), len 28, policy routed
*Feb 23 07:07:55.471: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
*Feb 23 07:07:55.471: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, policy match
*Feb 23 07:07:55.471: IP: route map R3-to-R1, item 10, permit
*Feb 23 07:07:55.475: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1 (Se
rial0/0/0), len 28, policy routed
*Feb 23 07:07:55.475: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
*Feb 23 07:07:55.475: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, FIB policy match
*Feb 23 07:07:55.475: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, g=
172.16.13.1, len 28, FIB policy routed
*Feb 23 07:08:04.483: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, FIB policy match
*Feb 23 07:08:04.483: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, g=
172.16.13.1, len 28, FIB policy routed
*Feb 23 07:08:04.491: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, FIB policy match
*Feb 23 07:08:04.491: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, g=
172.16.13.1, len 28, FIB policy routed
```

e. On R3, display the policy and matches using the **show route-map** command.

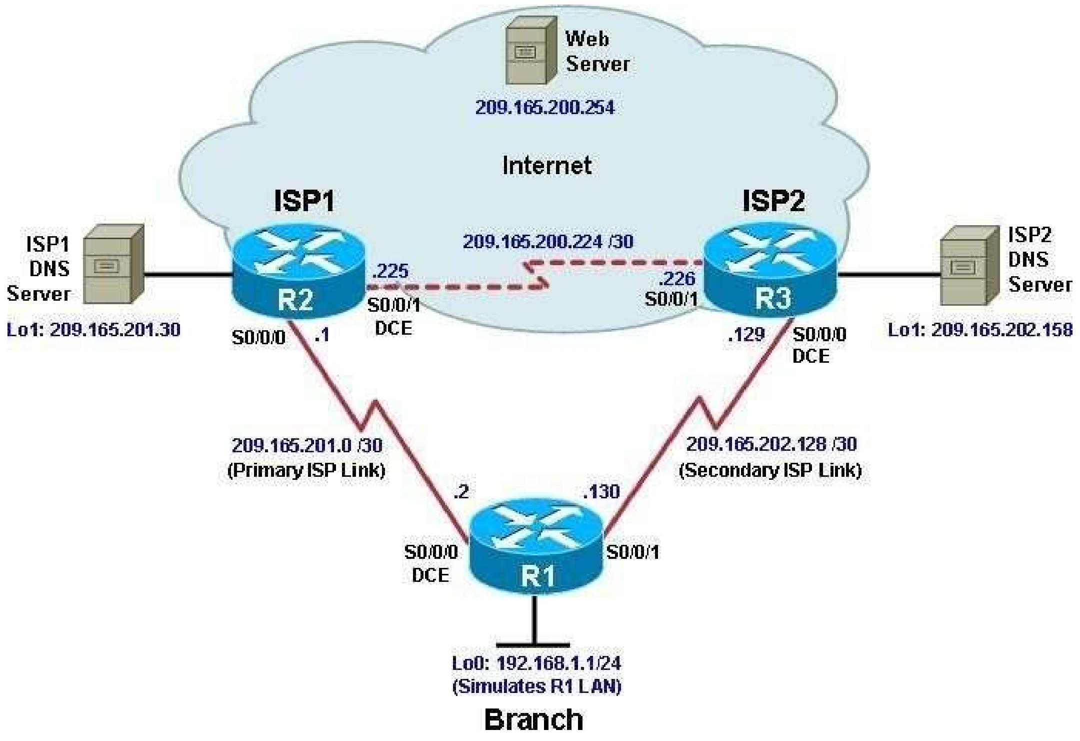
```
R3# show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-
  ACLSet clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 12 packets, 384 bytes
```

Note: There are now matches to the policy because packets matching the ACL have passed through R3S0/1/0.

Practical No.06

Aim: Configure IP SLA Tracking and Path Control with gateway

Topology



Objectives

- Configure and verify the IP SLA feature.
- Test the IP SLA tracking feature.
- Verify the configuration and operation using **show** and **debug** commands.

Background

You want to experiment with the Cisco IP Service Level Agreement (SLA) feature to study how it could be of value to your organization.

At times, a link to an ISP could be operational, yet users cannot connect to any other outside Internet resources. The problem might be with the ISP or downstream from them. Although policy-based routing (PBR) can be implemented to alter path control, you will implement the Cisco IOS SLA feature to monitor this behavior and intervene by injecting another default route to a backup ISP.

To test this, you have set up a three-router topology in a lab environment. Router R1 represents a branch office connected to two different ISPs. ISP1 is the preferred connection to the Internet, while ISP2 provides a backup link. ISP1 and ISP2 can also interconnect, and both can reach the web server. To monitor ISP1 for failure, you will configure IP SLA probes to track the reachability to the ISP1 DNS server. If connectivity to the ISP1 server fails, the SLA probes detect the failure and alter the default static route to point to the ISP2 server.

Note: This lab uses Cisco 1941 routers with Cisco IOS Release 15.2 with IP Base. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco IOS Release 15.2 or comparable)
- Serial and Ethernet cables

Step 1: Configure loopbacks and assign addresses.

p. Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear the previous configurations. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to them as well as the serial interfaces on R1, ISP1, and ISP2.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
hostname R1

interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0

interface Serial0/0/0
 description R1 --> ISP1
 ip address 209.165.201.2 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description R1 --> ISP2
 ip address 209.165.202.130 255.255.255.252
 bandwidth 128
 no shutdown
```

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255

interface Serial0/0/0
 description ISP1 --> R1
 ip address 209.165.201.1 255.255.255.252
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP1 --> ISP2
 ip address 209.165.200.225 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown
```

Router ISP2 (R3)

```
hostname ISP2

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0
 description ISP2 --> R1
 ip address 209.165.202.129 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP2 --> ISP1
 ip address 209.165.200.226 255.255.255.252
 bandwidth 128
 no shutdown
```

q. Verify the configuration by using the **show interfaces description** command. The output from router R1 is shown here as an example.

```
R1# show interfaces description | include up
Se0/0/0                        up             up        R1 --> ISP1
Se0/0/1                        up             up        R1 --> ISP2
Lo0                            up             up        R1 LAN
R1#
```

All three interfaces should be active. Troubleshoot if necessary.

Step 2: Configure static routing.

The current routing policy in the topology is as follows:

- Router R1 establishes connectivity to the Internet through ISP1 using a default static route.
- ISP1 and ISP2 have dynamic routing enabled between them, advertising their respective public address pools.
- ISP1 and ISP2 both have static routes back to the ISP LAN.

Note: For the purpose of this lab, the ISPs have a static route to an RFC 1918 private network address on the branch router R1. In an actual branch implementation, Network Address Translation (NAT) would be configured for all traffic exiting the branch LAN. Therefore, the static routes on the ISP routers would be pointing to the provided public pool of the branch office.

a. Implement the routing policies on the respective routers. You can copy and paste the following configurations.

Router R1

```
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1
R1(config)#
```

Router ISP1 (R2)

```
ISP1(config)# router eigrp 1
```

```
ISP1(config-router)# network 209.165.200.224 0.0.0.3
ISP1(config-router)# network 209.165.201.0 0.0.0.31
ISP1(config-router)# no auto-summary
ISP1(config-router)# exit
ISP1(config)#
ISP1(config-router)# ip route 192.168.1.0 255.255.255.0 209.165.201.2
ISP1(config)#
```

Router ISP2 (R3)

```
ISP2(config)# router eigrp 1
ISP2(config-router)# network 209.165.200.224 0.0.0.3
ISP2(config-router)# network 209.165.202.128 0.0.0.31
ISP2(config-router)# no auto-summary
ISP2(config-router)# exit
ISP2(config)#
ISP2(config)# ip route 192.168.1.0 255.255.255.0 209.165.202.130
ISP2(config)#
```

EIGRP neighbor relationship messages on ISP1 and ISP2 should be generated. Troubleshoot if necessary.

b. The Cisco IOS IP SLA feature enables an administrator to monitor network performance between Cisco devices (switches or routers) or from a Cisco device to a remote IP device. IP SLA probes continuously check the reachability of a specific destination, such as a provider edge router interface, the DNS server of the ISP, or any other specific destination, and can conditionally announce a default route only if the connectivity is verified.

Before implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2 DNS server to verify connectivity. You can copy the following Tcl script and paste it into R1.

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
ping $address source 192.168.1.1
}
```

All pings should be successful. Troubleshoot if necessary.

c. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server. You can copy the following Tcl script and paste it into R1.

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
trace $address source 192.168.1.1
}
```

Through which ISP is traffic flowing?

____

____

Step 3: Configure IP SLA probes.

When the reachability tests are successful, you can configure the Cisco IOS IP SLAs probes.

Different types of probes can be created, including FTP, HTTP, and jitter probes.

In this scenario, you will configure ICMP echo probes.

d. Create an ICMP echo probe on R1 to the primary DNS server on ISP1 using the **ip sla** command.

```
R1(config)# ip sla 11
R1(config-ip-sla)# icmp-echo 209.165.201.30
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)#

R1(config)# ip sla schedule 11 life forever start-time now
R1(config)#
```

The operation number of 11 is only locally significant to the router. The **frequency 10** command schedules the connectivity test to repeat every 10 seconds. The probe is scheduled to start now and to run forever.

e. Verify the IP SLAs configuration of operation 11 using the **show ip sla configuration 11** command.

```
R1# show ip sla configuration 11
IP SLAs Infrastructure Engine-III
Entry number: 11
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.30/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly
scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds):  Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None


R1#
```

The output lists the details of the configuration of operation 11. The operation is an ICMP echo to 209.165.201.30, with a frequency of 10 seconds, and it has already started (the start time has already passed).

f. Issue the **show ip sla statistics** command to display the number of successes, failures, and results of the latest operations.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11
   Latest RTT: 8 milliseconds
Latest operation start time: 10:33:18 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 51
Number of failures: 0
Operation time to live: Forever
R1#
```

You can see that operation 11 has already succeeded five times, has had no failures, and the last operation returned an OK result.

g. Although not actually required because IP SLA session 11 alone could provide the desired fault tolerance, create a second probe, 22, to test connectivity to the second DNS server located on router ISP2.

```
R1(config)# ip sla 22
R1(config-ip-sla)# icmp-echo 209.165.202.158
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)#
R1(config)# ip sla schedule 22 life forever start-time now
R1(config)# end
R1#
```

h. Verify the new probe using the **show ip sla configuration** and **show ip sla statistics** commands.

```
R1# show ip sla configuration 22
IP SLAs Infrastructure Engine-III
Entry number: 22
Owner:
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.202.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly
scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds):  Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
```

```
    Number of history Buckets kept: 15
    History Filter Type: None


R1#


R1# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10 (not considered if randomly
scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds):  Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is
configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:

R1#
R1# show ip sla statistics 22
IPSLAs Latest Operation Statistics

IPSLA operation id: 22
    Latest RTT: 16 milliseconds
Latest operation start time: 10:38:29 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 82
Number of failures: 0
Operation time to live: Forever


R1#
```

The output lists the details of the configuration of operation 22. The operation is an ICMP echo to 209.165.202.158, with a frequency of 10 seconds, and it has already started (the start time has already passed). The statistics also prove that operation 22 is active.

Step 4: Configure tracking options.

Although PBR could be used, you will configure a floating static route that appears or disappears depending on the success or failure of the IP SLA.

i. On R1, remove the current default route and replace it with a floating static route having an administrative distance of 5.

```
R1(config)# no ip route 0.0.0.0 0.0.0.0 209.165.201.1
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 5
R1(config)# exit
```

j. Verify the routing table.

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

S*    0.0.0.0/0 [5/0] via 209.165.201.1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30 is directly connected, Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
R1#
```

Notice that the default static route is now using the route with the administrative distance of 5. The first tracking object is tied to IP SLA object 11.

k. From global configuration mode on R1, use the **track 1 ip sla 11 reachability** command to enter the config-track subconfiguration mode.

```
R1(config)# track 1 ip sla 11 reachability
R1(config-track)#
```

l. Specify the level of sensitivity to changes of tracked objects to 10 seconds of down delay and 1 second of up delay using the **delay down 10 up 1** command. The delay helps to alleviate the effect of flapping objects—objects that are going down and up rapidly. In this situation, if the DNS server fails momentarily and comes back up within 10 seconds, there is no impact.

```
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)#
```

m. To view routing table changes as they happen, first enable the **debug ip routing** command.

```
R1# debug ip routing
IP routing debugging is on
R1#
```

n. Configure the floating static route that will be implemented when tracking object 1 is active. Use the **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1** command to create a floating static default route via 209.165.201.1 (ISP1). Notice that this command references the tracking object number 1, which in turn references IP SLA operation number 11.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
R1(config)#
Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1   0 1048578
```

```
Jan 10 10:45:39.119: RT: closer admin distance for 0.0.0.0, flushing 1
routes
Jan 10 10:45:39.119: RT: add 0.0.0.0/0 via 209.165.201.1, static metric
[2/0]
Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1   0 1048578

Jan 10 10:45:39.119: RT: rib update return code: 17
Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1   0 1048578

Jan 10 10:45:39.119: RT: rib update return code: 17
R1(config)#
```

Notice that the default route with an administrative distance of 5 has been immediately flushed because of a route with a better admin distance. It then adds the new default route with the admin distance of 2.

o. Repeat the steps for operation 22, track number 2, and assign the static route an admin distance higher than track 1 and lower than 5. On R1, copy the following configuration, which sets an admin distance of 3.

```
R1(config)# track 2 ip sla 22 reachability
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)#
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
R1(config)#
```

p. Verify the routing table again.

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

S*    0.0.0.0/0 [2/0] via 209.165.201.1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30  is  directly  connected,  Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
R1#
```

Although a new default route was entered, its administrative distance is not better than 2. Therefore, it does not replace the previously entered default route.

Step 5: Verify IP SLA operation.

In this step you observe and verify the dynamic operations and routing changes when tracked objects fail. The following summarizes the process:

- Disable the DNS loopback interface on ISP1 (R2).
- Observe the output of the **debug** command on R1.
- Verify the static route entries in the routing table and the IP SLA statistics of R1.
- Re-enable the loopback interface on ISP1 (R2) and again observe the operation of the IP SLA tracking feature.

q. On ISP1, disable the loopback interface 1.

```
ISP1(config-if)# int lo1
ISP1(config-if)# shutdown
ISP1(config-if)#
Jan 10 10:53:25.091: %LINK-5-CHANGED: Interface Loopback1, changed
state to administratively down
Jan 10 10:53:26.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback1, changed state to down
ISP1(config-if)#
```

r. On R1, observe the **debug** output being generated. Recall that R1 will wait up to 10 seconds before initiating action therefore several seconds will elapse before the output is generated.

```
R1#
Jan 10 10:53:59.551: %TRACK-6-STATE: 1 ip sla 11 reachability Up ->
Down
Jan 10 10:53:59.551: RT: del 0.0.0.0 via 209.165.201.1, static metric
[2/0]
Jan 10 10:53:59.551: RT: delete network route to 0.0.0.0/0
Jan 10 10:53:59.551: RT: default path has been cleared
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.202.129   0 1048578

Jan 10 10:53:59.551: RT: add 0.0.0.0/0 via 209.165.202.129, static
metric [3/0]
Jan 10 10:53:59.551: RT: default path is now 0.0.0.0 via
209.165.202.129
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1   0 1048578

Jan 10 10:53:59.551: RT: rib update return code: 17
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.202.129   0 1048578

Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1   0 1048578

Jan 10 10:53:59.551: RT: rib update return code: 17
R1#
```

The tracking state of track 1 changes from up to down. This is the object that tracked reachability for IP SLA object 11, with an ICMP echo to the ISP1 DNS server at 209.165.201.30.

R1 then proceeds to delete the default route with the administrative distance of 2 and installs the next highest default route to ISP2 with the administrative distance of 3.

s. On R1, verify the routing table.

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.202.129 to network 0.0.0.0

S*    0.0.0.0/0 [3/0] via 209.165.202.129
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
```

```
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30 is directly connected, Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
R1#
```

The new static route has an administrative distance of 3 and is being forwarded to ISP2 as it should.

t. Verify the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11
   Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 11:01:08 UTC Sat Jan 10 2015
Latest operation return code: Timeout
Number of successes:  173
Number of failures: 45
Operation time to live: Forever
IPSLA operation id: 22
   Latest RTT: 8 milliseconds
Latest operation start time: 11:01:09 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes: 218
Number of failures: 0
Operation time to live: Forever


R1#
```

Notice that the latest return code is **Timeout** and there have been 45 failures on IP SLA object 11.

u. On R1, initiate a trace to the web server from the internal LAN IP address.

```
R1# trace 209.165.200.254 source 192.168.1.1
Type escape sequence to abort.
Tracing the route to 209.165.200.254
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.202.129 4 msec *  *
R1#
```

This confirms that traffic is leaving router R1 and being forwarded to the ISP2 router.

v. On ISP1, re-enable the DNS address by issuing the **no shutdown** command on the loopback 1 interface to examine the routing behavior when connectivity to the ISP1 DNS is restored.

```
ISP1(config-if)# no shutdown
Jan 10 11:05:45.847: %LINK-3-UPDOWN: Interface Loopback1, changed state
to up
Jan 10 11:05:46.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback1, changed state to up
ISP1(config-if)#
```

Notice the output of the **debug ip routing** command on R1.

```
R1#
Jan 10 11:06:20.551: %TRACK-6-STATE: 1 ip sla 11 reachability Down ->
Up
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1    0 1048578

Jan 10 11:06:20.551: RT: closer admin distance for 0.0.0.0, flushing 1
routes
Jan 10 11:06:20.551: RT: add 0.0.0.0/0 via 209.165.201.1, static metric
[2/0]
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.202.129    0 1048578

Jan 10 11:06:20.551: RT: rib update return code: 17
Jan 10 11:06:20.551: RT: u
R1#pdating static 0.0.0.0/0 (0x0) :
    via 209.165.202.129    0 1048578

Jan 10 11:06:20.551: RT: rib update return code: 17
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0) :
    via 209.165.201.1    0 1048578

Jan 10 11:06:20.551: RT: rib update return code: 17
R1#
```

Now the IP SLA 11 operation transitions back to an up state and reestablishes the default static route to ISP1 with an administrative distance of 2.

w. Again examine the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11
   Latest RTT: 8 milliseconds
Latest operation start time: 11:07:38 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes:  182
Number of failures: 75
Operation time to live: Forever



IPSLA operation id: 22
   Latest RTT: 16 milliseconds
Latest operation start time: 11:07:39 UTC Sat Jan 10 2015
Latest operation return code: OK
Number of successes:  257
Number of failures:  0
Operation time to live: Forever


R1#
```

The IP SLA 11 operation is active again, as indicated by the OK return code, and the number of successes is incrementing.

x. Verify the routing table.

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

S*    0.0.0.0/0 [2/0] via 209.165.201.1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
```

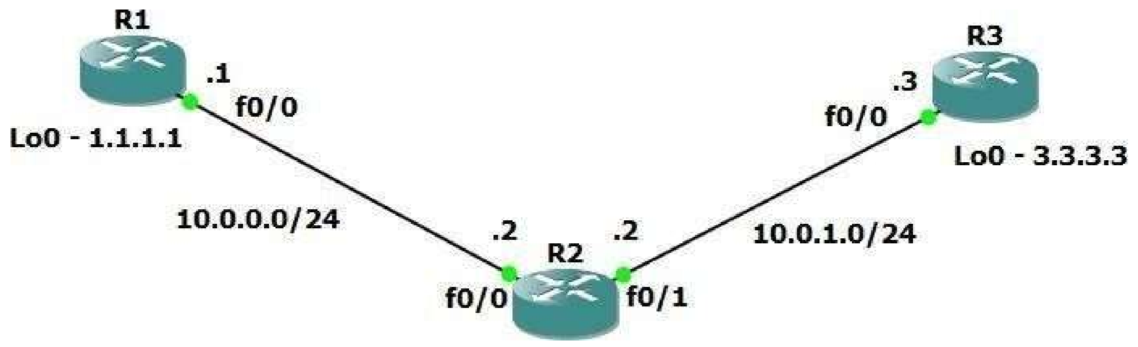
```
C        192.168.1.0/24 is directly connected, Loopback0
L        192.168.1.1/32 is directly connected, Loopback0
      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.201.0/30 is directly connected, Serial0/0/0
L        209.165.201.2/32 is directly connected, Serial0/0/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.202.128/30 is directly connected, Serial0/0/1
L        209.165.202.130/32 is directly connected, Serial0/0/1
R1#
```

Practical No. 07

Aim: Configuring Basic MPLS Using OSPF

Step 1 – IP addressing of MPLS Core and OSPF

First bring 3 routers into your topology R1, R2, R3 position them as below. We are going to address the routers and configure ospf to ensure loopbackto loopback connectivity between R1 and R3



```
R1

hostname

R1int lo0

ip add 1.1.1.1 255.255.255.255

ip ospf 1 area 0



int f0/0

ip add 10.0.0.1 255.255.255.0

no shut

ip ospf 1 area 0



R2

hostname

r2

int lo0
```

```
ip add 2.2.2.2 255.255.255.255

ip ospf 1 are 0


int f0/0

ip add 10.0.0.2 255.255.255.0

no shut

ip ospf 1 area 0


int f0/1

ip add 10.0.1.2 255.255.255.0

no shut

ip ospf 1 area 0




R3

hostname R3int
lo0
ip add 3.3.3.3 255.255.255.255

ip ospf 1 are 0


int f0/0

ip add 10.0.1.3 255.255.255.0

no shut

ip ospf 1 area 0
```

You should now have full ip connectivity between R1, R2, R3 to verify thiswe need to see if we can ping between the loopbacks of R1 and R3

```
R1#ping 3.3.3.3 source lo0



Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2seconds:

Packet sent with a source address of 1.1.1.1

!!!!!

Success rate is 100 percent (5/5), round-tripmin/avg/max =
40/52/64 ms

R1#
```

You could show the routing table here, but the fact that you can ping
between the loopbacks is verification enough and it is safe to move on.

Step 2 – Configure LDP on all the interfaces in the MPLS Core

In order to run MPLS you need to enable it, there are two ways to do this.

- At each interface enter the **mpls ip** command
- Under the ospf process use the **mpls ldp autoconfig** command

For this tutorial we will be using the second option, so go int the ospf process and enter mpls ldp autoconfig – this will enable mpls label distribution protocol on every interface running ospf under that specificprocess.

```
R1

router ospf 1

mpls ldp autoconfig



R2

router ospf 1
```

```
mpls ldp autoconfig



R3

router ospf 1

mpls ldp autoconfig
```

You should see log messages coming up showing the LDP neighbors are up.

```
R2#

*Mar 1 00:31:53.643: %SYS-5-CONFIG_I: Configured fromconsole

*Mar   1 00:31:54.423: %LDP-5-NBRCHG: LDP Neighbor
1.1.1.1:0 (1) is UPR2#

*Mar   1 00:36:09.951: %LDP-5-NBRCHG: LDP Neighbor
3.3.3.3:0 (2) is UP
```

To verify the mpls interfaces the command is very simple – **sh mpls interface**

This is done on R2 and you can see that both interfaces are running mplsand using LDP

```
R2#sh mpls interface

Interface              IP            Tunnel
Operationa
l
FastEthernet0/0        Yes (ldp)     No        Yes

FastEthernet0/1        Yes (ldp)     No        Yes
```

You can also verify the LDP neighbors with the **sh mpls ldp neighbors** command.

```
R2#sh mpls ldp neigh

    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0TCP connection:

        1.1.1.1.646 - 2.2.2.2.37909
        State: Oper; Msgs sent/rcvd: 16/17;

        DownstreamUp time: 00:07:46

        LDP discovery sources:

          FastEthernet0/0, Src IP addr:

        10.0.0.1Addresses bound to peer LDP

          10.0.0.         1.1.1.

    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident

        2.2.2.2:0TCP connection: 3.3.3.3.22155 -

        2.2.2.2.646

        State: Oper; Msgs sent/rcvd: 12/11;

        DownstreamUp time: 00:03:30

        LDP discovery sources:
          FastEthernet0/1, Src IP addr:

          10.0.1.         3.3.3.
```

One more verification to confirm LDP is running ok is to do a trace between R1 and R3 and verify if you get MPLS Labels show up in the trace.

```
R1#trace 3.3.3.3



Type escape sequence to abort.Tracing the

route to 3.3.3.3



   1 10.0.0.2 [MPLS: Label 17 Exp 0] 84 msec 72 msec 44msec

   2 10.0.1.3 68 msec 60 msec *
```

As you can see the trace to R2 used an MPLS Label in the path, as this is a very small MPLS core only one label was used as R3 was the final hop.

So to review we have now configured IP addresses on the MPLS core,enabled OSPF and full IP connectivity between all routers and finally enabled mpls on all the interfaces in the core and have established ldpneighbors between all routers.

The next step is to configure MP-BGP between R1 and R3

This is when you start to see the layer 3 vpn configuration come to life

Step 3 – MPLS BGP Configuration between R1 and R3

We need to establish a Multi Protocol BGP session between R1 and R3 thisis done by configuring the vpnv4 address family as below

```
R1#

router bgp 1

 neighbor 3.3.3.3 remote-as 1

 neighbor 3.3.3.3 update-source Loopback0no auto-summary
 !

 address-family vpnv4 neighbor
  3.3.3.3 activate


R3#

router bgp 1

 neighbor 1.1.1.1 remote-as 1

 neighbor 1.1.1.1 update-source Loopback0no auto-summary
 !

 address-family vpnv4 neighbor
  1.1.1.1 activate
```

```
*Mar  1 00:45:01.047: %BGP-5-ADJCHANGE: neighbor 1.1.1.1
Up
```

You should see log messages showing the BGP sessions coming up.

To verify the BGP session between R1 and R3 issue the command **sh bgp vpnv4 unicast all summary**

```
R1#sh bgp vpnv4 unicast all summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 1, main routing table



Neighbor        V    AS MsgRcvd          TblVer
OutQ           State/PfxRc
Up/Down
                4     1     21      21          1    0
3.3.3.3           0
```

You can see here that we do have a bgp vpnv4 peering to R3 – looking at
the PfxRcd you can see it says 0 this is because we have not got any routesin BGP. We are now going to add two more routers to the topology. These will be the customer sites connected to R1 and R3. We will then create a VRF on each router and put the interfaces connected to each site router intothat VRF.

Step 4 – Add two more routers, create VRFs

We will add two more routers into the topology so it now looks like the finaltopology

Router 4 will peer OSPF using process number 2 to a VRF configured onR1. It will use the local site addressing of 192.168.1.0/24.

```
R4

int lo0

ip add 4.4.4.4 255.255.255.255


ip ospf 2 area 2int


f0/0
```

```
ip add 192.168.1.4 255.255.255.0

ip ospf 2 area 2no

shut



R1

int f0/1

no shut

ip add 192.168.1.1 255.255.255.0
```

Now at this point we have R4 peering to R1 but in the global routing table of

R1 which is not what we want.

We are now going to start using VRF's

What is a VRF in networking?

Virtual routing and forwarding (**VRF**) is a technology included in IP (InternetProtocol) that allows multiple instances of a routing table to co-exist in a router and work together but not interfere with each other.. This increases functionality by allowing network paths to be segmented without using multiple devices.

As an example if R1 was a PE Provider Edge router of an ISP and it had twocustomers that were both addressed locally with the 192.168.1.0/24 addressspace it could accommodate both their routing tables in different VRFs – it distinguishes between the two of them using a Route Distinguisher

So back to the topology – we now need to create a VRF on R1For this mpls tutorial I will be

using VRF RED

```
R1

ip vrf RED

rd 4:4

route-target both 4:4
```

The RD and route-target do not need to be the same – and for a fullexplanation please read this post on Route Distinguishers
Route Distinguisher vs Route Target before proceeding.

So now we have configured the VRF on R1 we need to move the interfaceF0/1 into that VRF

```
R1

int f0/1

ip vrf forwarding RED
```

Now notice what happens when you do that – the IP address is removed

```
R1(config-if)#ip vrf fo

R1(config-if)#ip vrf forwarding RED

% Interface FastEthernet0/1 IP address 192.168.1.1removed due to
enabling VRF RED
```

You just need to re-apply it

```
R1

int f0/1

ip address 192.168.1.1 255.255.255.0
```

Now if we view the config on R1 int f0/1 you can see the VRF configured.

```
R1




R1#sh run int f0/1 Building

configuration...


Current configuration : 119 bytes

!
```

```
interface FastEthernet0/1ip vrf

 forwarding RED

 ip address 192.168.1.1 255.255.255.0

 duplex auto

 speed auto

end



R1#
```

Now we can start to look int VRF’s and how they operate – you need to understand now that there are 2 routing tables within R1

- The Global Routing Table
- The Routing Table for VRF RED

If you issue the command **sh ip route** this shows the routes in the globaltable and you will notice that you do not see 192.168.1.0/24

```
R1#sh ip route

Codes: 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 externaltype 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



Gateway of last resort is not set
```

```
1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback02.0.0.0/32 is

  subnetted, 1 subnets

O 2.2.2.2 [110/11] via 10.0.0.2, 01:03:48,
FastEthernet0/0

  3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/21] via 10.0.0.2, 01:02:29,
FastEthernet0/0

  10.0.0.0/24 is subnetted, 2 subnets

C 10.0.0.0 is directly connected, FastEthernet0/0O 10.0.1.0 [110/20]


via 10.0.0.2, 01:02:39,
FastEthernet0/0R1#
```

If you now issue the command sh ip route vrf red – this will show the routes in the routing table for VRF RED

```
R1#sh ip route vrf red

% IP routing table red does not exist
```

NOTE: The VRF name is case sensitive!

```
R1#sh ip route vrf RED



Routing Table: RED

Codes: 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 externaltype 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



Gateway of last resort is not set



C 192.168.1.0/24 is directly connected, FastEthernet0/1R1#
```

We just need to enable OSPF on this interface and get the loopback address for R4 in the VRF RED routing table before proceeding.

```
R1

  int f0/1

  ip ospf 2 area 2
```

You should see a log message showing the OSPF neighbor come up

```
R1(config-if)#

             *Mar 1 01:12:54.323: %OSPF-5-ADJCHG: Process 2, Nbr
4.4.4.4

            on FastEthernet0/1 from LOADING to FULL, Loading Done
```

If we now check the routes in the VRF RED routing table you should see 4.4.4.4 in there as well.

```
R1#sh ip route vrf RED



Routing Table: RED

  Codes: 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 externaltype 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



Gateway of last resort is not set



4.0.0.0/32 is subnetted, 1 subnets

  O 4.4.4.4 [110/11] via 192.168.1.4, 00:00:22,
FastEthernet0/1

  C 192.168.1.0/24 is directly connected, FastEthernet0/1R1#
```

We now need to repeat this process for R3 & R6

Router 6 will peer OSPF using process number 2 to a VRF configured onR3. It will use the local site addressing of 192.168.2.0/24.

```
 R6

 int lo0

 ip add 6.6.6.6 255.255.255.255

 ip ospf 2 area 2int

 f0/0

 ip add 192.168.2.6 255.255.255.0

 ip ospf 2 area 2



no shut



R3

int f0/1

no shut

ip add 192.168.2.3 255.255.255.0
```

We also need to configure a VRF onto R3 as well.

```
R3

ip vrf RED

rd 4:4

route-target both 4:4
```

So now we have configured the VRF on R3 we need to move the interface F0/1 into that VRF

```
R3

int f0/1

ip vrf forwarding RED
```

Now notice what happens when you do that – the IP address is removed

```
R3(config-if)#ip vrf forwarding RED

% Interface FastEthernet0/1 IP address 192.168.2.1removed due to
enabling VRF RED
```

You just need to re-apply it

```
R3
int f0/1
ip address 192.168.2.1 255.255.255.0
```

Now if we view the config on R3 int f0/1 you can see the VRF configured.

```
R3
```

```
R3#sh run int f0/1 Building

configuration...



Current configuration : 119 bytes

!

interface FastEthernet0/1ip vrf

forwarding RED

ip address 192.168.2.1 255.255.255.0

duplex auto

speed auto

end
```

Finally we just need to enable OSPF on that interface and verify the routes are in the RED routing table.

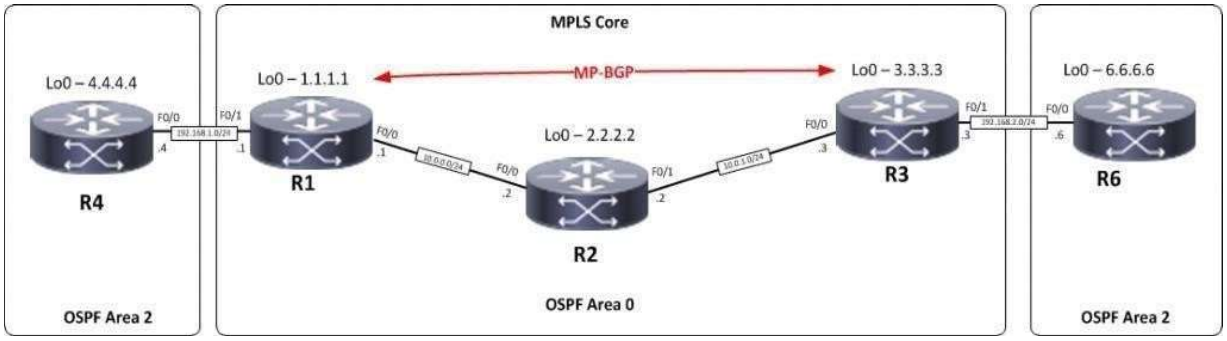
```
R3

int f0/1

ip ospf 2 area 2
```

Check the routes in vrf RED

```
R3

R3#sh ip route vrf RED



Routing Table: RED

Codes: C - connected, S - static, R - RIP, M - mobile, B
- BGP



Gateway of last resort is not set
```

```
     6.0.0.0/32 is subnetted, 1 subnets

O       6.6.6.6 [110/11] via 192.168.2.6,
FastEthernet0/

C    192.168.2.0/24 is directly
FastEthernet0/
1
```

Ok so we have come a long way now let's review the current situation. We now have this setup



R1,R2,R3 form the MPLS Core and are running OSPF with all loopbacks running a /32 address and all have full connectivity. R1 and R3 are peeringwith MP-BGP. LDP is enabled on all the internal interfaces. The external interfaces of the MPLS core have been placed into a VRF called RED and then a site router has been joined to that VRF on each side of the MPLS core – (These represent a small office)

The final step to get full connectivity across the MPLS core is to redistributethe routes in OSPF on R1 and R3 into MP-BGP and MP-BGP into OSPF, this is what we are going to do now.

We need to redistribute the OSPF routes from R4 into BGP in the VRF on R1, the OSPF routes from R6 into MP-BGP in the VRF on R3 and then theroutes in MP-BGP in R1 and R3 back out to OSPF

Before we start lets do some verifications

Check the routes on R4

```
R4#sh ip route

4.0.0.0/32 is subnetted, 1 subnets

C 4.4.4.4 is directly connected, Loopback0
```

```
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

As expected we have the local interface and the loopback address.

When we are done we want to see 6.6.6.6 in there so we can ping acrossthe MPLS

Check the routes on R1

```
R1#sh ip route



1.0.0.0/32 is subnetted, 1 subnets

C 1.1.1.1 is directly connected, Loopback02.0.0.0/32 is

  subnetted, 1 subnets

O 2.2.2.2 [110/11] via 10.0.0.2, 00:01:04,
FastEthernet0/0

  3.0.0.0/32 is subnetted, 1 subnets

O 3.3.3.3 [110/21] via 10.0.0.2, 00:00:54,
FastEthernet0/0

  10.0.0.0/24 is subnetted, 2 subnets

C 10.0.0.0 is directly connected, FastEthernet0/0O 10.0.1.0 [110/20]

via 10.0.0.2, 00:00:54,
FastEthernet0/0
```

Remember we have a VRF configured on this router so this command will
show routes in the global routing table (the MPLS Core) and it will not showthe 192.168.1.0/24 route as that is in VRF RED – to see that we run the following command

```
R1#sh ip route vrf RED



Routing Table: RED



4.0.0.0/32 is subnetted, 1 subnets
```

```
O 4.4.4.4 [110/11] via 192.168.1.4, 00:02:32,
FastEthernet0/1

C 192.168.1.0/24 is directly connected, FastEthernet0/1
```

Here you can see Routing Table: RED is shown and the routes to R4 are now visible with 4.4.4.4 being in OSPF.So we need to do

the following;

- Redistribute OSPF into MP-BGP on R1
- Redistribute MP-BGP into OSPF on R1
- Redistribute OSPF into MP-BGP on R3
- Redistribute MP-BGP into OSPF on R3

Redistribute OSPF into MP-BGP on R1

```
R1

router bgp 1

address-family ipv4 vrf RED

redistribute ospf 2
```

Redistribute OSPF into MP-BGP on R3

```
R3

router bgp 1

address-family ipv4 vrf RED

redistribute ospf 2
```

This has enabled redistribution of the OSPF routes into BGP. We can check the routes from R4 and R6 are now showing in the BGP table for their VRFwith this command

sh ip bgp vpnv4 vrf RED

```
R1#sh ip bgp vpnv4 vrf RED

BGP table version is 9, local router ID is 1.1.1.1

Status codes: s suppressed, d damped, h history, *valid, > best,
 r RIB-failure, S Stale

  Origin codes: i IGP, e - EGP, ? - incomplete
                -

 Network Next Hop Metric LocPrf Weight Path


Route Distinguisher: 4:4 (default for vrf RED)
*> 4.4.4.4/32 192.168.1.4 11 32768 ?

*>i6.6.6.6/32 3.3.3.3    100 0 ?
11
*> 192.168.1.0 0.0.0.0   32768 ?
0
*>i192.168.2.0 3.3.3.3   100 0 ?
0
```

Here we can see that 4.4.4.4 is now in the BGP table in VRF RED on
R1 with a next hop of 192.168.1.4 (R4) and also 6.6.6.6 is in there as wellwith a next hop of 3.3.3.3 (which is the loopback of R3 – showing that it isgoing over the MPLS and R1 is not in the picture)

The same should be true on R3

```
R3#sh ip bgp vpnv4 vrf RED

BGP table version is 9, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, *valid, > best, i -
internal,

r RIB-failure, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete



Network Next Hop Metric LocPrf Weight Path Route Distinguisher: 4:4

(default for vrf RED)

      *>i4.4.4.4/32 1.1.1.1 11 100 0 ?

*> 6.6.6.6/32 192.168.2.6 11 32768 ?

*>i192.168.1.0 1.1.1.1 0 100 0 ?
```

```
*> 192.168.2.0 0.0.0.0 0 32768 ?
```

Which it is! 6.6.6.6 is now in the BGP table in VRF RED on R3 with a next
hop of 192.168.2.6 (R6) and also 4.4.4 is in there as well with a next hop of
1.1.1.1 (which is the loopback of R1 – showing that it is going over the MPLSand R2 is not in the picture)
The final step is to get the routes that have come across the MPLS back intoOSPF and then we can get end to end connectivity

```
R1



router ospf   2 redistribute bgp 1

subnets



R3



router ospf   2 redistribute bgp 1

subnets
```

If all has worked we should be now able to ping 6.6.6.6 from R4

Before we do let's see what the routing table looks like on R4

```
R4#sh ip route





4.0.0.0/32 is subnetted, 1 subnets

C 4.4.4.4 is directly connected, Loopback06.0.0.0/32 is

  subnetted, 1 subnets

O IA 6.6.6.6 [110/21] via 192.168.1.1, 00:01:31,
FastEthernet0/0

C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

```
O E2 192.168.2.0/24 [110/1] via 192.168.1.1, 00:01:31,
FastEthernet0/0
```

Great we have 6.6.6.6 in there

Also check the routing table on R6

```
R6#sh ip route



4.0.0.0/32 is subnetted, 1 subnets

O IA 4.4.4.4 [110/21] via 192.168.2.1, 00:01:22,
FastEthernet0/0

6.0.0.0/32 is subnetted, 1 subnets

C 6.6.6.6 is directly connected, Loopback0O IA

192.168.1.0/24 [110/11] via
192.168.2.1,00:01:22,FastEthernet0/0

C 192.168.2.0/24 is directly connected, FastEthernet0/0
```

Brilliant we have 4.4.4.4 in there so we should be able to ping across the MPLS

```
R4#ping 6.6.6.6



Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2seconds:

!!!!!

Success rate is 100 percent (5/5), round-tripmin/avg/max=
40/48/52ms
```

Which we can – to prove this is going over the MPLS and be label switched and not routed, lets do a trace

```
R4#trace 6.6.6.6
```

```
Type escape sequence to abort.Tracing the

route to 6.6.6.6



1 192.168.1.1 20 msec 8 msec 8 msec

  2 10.0.0.2 [MPLS: Labels 17/20 Exp 0] 36 msec 40 msec
36 msec

  3 192.168.2.1 [MPLS: Label 20 Exp 0] 16 msec 40 msec 16msec

  4 192.168.2.6 44 msec 40 msec 56 msecR4#
```