



# **SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE**

NAAC ACCREDITED 'A' GRADE COLLEGE

(ISO 9001:2015 CERTIFIED INSTITUTION)

NERUL, NAVI MUMBAI - 400706

## *Certificate*

**Seat No: 2630267**

**Certified that Vishal Varma**

**Of Class MSc. IT Part 1 has duly completed the practical**

**course in the subject of Modern Networking**

**during the academic year 2021-22 as per the syllabus**

**prescribed by the University of Mumbai.**

**Subject Teacher**

**External Examiner**

**Head of Department**

**Principal**

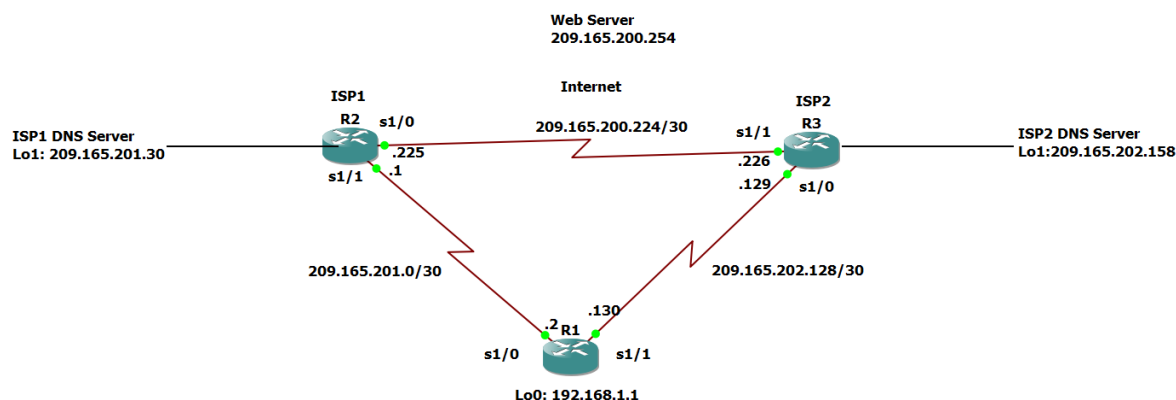
## INDEX

<b>Sr.No</b>	<b>Practical</b>	<b>Page No</b>
<b>1.</b>	<b>Configure IP SLA Tracking and Path Control</b>	
<b>2.</b>	<b>Using the AS_PATH Attribute</b>	
<b>3.</b>	<b>Configuring IBGP and EBGP Sessions, Local Preference, and MED</b>	
<b>4.</b>	<b>Secure the Management Plane</b>	
<b>5.</b>	<b>Configure and Verify Path Control</b>	
<b>6.</b>	<b>Configure IP SLA Tracking and Path Control</b>	
<b>7.</b>	<b>Inter-VLAN Routing</b>	
<b>8.</b>	<b>Simulating MPLS environment</b>	

## Practical 1:

### Configure IP SLA Tracking and Path Control

#### TOPOLOGY



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

- a. Cable the network as shown in the topology diagram.

####Router R1

hostname R1

```
interface Loopback 0
description R1 LAN
ip address 192.168.1.1 255.255.255.0
exit
```

```
interface Serial1/0
description R1 --> ISP1
ip address 209.165.201.2 255.255.255.252
clock rate 128000
bandwidth 128
no shutdown
exit
```

```
interface Serial1/1
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
exit
```

```
interface Loopback1
description ISP1 DNS Server
ip address 209.165.201.30 255.255.255.255
exit
```

```
interface Serial1/1
description ISP1 --> R1
ip address 209.165.201.1 255.255.255.252
bandwidth 128
no shutdown
exit
```

```
interface Serial1/0
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
exit
```

```
interface Loopback1
description ISP2 DNS Server
ip address 209.165.202.158 255.255.255.255
exit
```

```
interface Serial1/0
description ISP2 --> R1
ip address 209.165.202.129 255.255.255.252
clock rate 128000
bandwidth 128
no shutdown
exit
```

```
interface Serial1/1
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.

R1# show interfaces description

```
R1#sh int des | include up
Se1/0                up          up          R1 --> ISP1
Se1/1                up          up          R1 --> ISP2
Lo0                  up          up          R1 LAN
R1#
```

- 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.

Implement the routing policies on the respective routers.

```
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
R1(config)#
```

```
*Apr 28 20:47:05.587: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.226 (Serial1/0) is up: new adjacency
ISP1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
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)#ip route 192.168.1.0 255.255.255.0 209.165.201.2
ISP1(config)#
*Apr 28 20:47:05.587: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.226 (Serial1/0) is up: new adjacency
ISP1(config)#
```

```
ISP2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
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)#
*Apr 28 20:47:05.351: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.225 (Serial1/1) is up: new adjacency
ISP2(config-router)#exit
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.

```
*Apr 28 20:47:05.587: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.226 (Serial1/0) is up: new adjacency
```

## Step 2: Verify server reachability

- 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.

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

```
R1#tclsh  
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 = 20/26/32 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 = 20/30/44 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 = 32/39/48 ms  
R1(tcl)#
```

- b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server.

```
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
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.201.1 32 msec 28 msec 32 msec
Type escape sequence to abort.
Tracing the route to 209.165.201.30
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.201.1 32 msec 36 msec 32 msec
Type escape sequence to abort.
Tracing the route to 209.165.202.158
VRF info: (vrf in name/id, vrf out name/id)
  1 209.165.201.1 40 msec 40 msec 36 msec
  2 209.165.200.226 52 msec 56 msec 52 msec
R1(tcl)#
```

### Step 3: Configure IP SLA probes.

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

- b. 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
```

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

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

IPSLA operation id: 11
    Latest RTT: 12 milliseconds
Latest operation start time: 21:06:31 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 22
Number of failures: 0
Operation time to live: Forever
```

- 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.

```
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)#ip sla schedule 22 life forever start-time now
R1(config)#
```

- 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-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:
```



```

R1#show ip sla statistics 22
IPSLAs Latest Operation Statistics

IPSLA operation id: 22
    Latest RTT: 68 milliseconds
Latest operation start time: 21:11:23 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 21
Number of failures: 0
Operation time to live: Forever

```

#### Step 4: Configure tracking options.

- 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
R1#

```

- b. Verify the routing table.

```

R1#sh 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 a
rea
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external typ
e 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user
static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l
- LISP
        + - replicated route, % - next hop override

Gateway of last resort is 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, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1

```

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

```
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.

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

```
R1#debug ip routing
IP routing debugging is on
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 28 21:20:36.103: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:20:36.107: RT: closer admin distance for 0.0.0.0, flushing 1 routes
*Apr 28 21:20:36.107: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]
*Apr 28 21:20:36.111: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:20:36.115: RT: rib update return code: 17
*Apr 28 21:20:36.119: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:20:36.123: RT: rib update return code: 17
R1(config)#
```

- 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.

```
R1(config)#track 2 ip sla 22 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
R1(config)#
*Apr 28 21:31:11.075: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:31:11.079: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:31:11.083: RT: rib update return code: 17
*Apr 28 21:31:11.091: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.202.129    1048578

*Apr 28 21:31:11.095: RT: rib update return code: 17
```

- g. Verify the routing table again.

```

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

Gateway of last resort is 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, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1

```

### Step 5: Verify IP SLA operation.

```

ISP1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP1(config)#int lo1
ISP1(config-if)#sh
ISP1(config-if)#
*Apr 28 21:35:31.435: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to down
*Apr 28 21:35:31.439: %LINK-5-CHANGED: Interface Loopback1, changed state to adm
inistratively down
ISP1(config-if)#

```

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

```

R1#
*Apr 28 21:35:43.791: %TRACKING-5-STATE: 1 ip sla 11 reachability
Up->Down
*Apr 28 21:35:43.791: RT: del 0.0.0.0 via 209.165.201.1, static me
tric [2/0]
*Apr 28 21:35:43.791: RT: delete network route to 0.0.0.0/0
*Apr 28 21:35:43.791: RT: default path has been cleared
*Apr 28 21:35:43.791: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.202.129    1048578

*Apr 28 21:35:43.795: RT: add 0.0.0.0/0 via 209.165.202.129, stati
c metric [3/0]
*Apr 28 21:35:43.799: RT: default path is now 0.0.0.0 via 209.165.
202.129
*Apr 28 21:35:43.799: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:35:43.803: RT: rib update return code: 17
*Apr 28 21:35:43.827: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.202.129    1048578

R1#
*Apr 28 21:35:43.831: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:35:43.835: RT: rib update return code: 17

```

b. Verify the routing table.

```
R1#sh 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 a
rea
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external typ
e 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l
- LISP
       + - replicated route, % - next hop override

Gateway of last resort is 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, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1
```

c. Verify the IP SLA statistics.

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

IPSLA operation id: 11
      Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 21:39:41 UTC Thu Apr 28 2022
Latest operation return code: Timeout
Number of successes: 194
Number of failures: 27
Operation time to live: Forever

IPSLA operation id: 22
      Latest RTT: 32 milliseconds
Latest operation start time: 21:39:43 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 190
Number of failures: 1
Operation time to live: Forever
```

- 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
VRF info: (vrf in name/id, vrf out name/id)
  0  209.165.202.129  32 msec 20 msec 28 msec
```

- 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#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP1(config)#int lo1
ISP1(config-if)#no sh
ISP1(config-if)#
*Apr 28 21:42:07.311: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to up
ISP1(config-if)#
*Apr 28 21:42:07.315: %LINK-3-UPDOWN: Interface Loopback1, changed state to up
ISP1(config-if)#
```

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

```
R1#
*Apr 28 21:42:14.807: %TRACKING-5-STATE: 1 ip sla 11 reachability
Down->Up
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:42:14.807: RT: closer admin distance for 0.0.0.0, flush
ing 1 routes
*Apr 28 21:42:14.807: RT: add 0.0.0.0/0 via 209.165.201.1, static
metric [2/0]
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.202.129    1048578

*Apr 28 21:42:14.807: RT: rib update return code: 17
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.202.129    1048578

*Apr 28 21:42:14.807: RT: rib update return code: 17
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
    via 209.165.201.1    1048578

*Apr 28 21:42:14.807: RT:
R1#rib update return code: 17
R1#
```

- f. Again examine the IP SLA statistics.

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

IPSLA operation id: 11
    Latest RTT: 36 milliseconds
Latest operation start time: 21:45:11 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 213
Number of failures: 41
Operation time to live: Forever

IPSLA operation id: 22
    Latest RTT: 64 milliseconds
Latest operation start time: 21:45:13 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 223
Number of failures: 1
Operation time to live: Forever
```

- g. Verify the routing table.

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

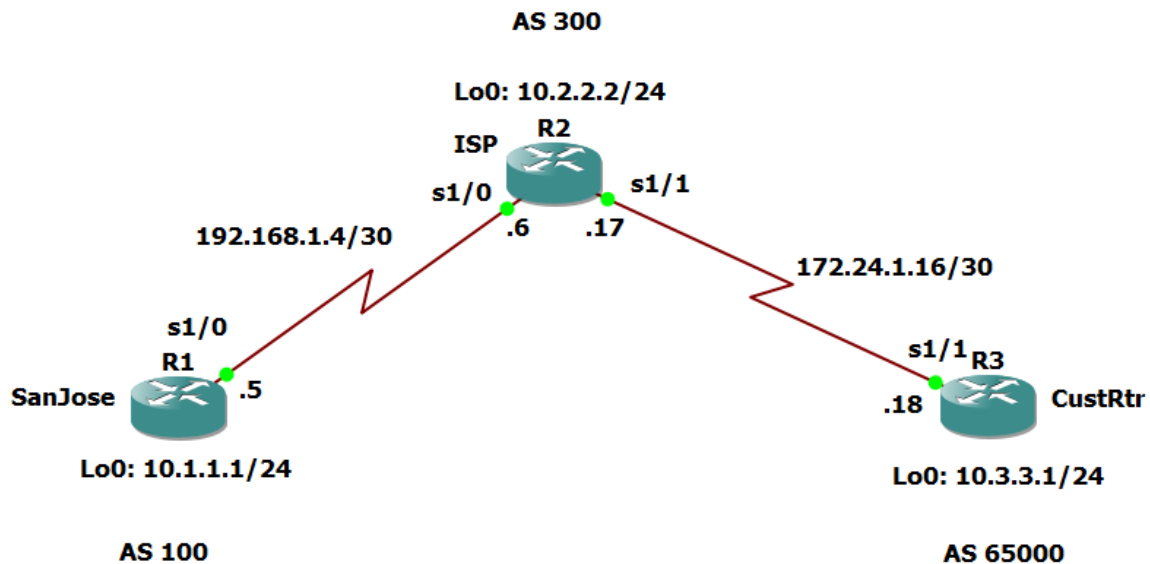
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

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, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1
```

The default static through ISP1 with an administrative distance of 2 is re-established.

## Practical 2: Using the AS\_PATH Attribute

### Topology



**Step 1: Prepare the routers for the lab.**

**Step 2: Configure the hostname and interface addresses.**

- a. Configure Routers.
- ```
#####Router R1 (hostname SanJose)
hostname SanJose

interface Loopback0
ip address 10.1.1.1 255.255.255.0
exit

interface Serial1/0
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
exit

interface Serial1/0
```



```
ip address 192.168.1.6 255.255.255.252
no shutdown
exit
```

```
interface Serial1/1
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
exit
```

```
interface Serial1/1
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).

```
SanJose#ping 10.2.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
SanJose#ping 10.3.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
SanJose#ping 172.24.1.17
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.17, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
SanJose#ping 172.24.1.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
SanJose#
```



### 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
SanJose(config-router)#
*Apr 29 19:31:03.663: %BGP-5-ADJCHANGE: neighbor 192.168.1.6
Up
SanJose(config-router)#
```

```
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
*Apr 29 19:31:03.551: %BGP-5-ADJCHANGE: neighbor 192.168.1.5
Up
ISP(config-router)#neighbor 172.24.1.18 remote-as 65000
ISP(config-router)#network 10.2.2.0 mask 255.255.255.0
ISP(config-router)#
*Apr 29 19:31:55.179: %BGP-5-ADJCHANGE: neighbor 172.24.1.18
Up
ISP(config-router)#
```

```
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
CustRtr(config-router)#
*Apr 29 19:31:55.215: %BGP-5-ADJCHANGE: neighbor 172.24.1.17
Up
CustRtr(config-router)#
```

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

```
SanJose#sh ip bgp neighbors
BGP neighbor is 192.168.1.6, remote AS 300, external link
  BGP version 4, remote router ID 10.2.2.1
  BGP state = Established, up for 00:03:36
  Last read 00:00:03, last write 00:00:02, hold time is 180,
  keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
```

```
ISP#sh 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 00:04:17
  Last read 00:00:18, last write 00:00:37, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
```

```
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 00:05:09
  Last read 00:00:45, last write 00:00:35, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
```

```
CustRtr#sh ip bgp neighbors
BGP neighbor is 172.24.1.17, remote AS 300, external link
  BGP version 4, remote router ID 10.2.2.1
  BGP state = Established, up for 00:05:44
  Last read 00:00:15, last write 00:00:05, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
```

#### 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.

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

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Loopback0
L       10.1.1.1/32 is directly connected, Loopback0
B       10.2.2.0/24 [20/0] via 192.168.1.6, 00:07:56
B       10.3.3.0/24 [20/0] via 192.168.1.6, 00:07:26
        192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.4/30 is directly connected, Serial1/0
L       192.168.1.5/32 is directly connected, Serial1/0
```

- b. Ping the 10.3.3.1 address from SanJose.

```
SanJose#ping 10.3.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- 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:
Packet sent with a source address of 10.1.1.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/64/76 ms
SanJose#
```

- 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#sh ip bgp
BGP table version is 4, local router ID is 10.1.1.1
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
*>  10.1.1.0/24      0.0.0.0            0         32768 i
*>  10.2.2.0/24      192.168.1.6        0             0 300 i
*>  10.3.3.0/24      192.168.1.6        0             0 300 65000 i
SanJose#
```

- 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
ISP(config-router)#
```

- f. After issuing these commands, use the clear ip bgp \* command on ISP to re-establish the BGP relationship between the three routers. Wait several seconds and then return to SanJose to check its routing table. 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 = 52/62/68 ms
SanJose#
```

- 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#sh ip bgp
BGP table version is 9, local router ID is 10.1.1.1
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
*>  10.1.1.0/24      0.0.0.0            0         32768 i
*>  10.2.2.0/24      192.168.1.6        0             0 300 i
*>  10.3.3.0/24      192.168.1.6        0             0 300 i
SanJose#
```

### Step 5: Use the AS\_PATH attribute to filter routes.

- 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 .*
ISP(config)#
```

- 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
ISP(config-router)#
```

- 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.

```
ISP#clear ip bgp *
ISP#
*Apr 29 19:57:07.903: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Down User reset
*Apr 29 19:57:07.903: %BGP_SESSION-5-ADJCHANGE: neighbor 172.24.1.18 IPv4 Unicast topology base removed from session User reset
*Apr 29 19:57:07.907: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down User reset
*Apr 29 19:57:07.911: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.5 IPv4 Unicast topology base removed from session User reset
*Apr 29 19:57:08.607: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
*Apr 29 19:57:08.619: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Up
```

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

Gateway of last resort is not set

```
    10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 192.168.1.5, 00:00:34
C       10.2.2.0/24 is directly connected, Loopback0
L       10.2.2.1/32 is directly connected, Loopback0
B       10.3.3.0/24 [20/0] via 172.24.1.18, 00:00:34
    172.24.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.24.1.16/30 is directly connected, Serial1/1
L       172.24.1.17/32 is directly connected, Serial1/1
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.4/30 is directly connected, Serial1/0
L       192.168.1.6/32 is directly connected, Serial1/0
```

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

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

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B       10.2.2.0/24 [20/0] via 172.24.1.17, 00:01:34
C       10.3.3.0/24 is directly connected, Loopback0
L       10.3.3.1/32 is directly connected, Loopback0
    172.24.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.24.1.16/30 is directly connected, Serial1/1
L       172.24.1.18/32 is directly connected, Serial1/1
```

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

```
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,
               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
 *> 10.1.1.0/24    192.168.1.5         0             0 100 i
```

- 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.

```
SanJose#tclsh
SanJose(tcl)#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 }
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 = 1/4/8 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 = 20/40/84 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/60/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/29/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.17, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
SanJose(tcl)#
```

```
ISP#tclsh
ISP(tcl)#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 }
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 = 24/29/36 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 10.3.3.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/27/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/59/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.17, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/59/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/32/36 ms
ISP(tcl)#
```

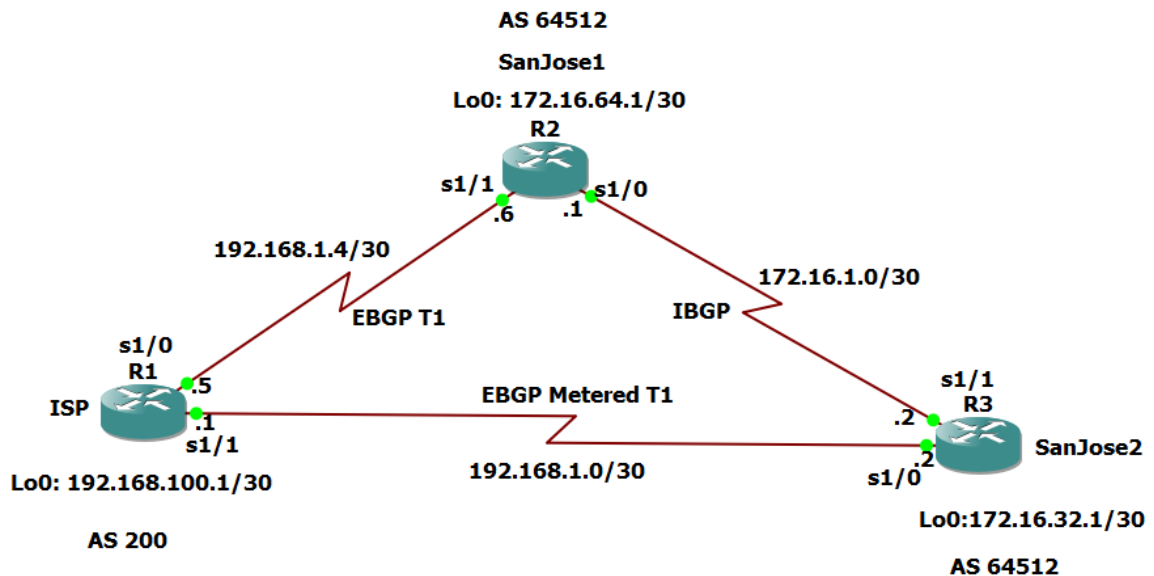


```
172.24.1.17/18 is already connected, 10.1.1.1/1
CustRtr#tclsh
CustRtr(tcl)#foreach address {
+>10.1.1.1
+>10.2.2.1
+>10.3.3.1
+>192.168.1.5
+>192.168.1.6
+>172.24.1.17
+>172.24.1.18
+>} {
+>ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
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 = 24/29/36 ms
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 = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.17, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/61/76 ms
CustRtr(tcl)#
```



### Practical 3 : Configuring IBGP and EBGP Sessions, Local Preference, and MED

#### Topology



#### Step 0: Suggested starting configurations.

- 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.

```
R1(config)#no ip domain-lookup
R1(config)#line con 0
R1(config-line)#logging synchronous
R1(config-line)#exec-timeout 0 0
R1(config-line)#
```

#### Step 1: Configure interface addresses.

- 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).

```
####ISP1
hostname ISP
int Lo0
ip add 192.168.100.1 255.255.255.0
no sh

int s1/0
ip add 192.168.1.5 255.255.255.0
no sh

int s1/1
```

```
ip add 192.168.1.1 255.255.255.0
no sh
```

```
####SanJose1
hostname SanJose1
int Lo0
ip add 172.16.64.1 255.255.255.0
no sh
```

```
int s1/1
ip add 192.168.1.6 255.255.255.0
no sh
```

```
int s1/0
ip add 172.16.1.1 255.255.255.0
no sh
```

```
####SanJose2
hostname SanJose2
int Lo0
ip add 172.16.32.1 255.255.255.0
no sh
```

```
int s1/1
ip add 172.16.1.2 255.255.255.0
no sh
```

```
int s1/0
ip add 192.168.1.2 255.255.255.0
no sh
```

- 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.

```
SanJose1#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 = 24/32/56 ms
```

```
SanJose2#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 100 percent (5/5), round-trip min/avg/max = 28/43/60 ms
```

## Step 2: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 routers.

```
SanJose1(config)#router eigrp 1
SanJose1(config-router)#network 172.16.0.0
SanJose1(config-router)#
*Apr 30 19:22:12.115: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.1.2 (Serial1/0) is up: new adjacency
SanJose1(config-router)#
```

```
SanJose2(config)#router eigrp 1
SanJose2(config-router)#network 172.16.0.0
SanJose2(config-router)#
*Apr 30 19:22:12.155: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.1.1 (Serial1/1) is up: new adjacency
SanJose2(config-router)#
```

## 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
SanJose1(config-router)#
```

- 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
SanJose2(config-router)#
*Apr 30 19:26:14.559: %BGP-5-ADJCHANGE: neighbor 172.16.64.1 Up
SanJose2(config-router)#
```

- c. Verify that SanJose1 and SanJose2 become BGP neighbors by issuing the show ip bgp neighbors command on SanJose1.

```
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 00:01:12
  Last read 00:00:17, last write 00:00:17, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
```

#### 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
ISP(config-router)#
```

- 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.

```
BGP neighbor is 192.168.1.5, remote AS 200, external link
BGP version 4, remote router ID 0.0.0.0
BGP state = Idle
Neighbor sessions:
  0 active, is not multisession capable (disabled)
Stateful switchover support enabled: NO
Default minimum time between advertisement runs is 30 seconds
```

- 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
SanJose2(config-router)#
```

### 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
*Apr 30 19:43:19.699: %SYS-5-CONFIG_I: Configured from console by console
SanJose2#show ip bgp summary
BGP router identifier 172.16.32.1, local AS number 64512
BGP table version is 3, main routing table version 3
1 network entries using 144 bytes of memory
2 path entries using 160 bytes of memory
2/1 BGP path/bestpath attribute entries using 272 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 576 total bytes of memory
BGP activity 1/0 prefixes, 2/0 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down  State
/PfxRcd
172.16.64.1    4      64512     22     22       3    0    0 00:17:08
1
192.168.1.1    4      200      0      0       1    0    0 never  Idle
```

### 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 *
```

- 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 100 percent (5/5), round-trip min/avg/max = 20/42/76 ms
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 100 percent (5/5), round-trip min/avg/max = 16/25/32 ms
```

- 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 0 percent (0/5)
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 0 percent (0/5)
```

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

```
ISP#sh ip bgp
BGP table version is 4, 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
* > 192.168.1.0      0.0.0.0               0        32768 i
* > 192.168.100.0    0.0.0.0               0        32768 i
```

- j. 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 = 28/30/32 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 0 percent (0/5)
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 0 percent (0/5)
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 = 24/28/32 ms
ISP#
```

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

- a. Issue the following commands on the ISP router.

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

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

```
ISP#sh ip bgp
BGP table version is 4, 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
* > 192.168.1.0     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#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

 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, Serial1/1
L    172.16.1.2/32 is directly connected, Serial1/1
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, 01:13:53, Serial1/1
B    192.168.1.0/24 [200/0] via 192.168.1.5, 00:11:45
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
ISP(config-router)#no network 192.168.1.4
ISP(config-router)#exit
ISP(config)#int s1/1
ISP(config-if)#sh
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#sh 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
*> 172.16.0.0        0.0.0.0              0         32768 i
* i 172.16.64.1      172.16.64.1          0         100      0 i
* i 192.168.100.0    192.168.1.5          0         100      0 200 i
SanJose2#
```

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

Gateway of last resort is not set

 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, Serial1/1
L    172.16.1.2/32 is directly connected, Serial1/1
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, 01:18:39, Serial1/1
SanJose2#
```

- 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
SanJose1(config-router)#
```

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



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

```
SanJose1#clear ip bgp *
SanJose1#
*Apr 30 20:44:53.919: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Down User reset
*Apr 30 20:44:53.919: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.32.1 IPv4 Unicast
topology base removed from session User reset
*Apr 30 20:44:53.927: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down User reset
*Apr 30 20:44:53.927: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.5 IPv4 Unicast
topology base removed from session User reset
*Apr 30 20:44:54.239: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
*Apr 30 20:44:54.283: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Up
SanJose1#
*Apr 30 20:45:06.791: %BGP-5-NBR_RESET: Neighbor 172.16.32.1 reset (Peer closed
the session)
*Apr 30 20:45:06.795: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Down Peer closed th
e session
*Apr 30 20:45:06.795: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.32.1 IPv4 Unicast
topology base removed from session Peer closed the session
*Apr 30 20:45:07.639: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Up
```

```
SanJose2#clear ip bgp *
SanJose2#
*Apr 30 20:45:06.795: %BGP-5-ADJCHANGE: neighbor 172.16.64.1 Down User reset
*Apr 30 20:45:06.799: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.64.1 IPv4 Unicast
topology base removed from session User reset
*Apr 30 20:45:07.707: %BGP-5-ADJCHANGE: neighbor 172.16.64.1 Up
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#sh ip bgp
BGP table version is 3, 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     172.16.64.1         0         100    0 i
*>i  192.168.100.0    172.16.64.1         0         100    0 200 i
```

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

Gateway of last resort is not set

    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, Serial1/1
L       172.16.1.2/32 is directly connected, Serial1/1
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, 01:26:21, Serial1/1
B       192.168.100.0/24 [200/0] via 172.16.64.1, 00:02:15
```

- 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.

```
ISP(config)#int s1/1
ISP(config-if)#no sh
ISP(config-if)#
*Apr 30 20:49:44.075: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
ISP(config-if)#
*Apr 30 20:49:45.083: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1,
changed state to up
ISP(config-if)#
```

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

Gateway of last resort is not set

    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, Serial1/1
L       172.16.1.2/32 is directly connected, Serial1/1
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, 01:28:11, Serial1/1
       192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, Serial1/0
L       192.168.1.2/32 is directly connected, Serial1/0
B       192.168.100.0/24 [20/0] via 192.168.1.1, 00:00:17
SanJose2#
```

## Step 8: Set BGP local preference.

- 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
SanJose1(config-router)#
```

```
SanJose2(config)#route-map SECONDARY_T1_IN permit 10
SanJose2(config-route-map)#set local-preference 125
SanJose2(config-route-map)#exit
SanJose2(config)#router bgp 64512
SanJose2(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
SanJose2(config-router)#
```

- 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#sh ip bgp
BGP table version is 6, 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          0.0.0.0             0              32768 i
*> 192.168.100.0   192.168.1.5         0      150      0 200 i
```

```
SanJose2#sh 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    172.16.64.1         0      100      0 i
* 192.168.100.0   192.168.1.1         0      125      0 200 i
*>i 172.16.64.1    172.16.64.1         0      150      0 200 i
```

## 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#sh ip bgp
BGP table version is 9, 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      192.168.1.2             0           0 64512 i
*>   192.168.1.6      192.168.1.6             0           0 64512 i
*>   192.168.100.0    0.0.0.0                 0          32768 i
```

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

Gateway of last resort is not set

B    172.16.0.0/16 [20/0] via 192.168.1.6, 00:12:29
    192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Serial1/1
           is directly connected, Serial1/0
L        192.168.1.1/32 is directly connected, Serial1/1
L        192.168.1.5/32 is directly connected, Serial1/0
    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
```

- 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)
(0.0.0.0)

Reply to request 0 (60 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.5)
(172.16.1.1)
(172.16.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list
```

```
Reply to request 1 (44 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.1)
(192.168.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list
```

```
Reply to request 2 (52 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.5)
(172.16.1.1)
(172.16.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list
```

```
Reply to request 3 (40 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.1)
(192.168.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list
```

```
Reply to request 4 (64 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.5)
(172.16.1.1)
(172.16.1.2) <*>
(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 = 40/52/64 ms
```

- b. Create a new policy to force the ISP router to return all traffic via SanJose1. Create a second route map utilizing the MED (metric) that is shared between EBGp neighbors.

```
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
SanJose1(config-router)#
```

```
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
SanJose2(config-router)#
```

- 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#sh ip bgp
BGP table version is 6, 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
```

```
SanJose2#sh 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
* 192.168.100.0    192.168.1.1         0      125      0 200 i
*>i              172.16.64.1         0      150      0 200 i
```



- 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)
(0.0.0.0)

Reply to request 0 (40 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.1)
(192.168.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list
```



```
Reply to request 1 (52 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
```

```
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.5)
(172.16.1.1)
(172.16.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
End of list
```

```
Reply to request 2 (36 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
```

```
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.1)
(192.168.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
End of list
```

```
Reply to request 3 (56 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
```

```
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.5)
(172.16.1.1)
(172.16.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
End of list
```

```
Reply to request 4 (44 ms). Received packet has options
Total option bytes= 40, padded length=40
```

```
Record route:
```

```
(172.16.1.2)
(192.168.1.6)
(192.168.1.5)
(192.168.1.1)
(192.168.1.2) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
End of list
```

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#sh ip bgp
BGP table version is 11, 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       192.168.1.2             75         0 64512 i
*>  172.16.0.0       192.168.1.6             50         0 64512 i
*>  192.168.100.0    0.0.0.0                 0         32768 i
```

### Step 10: Establish a default route.

- a. Configure ISP to inject a default route to both SanJose1 and SanJose2 using BGP using the default-originate command

```
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)#int lo 10
ISP(config-if)#
*Apr 30 21:24:16.787: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback10
, changed state to up
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#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

B*    0.0.0.0/0 [20/0] via 192.168.1.5, 00:00:59
     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, Serial1/0
L      172.16.1.1/32 is directly connected, Serial1/0
D      172.16.32.0/24 [90/2297856] via 172.16.1.2, 02:03:17, Serial1/0
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      192.168.1.0/24 is directly connected, Serial1/1
L      192.168.1.6/32 is directly connected, Serial1/1
B      192.168.100.0/24 [20/0] via 192.168.1.5, 00:30:36
```

```

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

Gateway of last resort is 172.16.64.1 to network 0.0.0.0

B*    0.0.0.0/0 [200/0] via 172.16.64.1, 00:02:26
      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, Serial1/1
L      172.16.1.2/32 is directly connected, Serial1/1
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, 02:04:44, Serial1/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.1.0/24 is directly connected, Serial1/0
L      192.168.1.2/32 is directly connected, Serial1/0
B      192.168.100.0/24 [200/0] via 172.16.64.1, 00:32:03

```

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

```

SanJose2#sh 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 0.0.0.0          172.16.64.1          0     150      0 200 i
*      192.168.1.1          125      0 200 i
*> 172.16.0.0        0.0.0.0              0     32768 i
* i      172.16.64.1          0     100      0 i
* 192.168.100.0      192.168.1.1          0     125      0 200 i
*>i      172.16.64.1          0     150      0 200 i

```

- 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 32 msec 24 msec 28 msec
 2 192.168.1.5 [AS 200] 44 msec 44 msec 48 msec

```

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

```
ISP(config)#int s1/0
ISP(config-if)#sh
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#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

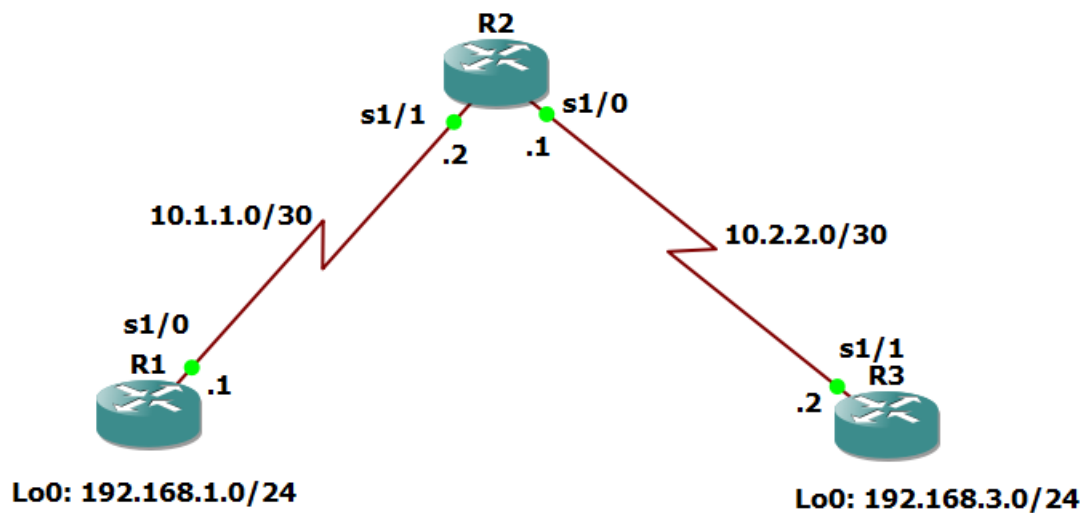
B*    0.0.0.0/0 [200/0] via 172.16.32.1, 00:00:23
      172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S      172.16.0.0/16 is directly connected, Null0
C      172.16.1.0/24 is directly connected, Serial1/0
L      172.16.1.1/32 is directly connected, Serial1/0
D      172.16.32.0/24 [90/2297856] via 172.16.1.2, 02:09:21, Serial1/0
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, 00:00:23
```

- 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 40 msec 32 msec 28 msec
  2 192.168.1.1 [AS 200] 68 msec 56 msec 64 msec
SanJose1#
```

## Practical 4: Secure the Management Plane

### Topology



### Step 1: Configure loopbacks and assign addresses.

```
####R1
interface Loopback 0
description R1 LAN
ip address 192.168.1.1 255.255.255.0
exit
```

```
interface Serial1/0
description R1 --> R2
ip address 10.1.1.1 255.255.255.252
clock rate 128000
no shutdown
exit
```

```
####R2
interface Serial1/1
description R2 --> R1
ip address 10.1.1.2 255.255.255.252
no shutdown
exit
```

```
interface Serial1/0
description R2 --> R3
ip address 10.2.2.1 255.255.255.252
clock rate 128000
no shutdown
exit
```

```
####R3
```

```

interface Loopback0
description R3 LAN
ip address 192.168.3.1 255.255.255.0
exit
interface Serial1/1
description R3 --> R2
ip address 10.2.2.2 255.255.255.252
no shutdown
exit

```

## Step 2: Configure static routes.

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

```
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.

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

- c. On R2, configure two static routes.

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

```

R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#
+>(tcl)#192.168.1.1
+>(tcl)#
+>(tcl)#10.1.1.1
+>(tcl)#
+>(tcl)#10.1.1.2
+>(tcl)#
+>(tcl)#10.2.2.1
+>(tcl)#
+>(tcl)#10.2.2.2
+>(tcl)#
+>(tcl)#192.168.3.1
+>(tcl)#
+>(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 = 1/5/8 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 = 48/63/72 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 = 20/35/56 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 = 24/28/32 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 = 56/61/76 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 = 56/60/64 ms
R1(tcl)#

```

### 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
```

- 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.

```
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?

```
R1#sh run
Building configuration...

Current configuration : 2024 bytes
!
! Last configuration change at 15:29:27 UTC Sun May 1 2022
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
security passwords min-length 10
enable secret 5 $1$5ojA$FEdQHjRk3fb4FyNOOZ9rYl
```

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

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

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

```
line con 0
exec-timeout 5 0
privilege level 15
password 7 070C285F4D061A0A19020A1F17
logging synchronous
login
stopbits 1
line aux 0
exec-timeout 0 0
privilege level 15
logging synchronous
no exec
stopbits 1
line vty 0 4
exec-timeout 5 0
password 7 121A0C0411041A10333B253B20
login
```

No. The passwords are now encrypted.



- i. Configure a warning to unauthorized users with a message-of-the-day (MOTD) banner using the banner motd command.

```
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?

```
!
banner motd ^CUnauthorized access strictly prohibited!^C
!
```

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

#### Step 4: Configure enhanced username password security.

- 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
```

- 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#
```

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

- 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
```

- 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
```

- 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)#
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
```

- 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)#$ication login TELNET-LOGIN group RADIUS-GROUP local-case
```

- g. Alter the VTY lines to use the TELNET-LOGIN AAA authenticaito0n 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.
- 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>
```

### Step 6: Enabling secure remote management using SSH.

- 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
% No Signature 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] (elapsed time was 2 seconds)

R1(config)#
*May  1 16:08:17.099: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#
```

- 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).

Configure SSH version 2 on R1.

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

- 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
```

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

```
R1#sh 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 AAAAB3NzaC1yc2EAAAADAQABAAQCAQTmUNvxm5hERb6FrDjmw1msUV8dEu0/PPMabBi+P
LSEKIIavtKYhOtaSP4R7BGprGGWPnOjFarNWz5msbfffGNLHEQvcHSAwfn3o6xiTlQTPilwhFnv2EulNu
ZGd1k8268d0/+H7ke/MdcALPgXJ+hVIB8aCEHQEnb8NL4oiCyQ==
```

- g. Repeat the steps 6a to 6f on R3.
- h. Although a user can SSH from a host using the SSH option of TeraTerm or 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#
```

```
R3#ssh -l ADMIN 10.1.1.1
Password:
Password:
Unauthorized access strictly prohibited!
R1>
R1>en
Password:
R1#
```

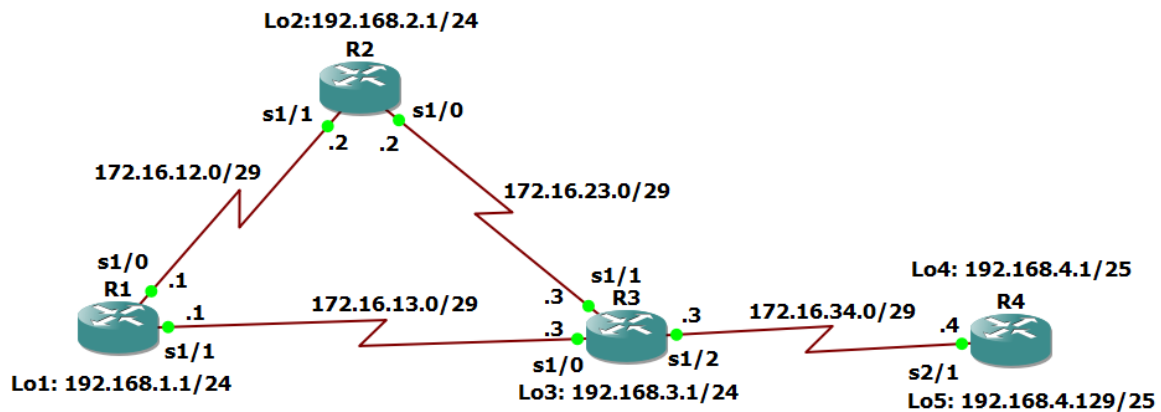
```
R1#ssh -l JR-ADMIN 10.2.2.2
Password:
Unauthorized access strictly prohibited!R3>
R3>en
Password:
R3#
```

```
R3#ssh -l JR-ADMIN 10.1.1.1
Password:
Unauthorized access strictly prohibited!
R1>en
Password:
R1#
```

## Practical 5:

### Configure and Verify Path Control

#### Topology



#### 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. #####Router R1

```
interface Lo1
description R1 LAN
ip address 192.168.1.1 255.255.255.0
exit

interface Serial1/0
description R1 --> R2
ip address 172.16.12.1 255.255.255.248
clock rate 128000
bandwidth 128
no shutdown
exit

interface Serial1/1
description R1 --> R3
ip address 172.16.13.1 255.255.255.248
bandwidth 64
no shutdown

end
```

#####Router R2

```
interface Lo2
description R2 LAN
ip address 192.168.2.1 255.255.255.0
exit
```

```
interface Serial1/1
description R2 --> R1
ip address 172.16.12.2 255.255.255.248
bandwidth 128
no shutdown
exit
```

```
interface Serial1/0
description R2 --> R3
ip address 172.16.23.2 255.255.255.248
clock rate 128000
bandwidth 128
no shutdown
```

```
end
```

```
####Router R3
```

```
interface Lo3
description R3 LAN
ip address 192.168.3.1 255.255.255.0
exit
```

```
interface Serial1/0
description R3 --> R1
ip address 172.16.13.3 255.255.255.248
clock rate 64000
bandwidth 64
no shutdown
exit
```

```
interface Serial1/1
description R3 --> R2
ip address 172.16.23.3 255.255.255.248
bandwidth 128
no shutdown
exit
```

```
interface Serial1/2
description R3 --> R4
ip address 172.16.34.3 255.255.255.248
clock rate 64000
bandwidth 64
no shutdown
```

end

####Router R4

interface Lo4

description R4 LAN A

ip address 192.168.4.1 255.255.255.128

exit

interface Lo5

description R4 LAN B

ip address 192.168.4.129 255.255.255.128

exit

interface Serial2/1

description R4 --> R3

ip address 172.16.34.4 255.255.255.248

bandwidth 64

no 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 | include up
Serial1/0      172.16.13.3      YES manual up      up
Serial1/1      172.16.23.3      YES manual up      up
Serial1/2      172.16.34.3      YES manual up      up
Loopback3      192.168.3.1       YES manual up      up
R3#
```

```
R3#sh protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
Serial1/0 is up, line protocol is up
  Internet address is 172.16.13.3/29
Serial1/1 is up, line protocol is up
  Internet address is 172.16.23.3/29
Serial1/2 is up, line protocol is up
  Internet address is 172.16.34.3/29
Serial1/3 is administratively down, line protocol is down
Serial2/0 is administratively down, line protocol is down
Serial2/1 is administratively down, line protocol is down
Serial2/2 is administratively down, line protocol is down
Serial2/3 is administratively down, line protocol is down
Serial2/4 is administratively down, line protocol is down
Serial2/5 is administratively down, line protocol is down
Serial2/6 is administratively down, line protocol is down
Serial2/7 is administratively down, line protocol is down
GigabitEthernet3/0 is administratively down, line protocol is down
Loopback3 is up, line protocol is up
  Internet address is 192.168.3.1/24
```



```
R3#sh int des | include up
Se1/0                up          up      R3 --> R1
Se1/1                up          up      R3 --> R2
Se1/2                up          up      R3 --> R4
Lo3                  up          up      R3 LAN
```

### Step 3: Configure basic EIGRP.

- Implement EIGRP AS 1 over the serial and loopback interfaces as you have configured it for the other EIGRP labs.
- Advertise networks 192.168.12.0/29, 192.168.13.0/29, 192.168.23.0/29, 192.168.34.0/29, 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.

```
R1(config)#router eigrp 1
R1(config-router)#network 192.168.1.0
R1(config-router)#network 172.16.12.0 0.0.0.7
R1(config-router)#network 172.16.13.0 0.0.0.7
R1(config-router)#no auto-summary
R1(config-router)#
*May  1 17:03:40.299: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.12.2 (Ser
ial1/0) is up: new adjacency
R1(config-router)#
*May  1 17:04:18.103: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.13.3 (Ser
ial1/1) is up: new adjacency
R1(config-router)#
```

```
R2(config)#router eigrp 1
R2(config-router)#network 192.168.2.0
R2(config-router)#network 172.16.12.0 0.0.0.7
R2(config-router)#network 172.16.23.0 0.0.0.7
R2(config-router)#no auto-summary
R2(config-router)#
*May  1 17:03:40.751: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.12.1 (Ser
ial1/1) is up: new adjacency
R2(config-router)#
```

```
R3(config)#router eigrp 1
R3(config-router)#network 192.168.3.0
R3(config-router)#network 172.16.13.0 0.0.0.7
R3(config-router)#network 172.16.23.0 0.0.0.7
R3(config-router)#network 172.16.34.0 0.0.0.7
R3(config-router)#no auto-summary
R3(config-router)#
*May  1 17:04:18.079: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.23.2 (Ser
ial1/1) is up: new adjacency
*May  1 17:04:18.083: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.13.1 (Ser
ial1/0) is up: new adjacency
R3(config-router)#
```

```

R4(config)#router eigrp 1
R4(config-router)#network 192.168.4.0
R4(config-router)#network 172.16.34.0 0.0.0.7
R4(config-router)#no auto-summary
R4(config-router)#
*May  1 17:05:03.807: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 172.16.34.3 (Serial2/1) is up: new adjacency
R4(config-router)#

```

You should see EIGRP neighbor relationship messages being generated.

#### Step 4: Verify EIGRP connectivity.

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

```

R1#sh ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                Interface      Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)        Cnt  Num
1   172.16.13.3             Se1/1         14 00:03:09    58   2340  0  10
0   172.16.12.2             Se1/0         13 00:03:47    55   1170  0   8
R1#

```

```

R2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                Interface      Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)        Cnt  Num
1   172.16.23.3             Se1/0         14 00:03:39    42   1170  0  11
0   172.16.12.1             Se1/1         13 00:04:17    61   1170  0  11
R2#

```

```

R3#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                Interface      Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)        Cnt  Num
2   172.16.34.4             Se1/2         13 00:03:05    42   2340  0   3
1   172.16.13.1             Se1/0         13 00:03:50    39   2340  0  10
0   172.16.23.2             Se1/1         14 00:03:50    40   1170  0   9
R3#

```

```

R4#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                Interface      Hold Uptime    SRTT    RTO  Q  Seq
                               (sec)          (ms)        Cnt  Num
0   172.16.34.3             Se2/1         13 00:03:16    57   2340  0   9
R4#

```

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

```

R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#192.168.4.129
+>(tcl)#} { ping $address }

```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/63/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/61/88 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/32/40 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/29/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/61/68 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/36 ms
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/64/80 ms
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 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/36 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 = 48/64/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/59/72 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.129, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/60/84 ms
R1(tcl)#
```

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

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks
C       172.16.12.0/29 is directly connected, Serial1/0
L       172.16.12.1/32 is directly connected, Serial1/0
C       172.16.13.0/29 is directly connected, Serial1/1
L       172.16.13.1/32 is directly connected, Serial1/1
D       172.16.23.0/29 [90/21024000] via 172.16.12.2, 00:13:44, Serial1/0
D       172.16.34.0/29 [90/41024000] via 172.16.13.3, 00:13:44, Serial1/1
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, Loopback1
L       192.168.1.1/32 is directly connected, Loopback1
D       192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:13:44, Serial1/0
D       192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:13:44, Serial1/0
    192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:12:58, Serial1/1
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:12:58, Serial1/1
R1#
```

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

```
R4#traceroute 192.168.1.1 source 192.168.4.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.34.3 44 msec 20 msec 28 msec
 2 172.16.23.2 64 msec 60 msec 60 msec
 3 172.16.12.1 76 msec 84 msec 76 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
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.34.3 28 msec 28 msec 24 msec
 2 172.16.23.2 80 msec 60 msec 52 msec
 3 172.16.12.1 76 msec 76 msec 76 msec
```

- 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 S1/1.

```
R3#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 7 subnets, 2 masks
D       172.16.12.0/29 [90/21024000] via 172.16.23.2, 00:19:11, Serial1/1
C       172.16.13.0/29 is directly connected, Serial1/0
L       172.16.13.3/32 is directly connected, Serial1/0
C       172.16.23.0/29 is directly connected, Serial1/1
L       172.16.23.3/32 is directly connected, Serial1/1
C       172.16.34.0/29 is directly connected, Serial1/2
L       172.16.34.3/32 is directly connected, Serial1/2
D       192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:19:11, Serial1/1
D       192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:19:11, Serial1/1
       192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, Loopback3
L       192.168.3.1/32 is directly connected, Loopback3
       192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/40640000] via 172.16.34.4, 00:18:26, Serial1/2
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:18:26, Serial1/2
```

- d. On R3, use the show interfaces serial 1/0 and show interfaces s1/1 commands.

```
R3#sh int s1/0
Serial1/0 is up, line protocol is up
  Hardware is M4T
  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, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  Last input 00:00:02, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing 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/sec
    680 packets input, 50081 bytes, 0 no buffer
    Received 289 broadcasts (0 IP multicasts)
    0 runs, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
--More--
```

```

R3#sh int s1/1
Serial1/1 is up, line protocol is up
  Hardware is M4T
  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, crc 16, loopback not set
  Keepalive set (10 sec)
  Restart-Delay is 0 secs
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing 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 96 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    735 packets input, 55200 bytes, 0 no buffer
    Received 296 broadcasts (0 IP multicasts)
    0 runs, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
--More--

```

- 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
EIGRP-IPv4 Topology Entry for AS(1)/ID(192.168.3.1) for 192.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000
  Descriptor Blocks:
    172.16.23.2 (Serial1/1), from 172.16.23.2, Send flag is 0x0
      Composite metric is (21152000/20640000), route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
        Originating router is 192.168.1.1
    172.16.13.1 (Serial1/0), from 172.16.13.1, Send flag is 0x0
      Composite metric is (40640000/128256), route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop 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.

- 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 traffic
R3(config-std-nacl)#permit 192.168.4.128 0.0.0.127
R3(config-std-nacl)#exit
R3(config)#
```

- b. Create a route map called R3-to-R1 that matches PBR-ACL and sets the next-hop interface to the R1 serial 1/1 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 172.16.13.1
R3(config-route-map)#exit
R3(config)#
```

- 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 S1/2.

```
R3(config)#int s1/2
R3(config-if)#ip policy route-map R3-to-R1
R3(config-if)#end
R3#
```

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

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

## Step 7: Test the policy.

- 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
R3#
```

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

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

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

- 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
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.34.3 48 msec 16 msec 32 msec
 2 172.16.23.2 60 msec 68 msec 60 msec
 3 172.16.12.1 76 msec 80 msec 72 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#
*May 1 17:47:26.211: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, poli
cy rejected -- normal forwarding
*May 1 17:47:26.267: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, poli
cy rejected -- normal forwarding
*May 1 17:47:26.287: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, poli
cy rejected -- normal forwarding
*May 1 17:47:26.335: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
*May 1 17:47:26.391: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
*May 1 17:47:26.459: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1
R3#, len 28, FIB policy rejected(no match) - normal forwarding
*May 1 17:47:26.511: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
*May 1 17:47:26.587: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
*May 1 17:47:26.671: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
R3#
```

- 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
VRF info: (vrf in name/id, vrf out name/id)
 1 172.16.34.3 32 msec 32 msec 20 msec
 2 172.16.13.1 60 msec 64 msec 60 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#.168.4.129 (Serial1/2), d=192.168.1.1, len 28, FIB policy match
*May  1 17:49:21.555: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, len 28, PB
R Counted
*May  1 17:49:21.555: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, g=172.16.1
3.1, len 28, FIB policy routed
*May  1 17:49:21.607: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, len 28, FI
B policy match
*May  1 17:49:21.607: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, len 28, PB
R Counted
*May  1 17:49:21.611: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, g=172.16.1
3.1, len 28, FIB policy routed
*May  1 17:49:21.671: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, len 28, FI
B policy match
*May  1 17:49:21.671: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, len 28, PB
R Counted
*May  1 17:49:21.675: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1, g=172.16.1
3.1, len 28, FIB policy routed
R3#
```

- 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-ACL
  Set clauses:
    ip next-hop 172.16.13.1
Nexthop tracking current: 0.0.0.0
172.16.13.1, fib_nh:0,oce:0,status:0

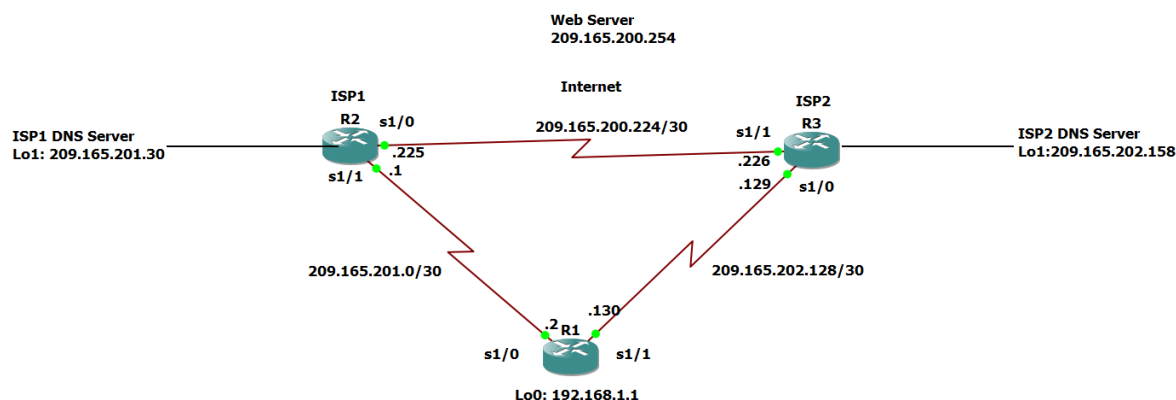
  Policy routing matches: 6 packets, 192 bytes
R3#
```

There are now matches to the policy because packets matching the ACL have passed through R3 S0/1/0.

## Practical 6:

### Configure IP SLA Tracking and Path Control

#### TOPOLOGY



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

- a. Cable the network as shown in the topology diagram.

####Router R1

hostname R1

```
interface Loopback 0
description R1 LAN
ip address 192.168.1.1 255.255.255.0
exit
```

```
interface Serial1/0
description R1 --> ISP1
ip address 209.165.201.2 255.255.255.252
clock rate 128000
bandwidth 128
no shutdown
exit
```

```
interface Serial1/1
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
exit
```

```
interface Loopback1
description ISP1 DNS Server
ip address 209.165.201.30 255.255.255.255
exit
```

```
interface Serial1/1
description ISP1 --> R1
ip address 209.165.201.1 255.255.255.252
bandwidth 128
no shutdown
exit
```

```
interface Serial1/0
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
exit
```

```
interface Loopback1
description ISP2 DNS Server
ip address 209.165.202.158 255.255.255.255
exit
```

```
interface Serial1/0
description ISP2 --> R1
ip address 209.165.202.129 255.255.255.252
clock rate 128000
bandwidth 128
no shutdown
exit
```

```
interface Serial1/1
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.

R1# show interfaces description

```
R1#sh int des | include up
Se1/0                up          up          R1 --> ISP1
Se1/1                up          up          R1 --> ISP2
Lo0                  up          up          R1 LAN
R1#
```

- 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.

Implement the routing policies on the respective routers.

```
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
R1(config)#
```

```
ISP1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
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)#ip route 192.168.1.0 255.255.255.0 209.165.201.2
ISP1(config)#
*Apr 28 20:47:05.587: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.226
(Serial1/0) is up: new adjacency
ISP1(config)#
```

```
ISP2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
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)#
*Apr 28 20:47:05.351: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.225
(Serial1/1) is up: new adjacency
ISP2(config-router)#exit
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.

```
*Apr 28 20:47:05.587: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 209.165.200.226
(Serial1/0) is up: new adjacency
```

## Step 2: Verify server reachability

- 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.

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

```
R1#tclsh  
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 = 20/26/32 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 = 20/30/44 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 = 32/39/48 ms  
R1(tcl)#
```

- b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server.

```
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  
VRF info: (vrf in name/id, vrf out name/id)  
  1 209.165.201.1 32 msec 28 msec 32 msec  
Type escape sequence to abort.  
Tracing the route to 209.165.201.30  
VRF info: (vrf in name/id, vrf out name/id)  
  1 209.165.201.1 32 msec 36 msec 32 msec  
Type escape sequence to abort.  
Tracing the route to 209.165.202.158  
VRF info: (vrf in name/id, vrf out name/id)  
  1 209.165.201.1 40 msec 40 msec 36 msec  
  2 209.165.200.226 52 msec 56 msec 52 msec  
R1(tcl)#
```

### Step 3: Configure IP SLA probes.

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

- b. 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
```

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

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

IPSLA operation id: 11
  Latest RTT: 12 milliseconds
Latest operation start time: 21:06:31 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 22
Number of failures: 0
Operation time to live: Forever
```

- 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.

```
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)#ip sla schedule 22 life forever start-time now
R1(config)#
```

- 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-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:
```

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

IPSLA operation id: 22
  Latest RTT: 68 milliseconds
Latest operation start time: 21:11:23 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 21
Number of failures: 0
Operation time to live: Forever
```

#### Step 4: Configure tracking options.

- 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
R1#
```



- b. Verify the routing table.

```
R1#sh 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 a
rea
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external typ
e 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l
- LISP
       + - replicated route, % - next hop override

Gateway of last resort is 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, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1
```

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

```
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.

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

```
R1#debug ip routing
IP routing debugging is on
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 28 21:20:36.103: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:20:36.107: RT: closer admin distance for 0.0.0.0, flush
ing 1 routes
*Apr 28 21:20:36.107: RT: add 0.0.0.0/0 via 209.165.201.1, static
metric [2/0]
*Apr 28 21:20:36.111: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:20:36.115: RT: rib update return code: 17
*Apr 28 21:20:36.119: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:20:36.123: RT: rib update return code: 17
R1(config)#
```

- 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.

```
R1(config)#track 2 ip sla 22 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
R1(config)#
*Apr 28 21:31:11.075: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:31:11.079: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.201.1    1048578

*Apr 28 21:31:11.083: RT: rib update return code: 17
*Apr 28 21:31:11.091: RT: updating static 0.0.0.0/0 (0x0):
      via 209.165.202.129   1048578

*Apr 28 21:31:11.095: RT: rib update return code: 17
```

- g. Verify the routing table again.

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

Gateway of last resort is 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
C      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.201.0/30 is directly connected, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
C      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1
```

### Step 5: Verify IP SLA operation.

```
ISP1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP1(config)#int lo1
ISP1(config-if)#sh
ISP1(config-if)#
*Apr 28 21:35:31.435: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
  changed state to down
*Apr 28 21:35:31.439: %LINK-5-CHANGED: Interface Loopback1, changed state to adm
  inistratively down
ISP1(config-if)#
```

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

```
R1#
*Apr 28 21:35:43.791: %TRACKING-5-STATE: 1 ip sla 11 reachability
  Up->Down
*Apr 28 21:35:43.791: RT: del 0.0.0.0 via 209.165.201.1, static me
  tric [2/0]
*Apr 28 21:35:43.791: RT: delete network route to 0.0.0.0/0
*Apr 28 21:35:43.791: RT: default path has been cleared
*Apr 28 21:35:43.791: RT: updating static 0.0.0.0/0 (0x0):
  via 209.165.202.129 1048578

*Apr 28 21:35:43.795: RT: add 0.0.0.0/0 via 209.165.202.129, stati
  c metric [3/0]
*Apr 28 21:35:43.799: RT: default path is now 0.0.0.0 via 209.165.
  202.129
*Apr 28 21:35:43.799: RT: updating static 0.0.0.0/0 (0x0):
  via 209.165.201.1 1048578

*Apr 28 21:35:43.803: RT: rib update return code: 17
*Apr 28 21:35:43.827: RT: updating static 0.0.0.0/0 (0x0):
  via 209.165.202.129 1048578

R1#
*Apr 28 21:35:43.831: RT: updating static 0.0.0.0/0 (0x0):
  via 209.165.201.1 1048578

*Apr 28 21:35:43.835: RT: rib update return code: 17
```

b. Verify the routing table.

```
R1#sh 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 a
rea
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external typ
e 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l
- LISP
       + - replicated route, % - next hop override

Gateway of last resort is 209.165.202.129 to network 0.0.0.0

S*    0.0.0.0/0 [3/0] via 209.165.202.129
C      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
L      192.168.1.0/24 is directly connected, Loopback0
L      192.168.1.1/32 is directly connected, Loopback0
C      209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.201.0/30 is directly connected, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
C      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1
```

c. Verify the IP SLA statistics.

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

IPSLA operation id: 11
    Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 21:39:41 UTC Thu Apr 28 2022
Latest operation return code: Timeout
Number of successes: 194
Number of failures: 27
Operation time to live: Forever

IPSLA operation id: 22
    Latest RTT: 32 milliseconds
Latest operation start time: 21:39:43 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 190
Number of failures: 1
Operation time to live: Forever
```

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
VRF info: (vrf in name/id, vrf out name/id)
  0 209.165.202.129 32 msec 20 msec 28 msec
```

- 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#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP1(config)#int lo1
ISP1(config-if)#no sh
ISP1(config-if)#
*Apr 28 21:42:07.311: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to up
ISP1(config-if)#
*Apr 28 21:42:07.315: %LINK-3-UPDOWN: Interface Loopback1, changed state to up
ISP1(config-if)#
```

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

```
R1#
*Apr 28 21:42:14.807: %TRACKING-5-STATE: 1 ip sla 11 reachability
Down->Up
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
via 209.165.201.1 1048578

*Apr 28 21:42:14.807: RT: closer admin distance for 0.0.0.0, flush
ing 1 routes
*Apr 28 21:42:14.807: RT: add 0.0.0.0/0 via 209.165.201.1, static
metric [2/0]
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
via 209.165.202.129 1048578

*Apr 28 21:42:14.807: RT: rib update return code: 17
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
via 209.165.202.129 1048578

*Apr 28 21:42:14.807: RT: rib update return code: 17
*Apr 28 21:42:14.807: RT: updating static 0.0.0.0/0 (0x0):
via 209.165.201.1 1048578

*Apr 28 21:42:14.807: RT:
R1#rib update return code: 17
R1#
```

- f. Again examine the IP SLA statistics.

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

IPSLA operation id: 11
Latest RTT: 36 milliseconds
Latest operation start time: 21:45:11 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 213
Number of failures: 41
Operation time to live: Forever

IPSLA operation id: 22
Latest RTT: 64 milliseconds
Latest operation start time: 21:45:13 UTC Thu Apr 28 2022
Latest operation return code: OK
Number of successes: 223
Number of failures: 1
Operation time to live: Forever
```

g. Verify the routing table.

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

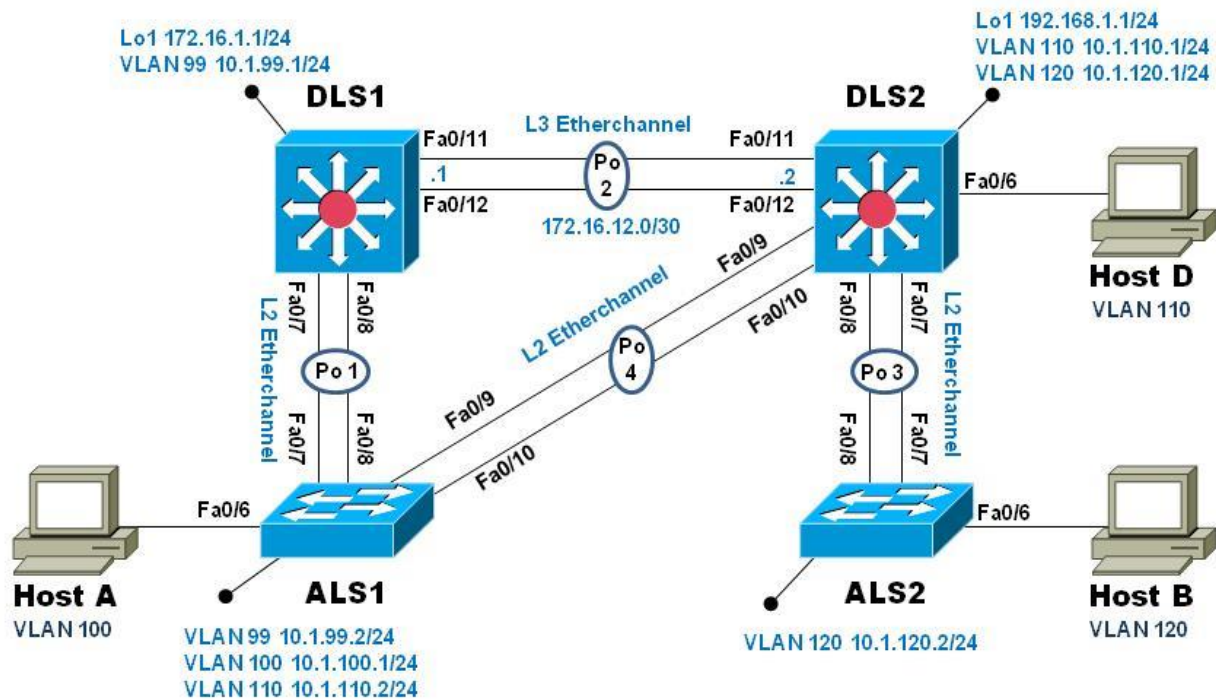
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

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, Serial1/0
L      209.165.201.2/32 is directly connected, Serial1/0
      209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks
C      209.165.202.128/30 is directly connected, Serial1/1
L      209.165.202.130/32 is directly connected, Serial1/1
```

The default static through ISP1 with an administrative distance of 2 is re-established.

## Practical 7: Inter-VLAN Routing

### Topology



### Part 1: Configure Multilayer Switching using Distribution Layer Switches

#### Step 1: Load base config

Use the reset.tcl script you created in Lab 1 “Preparing the Switch” to set your switches up for this lab. Then load the file BASE.CFG into the running-config with the command **copy flash:BASE.CFG running-config**. An example from DLS1:

```
DLS1# tclsh reset.tcl
```

```
Erasing the nvram filesystem will remove all configuration files! Continue? [confirm]  
[OK]
```

```
Erase of nvram: complete
```

```
Reloading the switch in 1 minute, type reload cancel to halt
```

```
Proceed with reload? [confirm]
```

```
*Mar 7 18:41:40.403: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
```

```
*Mar 7 18:41:41.141: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload command.
```



<switch reloads - output omitted>

Would you like to enter the initial configuration dialog? [yes/no]: n

Switch> **en**

\*Mar 1 00:01:30.915: %LINK-5-CHANGED: Interface Vlan1, changed state to administratively down

Switch# **copy BASE.CFG running-config**

Destination filename [running-config]?

184 bytes copied in 0.310 secs (594 bytes/sec)

DLS1#

## Step 2: Verify switch management database configuration

At each switch, use the show sdm prefer command to verify the appropriate template is chosen. The DLS switches should be using the "dual ipv4-and-ipv6 routing" template and the ALS switches should be using the "lanbase-routing" template. If any of the switches are using the wrong template, make the necessary change and reboot the switch with the **reload** command. An example from ALS1 is below:

ALS1# **sho sdm pref**

The current template is "default" template.

<output omitted>

ALS1# **conf t**

Enter configuration commands, one per line. End with CNTL/Z.

ALS1(config)# **sdm pref lanbase-routing**

Changes to the running SDM preferences have been stored, but cannot take effect until the next reload.

Use 'show sdm prefer' to see what SDM preference is currently active.

ALS1(config)# **end**

ALS1# **reload**

System configuration has been modified. Save? [yes/no]: y

\*Mar 1 02:12:00.699: %SYS-5-CONFIG\_I: Configured from console by console

Building configuration...

[OK]

Proceed with reload? [confirm]

## Step 3: Configure layer 3 interfaces on the DLS switches

Enable IP Routing, create broadcast domains (VLANs), and configure the DLS switches with the layer 3 interfaces and addresses shown:

| Switch | Interface | Address/Mask |
|--------|-----------|--------------|
| DLS1   | VLAN 99   | 10.1.99.1/24 |



|      |            |                |
|------|------------|----------------|
| DLS1 | Loopback 1 | 172.16.1.1/24  |
| DLS2 | VLAN 110   | 10.1.110.1/24  |
| DLS2 | VLAN 120   | 10.1.120.1/24  |
| DLS2 | Loopback 1 | 192.168.2.1/24 |

An example from DLS2:

```

DLS2(config)# ip routing
DLS2(config)# vlan 110
DLS2(config-vlan)# name Management
DLS2(config-vlan)# exit
DLS2(config)# vlan 120
DLS2(config-vlan)# name Local
DLS2(config-vlan)# exit
DLS2(config)# int vlan 110
DLS2(config-if)# ip address 10.1.110.1 255.255.255.0
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)# int vlan 120
DLS2(config-if)# ip address 10.1.120.1 255.255.255.0
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)# int loopback 1
DLS2(config-if)# ip address 192.168.1.1 255.255.255.0
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)#

```

At this point, basic intervlan routing can be demonstrated using an attached host. Host D is attached to DLS2 via interface Fa0/6. On DLS2, assign interface Fa0/6 to VLAN 110 and configure the host with the address 10.1.110.50/24 and default gateway of 10.1.110.1. Once you have done that, try and ping Loopback 1's IP address (192.168.1.1). This should work just like a hardware router; the switch will provide connectivity between two directly connected interfaces. In the output below, the **switchport host** macro was used to quickly configure interface Fa0/6 with host-relative commands:

```

DLS2(config)# int f0/6
DLS2(config-if)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled

DLS2(config-if)# switchport access vlan 110

```

```
DLS2(config-if)# no shut
DLS2(config-if)# exit
DLS2(config)#
```

```
C:\Windows\system32\cmd.exe
Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::59a7:56ce:f785:ed46%11
    IPv4 Address. . . . . : 10.1.110.50
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.1.110.1

C:\Users\student>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=2ms TTL=255
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms

C:\Users\student>
```

#### Step 4: Configure a Layer 3 Etherchannel between DLS1 and DLS2

Now you will interconnect the multilayer switches in preparation to demonstrate other routing capabilities. Configure a layer 3 EtherChannel between the DLS switches. This will provide the benefit of increased available bandwidth between the two multilayer switches. To convert the links from layer 2 to layer 3, issue the **no switchport** command. Then, combine interfaces F0/11 and F0/12 into a single PAgP EtherChannel and then assign an IP address as shown.

|      |                |      |                |
|------|----------------|------|----------------|
| DLS1 | 172.16.12.1/30 | DLS2 | 172.16.12.2/30 |
|------|----------------|------|----------------|

Example from DLS1:

```
DLS1(config)# interface range f0/11-12
DLS1(config-if-range)# no switchport
DLS1(config-if-range)# channel-group 2 mode desirable
Creating a port-channel interface Port-channel 2

DLS1(config-if-range)# no shut
DLS1(config-if-range)# exit
DLS1(config)# interface port-channel 2
DLS1(config-if)# ip address 172.16.12.1 255.255.255.252
DLS1(config-if)# no shut
DLS1(config-if)# exit
DLS1(config)#
```

DLS2# show ip route

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

<output omitted>

Gateway of last resort is 172.16.12.1 to network 0.0.0.0

```
S* 0.0.0.0/0 [1/0] via 172.16.12.1, Port-channel2
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C   10.1.110.0/24 is directly connected, Vlan110
L   10.1.110.1/32 is directly connected, Vlan110
    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C   172.16.12.0/30 is directly connected, Port-channel2
L   172.16.12.2/32 is directly connected, Port-channel2
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.1.0/24 is directly connected, Loopback1
L   192.168.1.1/32 is directly connected, Loopback1
```

DLS2# **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 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms

DLS2#

## Step 6: Configure the remaining EtherChannels for the topology

Configure the remaining EtherChannel links as layer 2 PagP trunks using VLAN 1 as the native VLAN.

| Endpoint 1   | Channel number | Endpoint 2   | VLANs Allowed  |
|--------------|----------------|--------------|----------------|
| ALS1 F0/7-8  | 1              | DLS1 F0/7-8  | All except 110 |
| ALS1 F0/9-10 | 4              | DLS2 F0/9-10 | 110 Only       |
| ALS2 F0/7-8  | 3              | DLS2 F0/7-8  | All            |

Example from ALS1:

```
ALS1(config)# interface range f0/7-8
```

```
ALS1(config-if-range)# switchport mode trunk
```

```
ALS1(config-if-range)# switchport trunk allowed vlan except 110
```

```
ALS1(config-if-range)# channel-group 1 mode desirable
```

Creating a port-channel interface Port-channel 1

```
ALS1(config-if-range)# no shut
ALS1(config-if-range)# exit
ALS1(config)# interface range f0/9-10
ALS1(config-if-range)# switchport mode trunk
ALS1(config-if-range)# switchport trunk allowed vlan 110
ALS1(config-if-range)# channel-group 4 mode desirable
```

Creating a port-channel interface Port-channel 4

```
ALS1(config-if-range)# no shut
ALS1(config-if-range)# exit
ALS1(config)#end
ALS1# show etherchannel summary
```

Flags: D - down      P - bundled in port-channel

      I - stand-alone s - suspended

      H - Hot-standby (LACP only)

      R - Layer3      S - Layer2

      U - in use      f - failed to allocate aggregator

      M - not in use, minimum links not met

      u - unsuitable for bundling

      w - waiting to be aggregated

      d - default port

Number of channel-groups in use: 2

Number of aggregators:        2

Group Port-channel Protocol Ports

```
-----+-----+-----+-----+-----+
1  Po1(SU)    PAgP   Fa0/7(P) Fa0/8(P)
4  Po4(SU)    PAgP   Fa0/9(P) Fa0/10(P)
```

ALS1# show interface trunk

| Port | Mode | Encapsulation | Status   | Native vlan |
|------|------|---------------|----------|-------------|
| Po1  | on   | 802.1q        | trunking | 1           |
| Po4  | on   | 802.1q        | trunking | 1           |

Port Vlan allowed on trunk

Po1 1-109,111-4094

Po4 110

<output omitted>

ALS1#

## Step 7: Enable and Verify Layer 3 connectivity across the network

In this step we will enable basic connectivity from the management VLANs on both sides of the network.

- Create the management VLANs (99 at ALS1, 120 at ALS2)
- Configure interface VLAN 99 at ALS1 and interface VLAN 120 at ALS2
- Assign addresses (refer to the diagram) and default gateways (at DLS1/DLS2 respectively).

Once that is all done, pings across the network should work, flowing across the layer 3 EtherChannel. An example from ALS2:

```
ALS2(config)# vlan 120
ALS2(config-vlan)# name Management
ALS2(config-vlan)# exit
ALS2(config)# int vlan 120
ALS2(config-if)# ip address 10.1.120.2 255.255.255.0
ALS2(config-if)# no shut
ALS2(config-if)# exit
ALS2(config)# ip default-gateway 10.1.120.1
ALS2(config)# end
```

```
ALS2# ping 10.1.99.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.99.2, timeout is 2 seconds:

..!!!

Success rate is 60 percent (3/5), round-trip min/avg/max = 1/3/8 ms

```
ALS2#
```

```
ALS2# traceroute 10.1.99.2
```

Type escape sequence to abort.

Tracing the route to 10.1.99.2

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.1.120.1 0 msec 0 msec 8 msec
```

```
 2 172.16.12.1 0 msec 0 msec 8 msec
```

```
 3 10.1.99.2 0 msec 0 msec *
```

```
ALS2#
```

## Part 2: Configure Multilayer Switching at ALS1

At this point all routing is going through the DLS switches, and the port channel between ALS1 and DLS2 is not passing anything but control traffic (BPDUs, etc).

The Cisco 2960 is able to support basic routing when it is using the LANBASE IOS. In this step you will configure ALS1 to support multiple SVIs and configure it for basic static routing. The objectives of this step are:

- Enable intervlan routing between two VLANs locally at ALS1
- Enable IP Routing
- Configure a static route for DLS2's Lo1 network travel via Port-Channel 4.

### Step 1: Configure additional VLANs and VLAN interfaces

At ALS1, create VLAN 100 and VLAN 110 and then create SVIs for those VLANs:

```
ALS1(config)# ip routing
ALS1(config)# vlan 100
ALS1(config-vlan)# name Local
ALS1(config-vlan)# exit
ALS1(config)# vlan 110
ALS1(config-vlan)# name InterNode
ALS1(config-vlan)# exit
ALS1(config)# int vlan 100
ALS1(config-if)# ip address 10.1.100.1 255.255.255.0
ALS1(config-if)# no shut
ALS1(config-if)# exit
ALS1(config)# int vlan 110
ALS1(config-if)# ip address 10.1.110.2 255.255.255.0
ALS1(config-if)# no shut
ALS1(config-if)# exit
ALS1(config)#
```

### Step 2: Configure and test Host Access

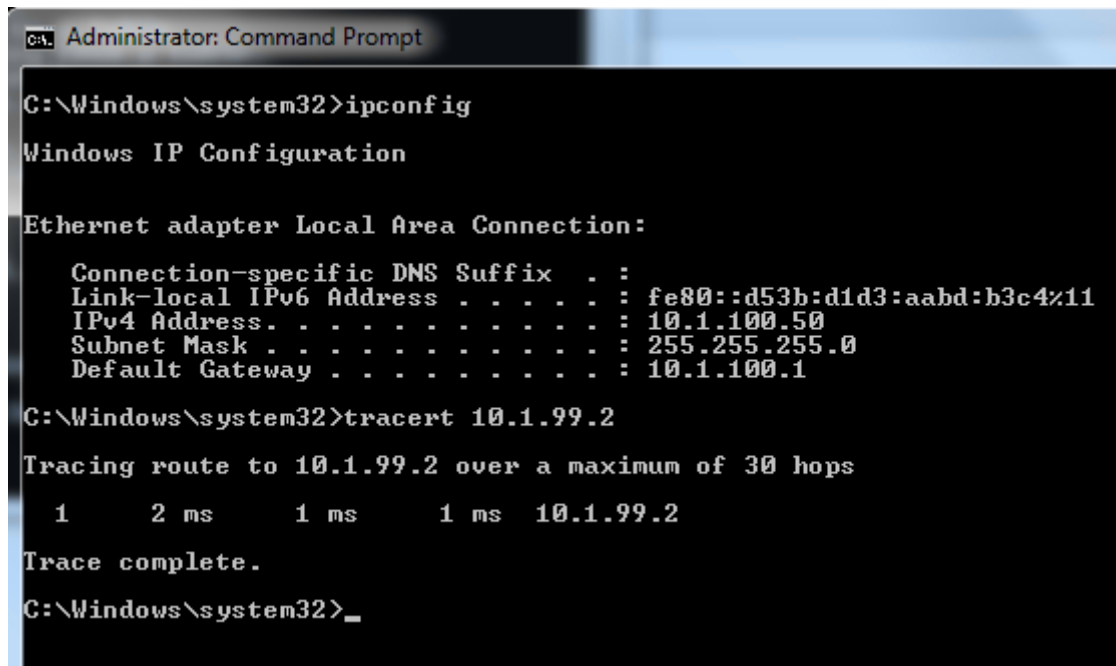
Assign interface Fa0/6 to VLAN 100. On the attached host (Host A) configure the IP address 10.1.100.50/24 with a default gateway of 10.1.100.1. Once configured, try a traceroute from the host to 10.1.99.2 and observe the results.

In the output below, the **switchport host** macro was used to quickly configure interface Fa0/6 with host-relative commands.

```
ALS1(config)# interface f0/6
ALS1(config-if)# switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled

ALS1(config-if)# switchport access vlan 100
ALS1(config-if)# no shut
```

ALS1(config-if)# **exit**



```
Administrator: Command Prompt

C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::d53b:d1d3:aabd:b3c4%11
    IPv4 Address. . . . . : 10.1.100.50
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.1.100.1

C:\Windows\system32>tracert 10.1.99.2

Tracing route to 10.1.99.2 over a maximum of 30 hops
  0  1 ms    1 ms    1 ms  10.1.99.2
Trace complete.

C:\Windows\system32>_
```

The output from the host shows that attempts to communicate with interface VLAN 99 at ALS1 were fulfilled locally, and not sent to DLS1 for routing.

### Step 3: Configure and verify static routing across the network

At this point, local routing (at ALS1) works, and off-net routing (outside of ALS1) will not work, because DLS1 doesn't have any knowledge of the 10.1.100.0 subnet. In this step you will configure routing on several different switches:

- At DLS1, configure:
  - a static route to the 10.1.100.0/24 network via VLAN 99
- At DLS2, configure
  - a static route to the 10.1.100.0/24 network via VLAN 110
- At ALS1, configure
  - a static route to the 192.168.1.0/24 network via VLAN 110
  - a default static route to use 10.1.99.1

Here is an example from ALS1:

```
ALS1(config)# ip route 192.168.1.0 255.255.255.0 vlan 110
ALS1(config)# ip route 0.0.0.0 0.0.0.0 10.1.99.1
```



ALS1(config)# **end**

ALS1# **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, I - LISP

+ - replicated route, % - next hop override

Gateway of last resort is 10.1.99.1 to network 0.0.0.0

**S\* 0.0.0.0/0 [1/0] via 10.1.99.1**

10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks

C 10.1.99.0/24 is directly connected, Vlan99

L 10.1.99.2/32 is directly connected, Vlan99

C 10.1.100.0/24 is directly connected, Vlan100

L 10.1.100.1/32 is directly connected, Vlan100

C 10.1.110.0/24 is directly connected, Vlan110

L 10.1.110.2/32 is directly connected, Vlan110

**S 192.168.1.0/24 is directly connected, Vlan110**

After configuring all of the required routes, test to see that the network behaves as expected.

From ALS1, a traceroute to 10.1.120.2 should take three hops:

ALS1# **traceroute 10.1.120.2**

Type escape sequence to abort.

Tracing the route to 10.1.120.2

VRF info: (vrf in name/id, vrf out name/id)

1 10.1.99.1 0 msec 0 msec 0 msec

2 172.16.12.2 9 msec 0 msec 0 msec

3 10.1.120.2 0 msec 8 msec \*

ALS1#

From ALS1, a traceroute to 192.168.1.1 should take one hop:

ALS1# **traceroute 192.168.1.1**

Type escape sequence to abort.

Tracing the route to 192.168.1.1

VRF info: (vrf in name/id, vrf out name/id)

1 10.1.110.1 0 msec 0 msec \*

ALS1#

Traces from Host A show an additional hop, but follow the appointed path:

```
C:\Windows\system32\cmd.exe

G:\Users\student>tracert 10.1.120.2

Tracing route to 10.1.120.2 over a maximum of 30 hops

  1      1 ms      1 ms      1 ms  10.1.100.1
  2      *        2 ms      1 ms  10.1.99.1
  3      1 ms      2 ms      1 ms  172.16.12.2
  4      1 ms      1 ms      1 ms  10.1.120.2

Trace complete.

G:\Users\student>tracert 192.168.1.1

Tracing route to 192.168.1.1 over a maximum of 30 hops

  1      1 ms      1 ms      1 ms  10.1.100.1
  2      1 ms      1 ms      1 ms  192.168.1.1

Trace complete.

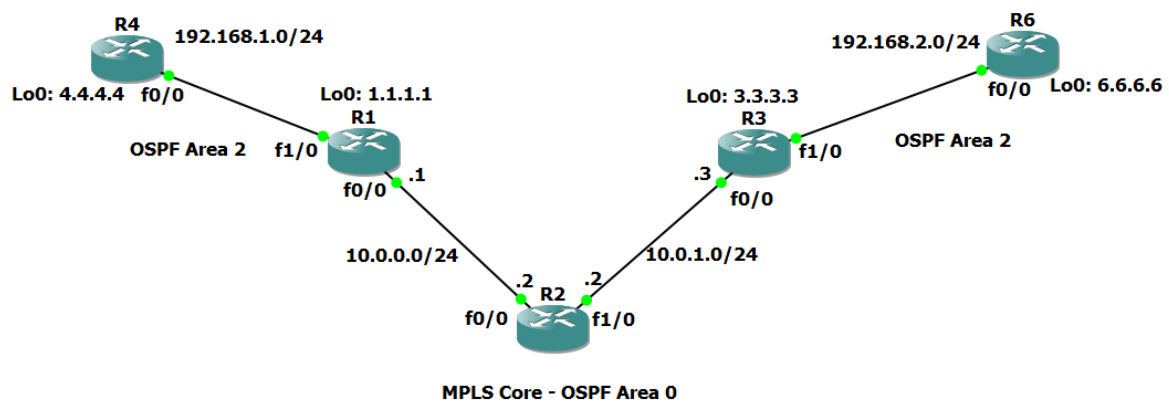
G:\Users\student>_
```

#### Step 4: End of Lab

Save your configurations. The switches will be used as configured now for lab 5-2, DHCP.

## Practical 8: Simulating MPLS environment

### Topology



### Cisco MPLS Configuration Commands

#### Step 1 – IP addressing of MPLS Core and OSPF

- a. We are going to address the routers and configure ospf to ensure loopback to loopback connectivity between R1 and R3

**\*R1\***

```
int lo0
ip add 1.1.1.1 255.255.255.255
ip ospf 1 area 0
exit
```

```
int f0/0
ip add 10.0.0.1 255.255.255.0
no shut
ip ospf 1 area 0
exit
```

**\*R2\***

```
int lo0
ip add 2.2.2.2 255.255.255.255
ip ospf 1 are 0
exit
```

```
int f0/0
```

```
ip add 10.0.0.2 255.255.255.0
no shut
ip ospf 1 area 0
exit
```

```
int f1/0
ip add 10.0.1.2 255.255.255.0
no shut
ip ospf 1 area 0
exit
```

```
*R3*
int lo0
ip add 3.3.3.3 255.255.255.255
ip ospf 1 are 0
exit
```

```
int f0/0
ip add 10.0.1.3 255.255.255.0
no shut
ip ospf 1 area 0
exit
```

- b. You should now have full ip connectivity between R1, R2, R3 to verify this we 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 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/66/96 ms
R1#
```

## 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
- a. 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 specific process.

```
R1(config)#router ospf 1
R1(config-router)#mpls ldp autoconfig
R1(config-router)#
*May  3 14:49:25.583: %PARSE_RC-3-PRC_INVALID_BLOCK_PTR:
R1(config-router)#
*May  3 14:50:05.407: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is UP
R1(config-router)#
```

```
R2(config)#router ospf 1
R2(config-router)#mpls ldp autoconfig
R2(config-router)#
```

```
R3(config)#router ospf 1
R3(config-router)#mpls ldp autoconfig
R3(config-router)#
*May  3 14:50:40.111: %PARSE_RC-3-PRC_INVALID_BLOCK_PTR:
R3(config-router)#
*May  3 14:50:40.491: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is UP
R3(config-router)#
```

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

```
R2(config-router)#
*May  3 14:50:05.515: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
R2(config-router)#
*May  3 14:50:40.511: %LDP-5-NBRCHG: LDP Neighbor 3.3.3.3:0 (2) is UP
R2(config-router)#
```

- b. 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 mpls and using LDP

```
R2#sh mpls int
Interface                IP                Tunnel    BGP Static Operational
FastEthernet0/0          Yes (ldp)         No        No  No    Yes
FastEthernet1/0          Yes (ldp)         No        No  No    Yes
R2#
```

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 2.2.2.2:0
    TCP connection: 1.1.1.1.646 - 2.2.2.2.26664
    State: Oper; Msgs sent/rcvd: 14/13; Downstream
    Up time: 00:05:19
    LDP discovery sources:
      FastEthernet0/0, Src IP addr: 10.0.0.1
    Addresses bound to peer LDP Ident:
      10.0.0.1        1.1.1.1
  Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
    TCP connection: 3.3.3.3.23570 - 2.2.2.2.646
    State: Oper; Msgs sent/rcvd: 13/13; Downstream
    Up time: 00:04:44
    LDP discovery sources:
      FastEthernet1/0, Src IP addr: 10.0.1.3
    Addresses bound to peer LDP Ident:
      10.0.1.3        3.3.3.3
```

- c. 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
VRF info: (vrf in name/id, vrf out name/id)
  1 10.0.0.2 [MPLS: Label 17 Exp 0] 156 msec 60 msec 32 msec
  2 10.0.1.3 52 msec 52 msec 64 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.

### Step 3 – MPLS BGP Configuration between R1 and R3

- d. We need to establish a Multi Protocol BGP session between R1 and R3 this is done by configuring the vpnv4 address family as below

```
R1(config)#router bgp 1
R1(config-router)#neighbor 3.3.3.3 remote-as 1
R1(config-router)# neighbor 3.3.3.3 update-source Loopback0
R1(config-router)# no auto-summary
R1(config-router)# !
R1(config-router)# address-family vpnv4
R1(config-router-af)#neighbor 3.3.3.3 activate
R1(config-router-af)#
```

```
R3(config)#router bgp 1
R3(config-router)# neighbor 1.1.1.1 remote-as 1
R3(config-router)# neighbor 1.1.1.1 update-source Loopback0
R3(config-router)# no auto-summary
R3(config-router)# !
R3(config-router)#exit
R3(config)#
*May 3 15:01:05.387: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
```

```
R3(config-router)#address-family vpnv4
R3(config-router-af)# neighbor 1.1.1.1 activate
R3(config-router-af)#
*May 3 15:01:43.591: %BGP-5-NBR_RESET: Neighbor 1.1.1.1 reset (Capability changed)
*May 3 15:01:43.595: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down Capability changed
*May 3 15:01:43.595: %BGP_SESSION-5-ADJCHANGE: neighbor 1.1.1.1 IPv4 Unicast topology base removed from session Capability changed
*May 3 15:01:44.283: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
```

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

- e. 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 version 1

Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ Up/Down  State/PfxRcd
3.3.3.3       4        1     10     10       1    0    0 00:05:08      0
```

### Step 4 – Add two more routers, create VRFs

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

- f. Router 4 will peer OSPF using process number 2 to a VRF configured on R1. 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 2
exit
```

```
int f0/0
ip add 192.168.1.4 255.255.255.0
ip ospf 2 area 2
no shut
exit
```

```
*R1*
int f1/0
no shut
ip add 192.168.1.1 255.255.255.0
exit
```

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

- g. So back to the topology – we now need to create a VRF on R1  
For this we will be using VRF RED

```
R1(config)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
R1(config-vrf)#
```

- h. So now we have configured the VRF on R1 we need to move the interface F0/1 into that VRF

```
R1(config)#int f1/0
R1(config-if)#ip vrf forwarding RED
```

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

```
% Interface FastEthernet1/0 IPv4 disabled and address(es) removed due to enabling VRF RED
R1(config-if)#
```

- i. You just need to re-apply it

```
R1(config)#int f1/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#
```

- j. Now if we view the config on R1 int f1/0 you can see the VRF configured.

```
R1#sh run int f1/0
Building configuration...

Current configuration : 107 bytes
!
interface FastEthernet1/0
 ip vrf forwarding RED
 ip address 192.168.1.1 255.255.255.0
 duplex full
end
```

- k. If you issue the command **sh ip route** this shows the routes in the global table and you will notice that you do not see 192.168.1.0/24

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

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
    2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/2] via 10.0.0.2, 00:46:52, FastEthernet0/0
    3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/3] via 10.0.0.2, 00:45:55, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.0.0.0/24 is directly connected, FastEthernet0/0
L       10.0.0.1/32 is directly connected, FastEthernet0/0
O       10.0.1.0/24 [110/2] via 10.0.0.2, 00:46:52, FastEthernet0/0
```

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

Gateway of last resort is not set

    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet1/0
L       192.168.1.1/32 is directly connected, FastEthernet1/0
```

- m. 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(config)#int f1/0
R1(config-if)#ip ospf 2 area 2
R1(config-if)#
```

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

```
R1(config-if)#
*May  3 15:28:29.795: %OSPF-5-ADJCHG: Process 2, Nbr 4.4.4.4 on FastEthernet1/0
from LOADING to FULL, Loading Done
```



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

Gateway of last resort is not set

      4.0.0.0/32 is subnetted, 1 subnets
O       4.4.4.4 [110/2] via 192.168.1.4, 00:01:48, FastEthernet1/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet1/0
L       192.168.1.1/32 is directly connected, FastEthernet1/0
R1#
*May  3 15:30:23.631: %SYS-5-CONFIG_I: Configured from console by console
```

### We now need to repeat this process for R3 & R6

- o. Router 6 will peer OSPF using process number 2 to a VRF configured on R3. 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 2

exit

int f0/0

ip add 192.168.2.6 255.255.255.0

ip ospf 2 area 2

no shut

exit

\*R3\*

int f1/0

no shut

```
ip add 192.168.2.3 255.255.255.0
```

```
exit
```

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

```
R3(config)#ip vrf RED
R3(config-vrf)#rd 4:4
R3(config-vrf)#route-target both 4:4
R3(config-vrf)#
```

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

```
R3(config)#int f1/0
R3(config-if)#ip vrf forwarding RED
```

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

```
% Interface FastEthernet1/0 IPv4 disabled and address(es) removed due to enabling VRF RED
```

- r. You just need to re-apply it

```
R3(config)#int f1/0
R3(config-if)#ip address 192.168.2.1 255.255.255.0
R3(config-if)#
```

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

```
R3#sh run int f1/0
Building configuration...

Current configuration : 107 bytes
!
interface FastEthernet1/0
 ip vrf forwarding RED
 ip address 192.168.2.1 255.255.255.0
 duplex full
end
```

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

```
R3(config)#int f1/0
R3(config-if)#ip ospf 2 area 2
R3(config-if)#
*May 3 15:44:33.079: %OSPF-5-ADJCHG: Process 2, Nbr 6.6.6.6 on FastEthernet1/0
from LOADING to FULL, Loading Done
```

Check the routes in vrf RED

```

R3#sh ip route vrf RED

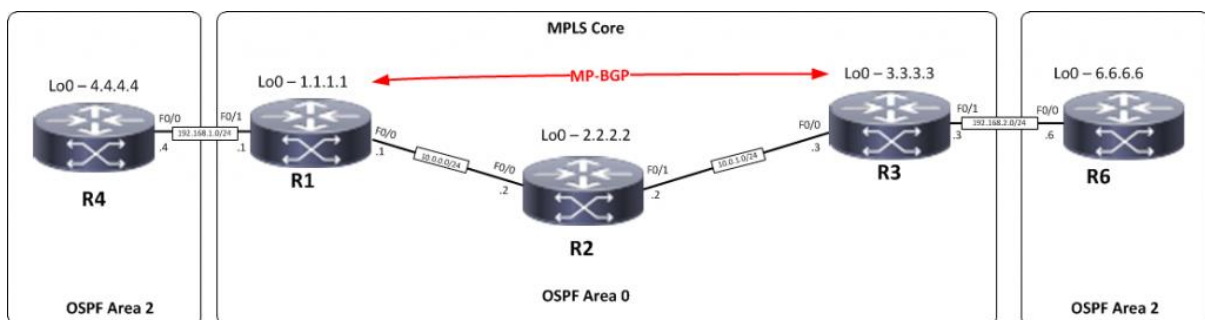
Routing Table: RED
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i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

Gateway of last resort is not set

    6.0.0.0/32 is subnetted, 1 subnets
O       6.6.6.6 [110/2] via 192.168.2.6, 00:00:54, FastEthernet1/0
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, FastEthernet1/0
L       192.168.2.1/32 is directly connected, FastEthernet1/0

```

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 peering with 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 redistribute the 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 the routes in MP-BGP in R1 and R3 back out to OSPF

## Before we start lets do some verifications

Check the routes on R4

```

    4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet0/0
L       192.168.1.4/32 is directly connected, FastEthernet0/0

```

### Check the routes on R1

```
1.0.0.0/32 is subnetted, 1 subnets
C    1.1.1.1 is directly connected, Loopback0
2.0.0.0/32 is subnetted, 1 subnets
O    2.2.2.2 [110/2] via 10.0.0.2, 01:14:36, FastEthernet0/0
3.0.0.0/32 is subnetted, 1 subnets
O    3.3.3.3 [110/3] via 10.0.0.2, 01:13:39, FastEthernet0/0
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C    10.0.0.0/24 is directly connected, FastEthernet0/0
L    10.0.0.1/32 is directly connected, FastEthernet0/0
O    10.0.1.0/24 [110/2] via 10.0.0.2, 01:14:36, FastEthernet0/0
```

- u. 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 show the 192.168.1.0/24 route as that is in VRF RED – to see that we run the following command

```
4.0.0.0/32 is subnetted, 1 subnets
O    4.4.4.4 [110/2] via 192.168.1.4, 00:24:48, FastEthernet1/0
192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.0/24 is directly connected, FastEthernet1/0
L    192.168.1.1/32 is directly connected, FastEthernet1/0
R1#
```

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(config)#router bgp 1
R1(config-router)#address-family ipv4 vrf RED
R1(config-router-af)#redistribute ospf 2
R1(config-router-af)#
```

### Redistribute OSPF into MP-BGP on R3

```
R3(config)#router bgp 1
R3(config-router)#address-family ipv4 vrf RED
R3(config-router-af)#redistribute ospf 2
R3(config-router-af)#
```

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 VRF with this command

```

R1#sh ip bgp vpnv4 vrf RED
BGP table version is 7, local router ID is 1.1.1.1
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
Route Distinguisher: 4:4 (default for vrf RED)
*> 4.4.4.4/32        192.168.1.4          2           32768 ?
*>i 6.6.6.6/32        3.3.3.3              2          100      0 ?
*> 192.168.1.0        0.0.0.0              0           32768 ?
*>i 192.168.2.0        3.3.3.3              0          100      0 ?
R1#

```

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 well with a next hop of 3.3.3.3 (which is the loopback of R3 – showing that it is going 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 7, 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, 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
Route Distinguisher: 4:4 (default for vrf RED)
*>i 4.4.4.4/32        1.1.1.1              2          100      0 ?
*> 6.6.6.6/32        192.168.2.6          2           32768 ?
*>i 192.168.1.0        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 4.4.4.4 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 MPLS and R2 is not in the picture)

- v. The final step is to get the routes that have come across the MPLS back into OSPF and then we can get end to end connectivity

```

R1(config)#router ospf 2
R1(config-router)#redistribute bgp 1 subnets
R1(config-router)#

```

```

R3(config)#router ospf 2
R3(config-router)#redistribute bgp 1 subnets
R3(config-router)#

```

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

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

```

    4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
    6.0.0.0/32 is subnetted, 1 subnets
O IA    6.6.6.6 [110/3] via 192.168.1.1, 00:02:29, FastEthernet0/0
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, FastEthernet0/0
L       192.168.1.4/32 is directly connected, FastEthernet0/0
O IA    192.168.2.0/24 [110/2] via 192.168.1.1, 00:02:29, FastEthernet0/0

```

Great we have 6.6.6.6 in there

x. **Also check the routing table on R6**

```

    4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/3] via 192.168.2.1, 00:03:06, FastEthernet0/0
    6.0.0.0/32 is subnetted, 1 subnets
C       6.6.6.6 is directly connected, Loopback0
O IA    192.168.1.0/24 [110/2] via 192.168.2.1, 00:03:06, FastEthernet0/0
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, FastEthernet0/0
L       192.168.2.6/32 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 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 120/126/140 ms

```

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
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.1.1 28 msec 20 msec 28 msec
  2 10.0.0.2 [MPLS: Labels 17/19 Exp 0] 140 msec 120 msec 116 msec
  3 192.168.2.1 [MPLS: Label 19 Exp 0] 100 msec 84 msec 92 msec
  4 192.168.2.6 120 msec 128 msec 112 msec

```

