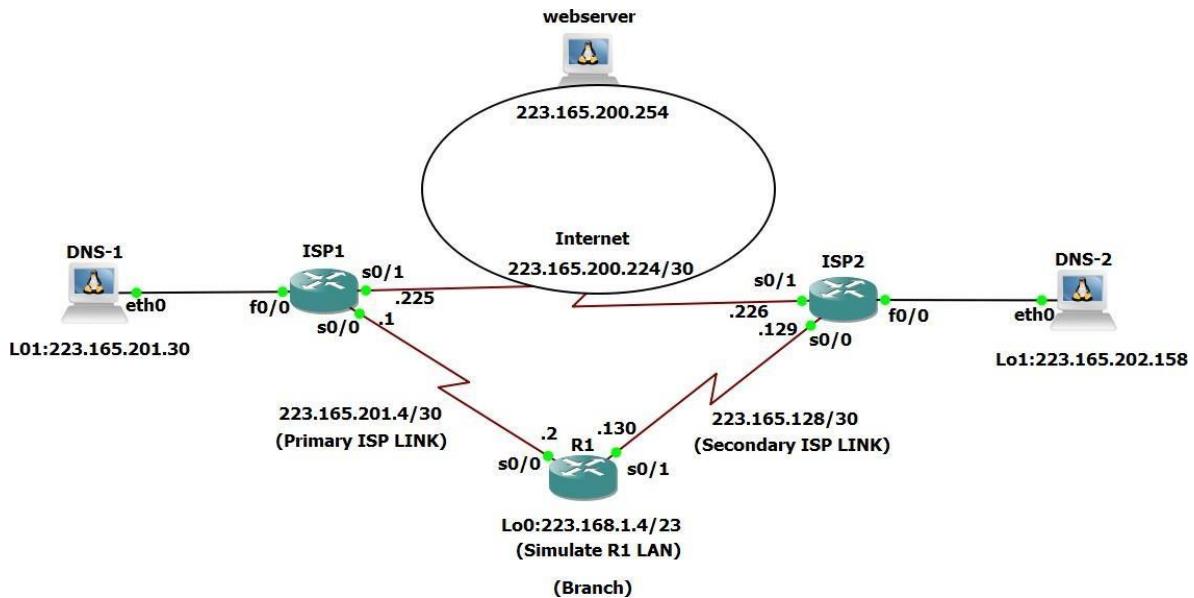


Practical No-1

Aim: Configure IP SLA Tracking and Path Control Topology

Topology:



Objectives:

- Configure and verify the IP SLA feature.
- Test the IP SLA tracking feature.
- Verify the Configuration and Operation using show and debug commands.

Step 1: Prepare the routers and Configure the Router hostname and Interface addresses.

Router R1

Interface Loopback 0

Ip address 223.168.1.4 255.255.255.0

Interface serial 0/0/0

Ip address 223.165.201.2 255.255.255.252

no shutdown

interface serial 0/0/1

ip address 223.165.202.130 255.255.255.252

```
R1#en
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#host R1
R1(config)#interface Loopback 0
R1(config-if)#
*Mar 1 00:00:47.607: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
R1(config-if)#ip address 223.168.1.4 255.255.255.0
R1(config-if)#
R1(config-if)#int s0/0
R1(config-if)#ip address 223.165.201.2 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#in
*Mar 1 00:02:54.835: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R1(config-if)#int
*Mar 1 00:02:55.839: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to up
R1(config-if)#int s0/1
R1(config-if)#ip address
*Mar 1 00:03:24.183: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to down
R1(config-if)#ip address 223.165.202.130 255.255.255.252
R1(config-if)#no shutdown
```

Router ISP1(R2)

Interface Loopback 0

Ip address 223.165.200.254 255.255.255.255

Interface Loopback 1

Ip address 223.165.201.30 255.255.255.255

Int s0/0

Ip address 223.165.201.1 255.255.255.252

no shutdown

int s0/1

ip address 223.165.200.225 255.255.255.252

no shutdown

```

ISP1#en
ISP1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP1(config)#interface Loopback 0
ISP1(config-if)#ip add
*Mar 1 00:03:11.651: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
ISP1(config-if)#ip address 223.165.200.254 255.255.255.255
ISP1(config-if)#interface Loopback 1
ISP1(config-if)#
*Mar 1 00:05:04.991: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
  changed state to up
ISP1(config-if)#ip address 223.165.201.30 255.255.255.255
ISP1(config-if)#int s0/0
ISP1(config-if)#ip address 223.165.201.1 255.255.255.252
ISP1(config-if)#no shutdown
ISP1(config-if)#
*Mar 1 00:07:06.251: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
ISP1(config-if)#
*Mar 1 00:07:07.255: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
  changed state to up
ISP1(config-if)#ints s0/1
^
% Invalid input detected at '^' marker.

ISP1(config-if)#int s0/1
ISP1(config-if)#ip address 223.165.200.225 255.255.252
% Incomplete command.

ISP1(config-if)#ip address 223.165.200.225 255.255.255.252
ISP1(config-if)#no shutdown
ISP1(config-if)#

```

ISP2 Router 3:

```

Interface Loopback 0
#
ip address 223.165.200.254 255.255.255.255
Interface Loopback 1
#
ip address 223.165.202.158 255.255.255.255

Int s0/0
Ip address 223.165.202.129 255.255.255.252
no shutdown
#
int s0/1
ip address 223.165.200.226 255.255.255.255

```

```

ISP2#en
ISP2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP2(config)#host ISP2
ISP2(config)#interface Loopback0
ISP2(config-if)#
ISP2(config-if)#
ISP2(config-if)#ip address 223.165.200.254 255.255.255.255
ISP2(config-if)#interface Loopback1
ISP2(config-if)#
ISP2(config-if)#ip address 223.165.202.158 255.255.255.255
ISP2(config-if)#int s0/0
ISP2(config-if)#ip address 223.165.202.129 255.255.255.252
ISP2(config-if)#no shutdown
ISP2(config-if)#
ISP2(config-if)#
*Mar 1 00:17:21.467: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
ISP2(config-if)#
*Mar 1 00:17:22.471: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to up
ISP2(config-if)#int s0/1
ISP2(config-if)#ip address 223.165.200.226 255.255.255.252
ISP2(config-if)#no shutdown
ISP2(config-if)#
*Mar 1 00:18:28.815: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar 1 00:18:29.823: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1,
changed state to up

```

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

R1# show interfaces description

Interface	Status	Protocol	Description
Fa0/0	admin down	down	
Se0/0	up	up	
Fa0/1	admin down	down	
Se0/1	up	up	
Se0/2	admin down	down	
Se0/3	admin down	down	
Se0/4	admin down	down	
Se0/5	admin down	down	
Fa1/0	admin down	down	
Lo0	up	up	

- 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.

Router R1 ip route 0.0.0.0 0.0.0.0 223.165.201.

```

R1(config)#ip route 0.0.0.0 0.0.0.0 223.165.201.1
R1(config)#

```

Router ISP1 (R2)

```
Router eigrp 1
network 223.165.200.224 0.0.0.3
network 223.165.201.4 0.0.0.31
no auto-summary
ip route 223.168.1.0 255.255.255.0 223.165.201.2
```

```
ISP2(config)#router eigrp 1
ISP2(config-router)#network 223.165.200.224 0.0.0.3
ISP2(config-router)#
*Mar 1 00:28:14.651: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 223.165.200.225
(Serial0/1) is up: new adjacency
ISP2(config-router)#network 223.165.202.128 0.0.0.31
ISP2(config-router)#no auto-summary
ISP2(config-router)#
ISP2(config-router)#
*Mar 1 00:29:25.351: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 223.165.200.225
(Serial0/1) is resync: summary configured
ISP2(config-router)#ip route 223.168.1.0 255.255.255.0 223.165.202.130
ISP2(config)#[
```

Router ISP2 (R3)

```
Router eigrp 1
Network 223.165.200.224 0.0.0.0
#
Network 223.165.202.128 0.0.0.31
no auto-summary
#
ip route 223.168.1.0 255.255.255.0 223.165.202.130
```

```
ISP1(config)#router eigrp 1
ISP1(config-router)#network 223.165.200.224 0.0.0.3
ISP1(config-router)#network 223.165.201.4 0.0.0.31
ISP1(config-router)#no auto-summary
ISP1(config-router)#ip route 223.168.1.0 255.255.255.0 223.165.201.2
ISP1(config)#[
```

Step 2: Verify server reachability.

- Before Implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2

DNS server to verify connectivity. You can copy the following TCL script and paste it into R1.

```
R1(tcl) # foreach address {
+>(tcl) #223.165.200.254
+>(tcl) #223.165.201.30
+>(tcl) #223.165.202.158
+>(tcl) #} {
+>(tcl) #ping $address source 223.168.1.4
+>(tcl) #}
```

```
R1#tclsh
R1(tcl)#foreach address{
wrong # args: should be "foreach varList list ?varList list ...? command"
R1(tcl)#foreach address {
+>(tcl) #223.165.200.254
+>(tcl) #223.165.201.30
+>(tcl) #223.165.202.158
+>(tcl) #} {
+>(tcl) #ping $address source 223.168.1.4
+>(tcl) #}
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 223.165.200.254, timeout is 2 seconds:
Packet sent with a source address of 223.168.1.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/44 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 223.165.201.30, timeout is 2 seconds:
Packet sent with a source address of 223.168.1.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/20/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 223.165.202.158, timeout is 2 seconds:
Packet sent with a source address of 223.168.1.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/39/68 ms
```

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

```
R1(tcl) #foreach address {
+>(tcl) #223.165.200.254
+>(tcl) #223.165.201.30
+>(tcl) #223.165.202.158
+>(tcl) #} {
+>(tcl) # trace $address source 223.168.1.4
+>(tcl) #}
```

```
R1(tcl) #foreach address {
+>(tcl) #223.165.200.254
+>(tcl) #223.165.201.30
+>(tcl) #223.165.202.158
+>(tcl) #} {
+>(tcl) #trace $address source 223.168.1.4
+>(tcl) #

Type escape sequence to abort.
Tracing the route to 223.165.200.254

 1 223.165.201.1 40 msec 40 msec 28 msec
Type escape sequence to abort.
Tracing the route to 223.165.201.30

 1 223.165.201.1 32 msec 28 msec 28 msec
Type escape sequence to abort.
Tracing the route to 223.165.202.158

 1 223.165.201.1 40 msec 36 msec 20 msec
 2 223.165.200.226 44 msec 40 msec 24 msec
R1(tcl)#
R1(tcl)#tclquit
```

Step 3: Configure IP SLA probes.

- a. Create and ICMP echo probe on R1 to the Primary DNS server on ISP1 using the ip sla command. The previous ip sla monitor command. In addition, the ICMP-echo command has replaced the type echo protocol Ip ICMP Echo command.

```
R1(config)# ip sla 11
R1(config-ip-sla)# icmp-echo 223.165.201.30
R1(config-ip-sla-echo) # frequency 10
R1(config-ip-sla-echo) #exit
```

```
R1#en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip sla 11
R1(config-ip-sla)#icmp-echo 223.165.201.30
R1(config-ip-sla-echo)#frequency 10
R1(config-ip-sla-echo)#
R1(config)#
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

```
R1(tcl)#show ip sla configuration 11
IP SLAs, Infrastructure Engine-II.
Entry number: 11
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 223.165.201.30/0.0.0.0
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Vrf Name:
Request size (ARR data portion): 28
Verify data: No
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): 4294967295
History Statistics:
  Number of history Lives kept: 0
  Number of history Buckets kept: 15
  History Filter Type: None
Enhanced History:
```

- c. Issue the show ip sla statistic command to display the number of successes, failures, and results of the latest operations.
- d. Although not actually required because IP SLA session 11 alone could provide the desired fault tolerance, create a second probe,22, to test connectivity to the second DNS server located on router ISP2. You can copy and paste the following commands on R1.
- e. Verify the new probe using the show ip sla configuration and show ip sla statistics commands.

R1# show ip sla configuration 22

```
R1#show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 223.165.202.158/0.0.0.0
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Vrf Name:
Request size (ARR data portion): 28
Verify data: No
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): 4294967295
History Statistics:
  Number of history Lives kept: 0
  Number of history Buckets kept: 15
  History Filter Type: None
Enhanced History:
```

R1# show ip sla statistics 22

```
R1#
R1#show ip sla statistics 22

Round Trip Time (RTT) for      Index 22
      Latest RTT: 40 milliseconds
Latest operation start time: *00:57:12.643 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 9
Number of failures: 0
Operation time to live: Forever
```

Step 4: Configure tracking options.

- 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 0.0.0.0 223.165.201.1

```
R1#en
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#no ip route 0.0.0.0 0.0.0.0 223.165.201.1
R1(config)#ip route 0.0.0.0 0.0.0.0 223.165.201.1 5
R1(config)#exit
```

R1(config) #ip route 0.0.0.0 0.0.0.0 223.165.201.1 5

- Verify the Routing table.

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

Gateway of last resort is 223.165.201.1 to network 0.0.0.0

C    223.168.1.0/24 is directly connected, Loopback0
     223.165.202.0/30 is subnetted, 1 subnets
C        223.165.202.128 is directly connected, Serial0/1
          223.165.201.0/30 is subnetted, 1 subnets
C            223.165.201.0 is directly connected, Serial0/0
S*   0.0.0.0/0 [5/0] via 223.165.201.1
```

R1# show ip route

- Use the trace 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) #

- Configure the floating static route that will be implemented with 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 0.0.0.0 223.165.201.1 2 track 1 command to create a floating static default route via 223.165.201.1 (ISP1). Notice that this command references the tracking object number 1, which in turn references IP SLA operation number 11.

R1# debug ip routing:

```
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 0.0.0.0 223.165.201.1 2 track 1
R1(config) #
```

R1(config)# ip route 0.0.0.0 0.0.0.0 223.165.201.1 2 track 1

```
R1(config)#track 1 rtr 11 reachability
R1(config-track)#
*Mar 1 03:28:01.267: RT: closer admin distance for 0.0.0.0, flushing 1 routes
*Mar 1 03:28:01.271: RT: NET-RED 0.0.0.0/0
*Mar 1 03:28:01.271: RT: add 0.0.0.0/0 via 223.165.201.1, static metric [2/0]
*Mar 1 03:28:01.271: RT: NET-RED 0.0.0.0/0
*Mar 1 03:28:01.271: RT: default path is now 0.0.0.0 via 223.165.201.1
*Mar 1 03:28:01.271: RT: new default network 0.0.0.0
*Mar 1 03:28:01.271: RT: NET-RED 0.0.0.0/0
R1(config-track)#
*Mar 1 03:28:06.271: RT: NET-RED 0.0.0.0/0
R1(config-track)#
*Mar 1 03:28:07.399: RT: NET-RED 0.0.0.0/0
R1(config-track)#
*Mar 1 03:29:07.399: RT: NET-RED 0.0.0.0/0
R1(config-track)#delay down 10 up
*Mar 1 03:30:07.399: RT: NET-RED 0.0.0.0/0
R1(config-track)#delay down 10 up 1
R1(config-track)##exit
```

- e. Repeat the steps for operation 22, track number 2, and assign the static route an admin distance higher than track 1 and lower than 5. On R1, copy the following configuration, which sets an admin distance of 3. Track 2 ip sla 22 reachability delay down 10 up 1 exit.

ip route 0.0.0.0 0.0.0.0 223.165.202.129 3 track 2

```
R1(config)#track 1 rtr 22 reachability
*Mar 1 03:40:07.399: RT: NET-RED 0.0.0.0/0
R1(config)#track 1 rtr 22 reachability
R1(config-track)#delay down 10 up 1
R1(config-track)##exit
R1(config)#ip route 0
*Mar 1 03:41:07.399: RT: NET-RED 0.0.0.0/0
R1(config)#ip route 0.0.0.0 0.0.0.0 223.165.201.1 2 track 1
R1(config)#ip route 0.0.0.0 0.0.0.0
*Mar 1 03:42:07.399: RT: NET-RED 0.0.0.0/0
R1(config)#ip route 0.0.0.0 0.0.0.0 223.165.202.129 3 track 2
R1(config) #
```

f. Verify the Routing table again.

R1# show ip route

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

Gateway of last resort is 223.165.201.1 to network 0.0.0.0

C    223.168.1.0/24 is directly connected, Loopback0
     223.165.202.0/30 is subnetted, 1 subnets
C          223.165.202.128 is directly connected, Serial0/1
     223.165.201.0/30 is subnetted, 1 subnets
C          223.165.201.0 is directly connected, Serial0/0
S*   0.0.0.0/0 [2/0] via 223.165.201.1
R1#
```

Step 5: Verify IP SLA operation.

The following summarizes the process:

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

ISP1(config)# interface loopback 1

ISP1(config-if) #shutdown

```
ISP1 (config)#
ISP1 (config) #interface loopback 1
ISP1 (config-if)#
ISP1 (config-if) #shutdown
ISP1 (config-if) #
```

b. Verify the routing table.

R1# show ip route

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

Gateway of last resort is 223.165.201.1 to network 0.0.0.0

C    223.168.1.0/24 is directly connected, Loopback0
     223.165.202.0/30 is subnetted, 1 subnets
C          223.165.202.128 is directly connected, Serial0/1
     223.165.201.0/30 is subnetted, 1 subnets
C          223.165.201.0 is directly connected, Serial0/0
S*   0.0.0.0/0 [2/0] via 223.165.201.1
R1#
```

c. Verify the SLA statistics.

R1# show ip sla statistics

```
R1#show ip sla statistics

Round Trip Time (RTT) for      Index 11
    Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *04:08:57.926 UTC Fri Mar 1 2002
Latest operation return code: No connection
Number of successes: 31
Number of failures: 97
Operation time to live: Forever

Round Trip Time (RTT) for      Index 22
    Latest RTT: 20 milliseconds
Latest operation start time: *04:09:02.642 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 79
Number of failures: 0
Operation time to live: Forever
```

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

R1# trace 223.165.200.254 source 223.168.1.4

e. Again Examine the IP SLA

statistics. R1# show ip sla statistics

```
R1#show ip sla statistics

Round Trip Time (RTT) for      Index 11
    Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *04:08:57.926 UTC Fri Mar 1 2002
Latest operation return code: No connection
Number of successes: 31
Number of failures: 97
Operation time to live: Forever

Round Trip Time (RTT) for      Index 22
    Latest RTT: 20 milliseconds
Latest operation start time: *04:09:02.642 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 79
Number of failures: 0
Operation time to live: Forever
```

g. Verify the Routing Table.**R1# show ip route**

```
R1#show ip route
```

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route

Gateway of last resort is 223.165.201.1 to network 0.0.0.0

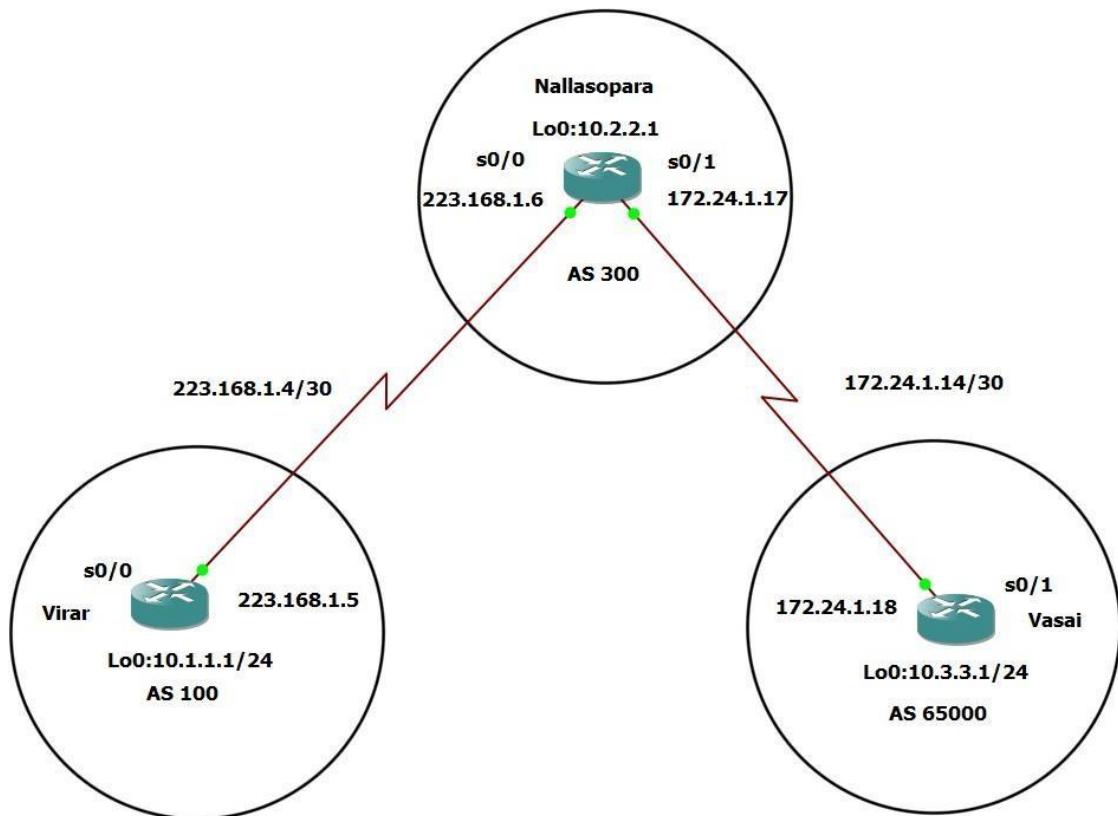
```
C    223.168.1.0/24 is directly connected, Loopback0
    223.165.202.0/30 is subnetted, 1 subnets
C        223.165.202.128 is directly connected, Serial0/1
        223.165.201.0/30 is subnetted, 1 subnets
C            223.165.201.0 is directly connected, Serial0/0
S*   0.0.0.0/0 [2/0] via 223.165.201.1
```

```
R1#
```

Practical No-2

Aim: Using AS_PATH Attribute

Topology:



Objective:

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

Step 1: Prepare the routers for the lab.

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

Step 2: Configure the hostname and interface addresses.

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

Router R1(hostname Virar)

```
Virar(config)# interface Loopback 0
Virar(config-if) #ip address 10.1.1.1 255.255.255.255
#
Virar(config-if) #exit
Virar(config-if) #int s0/0
Virar(config-if) #ip address 223.168.1.5 255.255.255.252
Virar(config-if) #no shutdown
Virar(config-if) #end
Virar#
```

```
Virar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Virar(config)#interface Loopback0
Virar(config-if)#
*Mar 1 00:01:55.115: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
Virar(config-if)#ip address 10.1.1.1 255.255.255.0
Virar(config-if)#
Virar(config)#int s0/0
Virar(config-if)#ip address 223.168.1.5 255.255.255.252
Virar(config-if)#
Virar(config-if)#
*Mar 1 00:04:18.019: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Virar(config-if)#
*Mar 1 00:04:19.023: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
  changed state to up
```

Router R2(hostname Nallasopara)

```
Nallasopara(config) #interface Loopback 0
Nallasopara(config-if) #ip address 10.2.2.1 255.255.255.0
Nallasopara(config-if) #
Nallasopara(config-if) #exit
Nallasopara(config-if) #int s0/0
Nallasopara(config-if) #
Nallasopara(config-if) #ip address 223.168.1.6 255.255.255.252
Nallasopara(config-if) #
Nallasopara(config-if) #no shutdown
Nallasopara(config-if) #exit
```

Nallasopara(config-if) #int s0/1

Nallasopara(config-if) #ip address 172.24.1.17 255.255.255.252

Nallasopara(config-if) #no shutdown

Nallasopara(config-if) #end

Nallasopara#

```
Nallasopara#
Nallasopara#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Nallasopara(config)#interface loopback 0
Nallasopara(config-if)#
*Mar 1 00:19:06.143: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
Nallasopara(config-if)#ip address 10.2.2.1 255
% Incomplete command.

Nallasopara(config-if)#ip address 10.2.2.1 255.255.255.0
Nallasopara(config-if)#exit
Nallasopara(config)#int s0/0
Nallasopara(config-if)#ip address 223.168.1.6 255.255.255.252
Nallasopara(config-if)#no shutdown
Nallasopara(config-if)#exit
*Mar 1 00:21:13.879: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Nallasopara(config-if)#exit
```

Router 3(hostname Vasai)

Vasai(config)# interface Loopback 0

Vasai(config-if) #ip address 10.3.3.1 255.255.255.0

Vasai(config-if) #exit

Vasai(config-if) #int s0/1

Vasai(config-if) #ip address 172.24.1.18 255.255.255.252

Vasai(config-if) #no shutdown

Vasai(config-if) #end

Vasai#

```
Nallasopara(config)#int s0/1
Nallasopara(config-if)#ip address 172.24.1.17 255.255.255.252
Nallasopara(config-if)#no shutdown
Nallasopara(config-if)#
*Mar 1 00:25:57.127: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
Nallasopara(config-if)#+
```

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config)#interface loopback 0
Vasai(config-if)#
*Mar 1 00:32:27.599: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
Vasai(config-if)#ip address 10.3.3.1 255.255.255.0
Vasai(config-if)#exit
Vasai(config)#int s0/1
Vasai(config-if)#ip address 172.24.1.18 255.255.252
Vasai(config-if)#no shutdown
Vasai(config-if)#
*Mar 1 00:33:59.403: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar 1 00:34:00.403: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1,
changed state to up
```

- b. Use Ping to test the Connectivity between the directly connected routers.

```
Nallasopara#ping 223.168.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 223.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/16 ms
Nallasopara#
Nallasopara#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 100 percent (5/5), round-trip min/avg/max = 1/10/48 ms
```

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.

```
Virar(config)# router bgp 100
Virar(config-router) #neighbor 223.168.1.6 remote-as 300
Virar(config-router) #network 10.1.1.0 mask 255.255.255.0
Virar(config-router) #
```

```
Virar(config)#
Virar(config) #router bgp 100
Virar(config-router) #neighbor 223.168.1.6 remote-as 300
Virar(config-router) #network 10.1.1.0 mask 255.255.255.0
Virar(config-router) #
```

```
Nallasopara(config)# router bgp 300
Nallasopara(config-router) #neighbor 223.168.1.5 remote-as 100
Nallasopara(config-router) #neighbor 172.24.1.18 remote-as 65000
Nallasopara(config-router) #10.20.2.0 mask 255.255.255.0
```

```
Nallasopara#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Nallasopara(config)#router bgp 300
Nallasopara(config-router) #neighbor 223.168.1.5 remote-as 100
Nallasopara(config-router) #
Nallasopara(config-router) #
*Mar 1 00:56:33.991: %BGP-5-ADJCHANGE: neighbor 223.168.1.5 Up
Nallasopara(config-router) #neighbor 172.24.1.18 remote-as 65000
Nallasopara(config-router) #network 10.2.2.0 mask 255.255.255.0
Nallasopara(config-router) #
```

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

```
Vasai(config) #
Vasai(config) #router bgp 65000
Vasai(config-router) #neighbor 172.24.1.17 remote-as 300
^
% Invalid input detected at '^' marker.

Vasai(config-router) #neighbor 172.24.1.17 remote-as 300
Vasai(config-router) #
*Mar 1 01:29:13.087: %BGP-5-ADJCHANGE: neighbor 172.24.1.17 Up
Vasai(config-router) #network 10.3.3.0 mask 255.255.255.0
Vasai(config-router) #
```

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

Nallasopara# show ip bgp neighbors

```
Nallasopara#
Nallasopara#show ip bgp neighbors
BGP neighbor is 172.24.1.18, remote AS 65000, external link
  BGP version 4, remote router ID 10.3.3.1
  BGP state = Established, up for 00:07:34
  Last read 00:00:34, last write 00:00:34, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
          Sent      Rcvd
  Opens:        1        1
  Notifications: 0        0
  Updates:      3        1
  Keepalives:   10       10
  Route Refresh: 0        0
  Total:        14       12
Default minimum time between advertisement runs is 30 seconds
```

Step 4: Remove the Private AS.

- a. Nallasopara the Virar routing table using the show ip route command. Virar should have a route to both 10.2.2.0 and 10.3.3.0 Troubleshoot if necessary.

Virar# Show ip route

```
Virar#
Virar#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      223.168.1.0/30 is subnetted, 1 subnets
C        223.168.1.4 is directly connected, Serial0/0
      10.0.0.0/24 is subnetted, 3 subnets
B          10.3.3.0 [20/0] via 223.168.1.6, 00:13:30
B          10.2.2.0 [20/0] via 223.168.1.6, 00:45:15
C            10.1.1.0 is directly connected, Loopback0
```

- b. Ping Again, this time as an extended ping, sourcing from the Loopback 0 interface address. Ping 10.3.3.1 source 10.1.1.1 or ping 10.3.3.1 source Lo0

```
Virar#ping 10.3.3.1 source 10.1.1.1
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 = 1/1/4 ms
Virar#
Virar#show 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
* > 10.1.1.0/24    0.0.0.0             0        32768 i
* > 10.2.2.0/24    223.168.1.6         0          300 i
* > 10.3.3.0/24    223.168.1.6         0          300  65000 i
```

- c. Now check the BGP table on Virar. The AS_PATH to the 10.3.3.0 network should be AS 300. It no longer has the private AS in the path.

Virar# show ip bgp

```
Virar#ping 10.3.3.1 source 10.1.1.1
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 = 1/1/4 ms
Virar#
Virar#show 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 10.1.1.0/24      0.0.0.0            0        32768  i
*-> 10.2.2.0/24    223.168.1.6          0        0 300  i
*-> 10.3.3.0/24    223.168.1.6          0        0 300 65000  i
```

Step 5: Use the AS_PATH attribute to filter routes.

- 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 Nallasopara.

Nallasopara(config) #ip as_path access-list 1 deny ^100\$

Nallasopara(config) #ip as-path access-list 1 permit .*

```
Nallasopara#
Nallasopara#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Nallasopara(config)#ip as-path access-list 1 deny ^100$
Nallasopara(config)#
Nallasopara(config)#ip as-path access-list 1 permit .*
```

- Apply the Configured access list using the neighbor command with the filter-list option.

Nallasopara(config) #router bgp 300

Nallasopara(config-router) #neighbor 223.168.1.5 remove-private-as

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

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

Virar# show ip route

```
Virar#
Virar#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      223.168.1.0/30 is subnetted, 1 subnets
C        223.168.1.4 is directly connected, Serial0/0
      10.0.0.0/24 is subnetted, 3 subnets
B          10.3.3.0 [20/0] via 223.168.1.6, 00:13:30
B          10.2.2.0 [20/0] via 223.168.1.6, 00:45:15
C        10.1.1.0 is directly connected, Loopback0
```

- d. Return to Nallasopara and Verify that the filter is working as intended.

Nallasopara# show ip bgp regexp ^100\$

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

      Network          Next Hop           Metric LocPrf Weight Path
*-> 10.1.1.0/24    223.168.1.5        0          0 100 i
```

- e. Run the following TCL scripts on all routers to verify whether there is connectivity.
All pings from Nallasopara should be successful. Virar should not be able to ping the

Vasai loopback 10.3.3.1 or The WAN link 172.24.1.6/30. Vasai should not be able to ping Virar Loopback 10.1.1.1 or the WAN link 223.168.1.4/30.

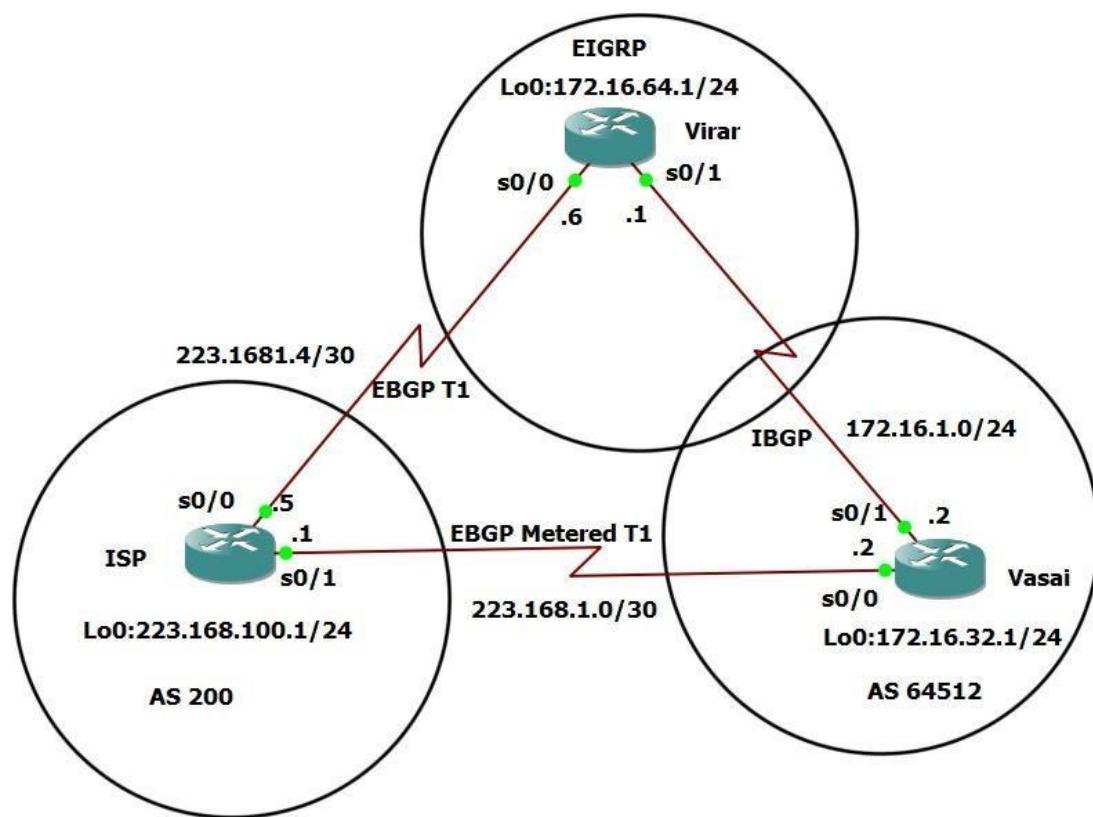
```
Nallasopara(tcl)#foreach address {
+>10.1.1.1
+>10.2.2.1
+>10.3.3.1
+>223.168.1.5
+>223.168.1.6
+>172.24.1.17
+>172.24.1.18
+>} { ping $address }
```

```
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 = 1/1/1 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 = 1/1/1 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 223.168.1.5, 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 223.168.1.6, timeout is 2 seconds:  
!!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/16 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 = 1/3/12 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 = 1/1/4 ms
```

Practical No – 3

Aim: Configure IBGP and EBGP Sessions, Local Preference, and MED.

Topology:



Objectives:

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

Step 1: Configure interface addresses.

Router R1(hostname ISP)

```

ISP(config)# interface Loopback 0
ISP(config-if) #ip address 223.168.100.1 255.255.255.0
    
```

```

ISP(config-if) #exit
ISP(config) #interface Serial 0/0/0
ISP(config-if) #ip address 223.168.1.5 255.255.255.252
ISP(config-if) #exit
ISP(config) #interface Serial 0/0/1
ISP(config-if) #ip address 223.168.1.1 255.255.255.252
ISP(config-if) #no shutdown
ISP(config-if) #end

```

```

ISP#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP(config)#interface Loopback 0
ISP(config-if)#ip addr
*Mar 1 00:01:29.551: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
ISP(config-if)#ip address 223.168.100.1 255.255.255.0
ISP(config-if)#exit
ISP(config)#
ISP(config)#int s0/0
ISP(config-if)#ip address 223.168.1.5 255.255.255.252
ISP(config-if)#no shutdown
ISP(config-if)#exit
ISP(config)#
ISP(config)#
*Mar 1 00:02:40.175: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
ISP(config)#
*Mar 1 00:02:41.179: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to up
ISP(config)#int s0/1
ISP(config-if)#ip address 223.168.1.1
*Mar 1 00:03:03.143: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to down
ISP(config-if)#ip address 223.168.1.1 255.255.255.252
ISP(config-if)#no shutdown

```

Router R2(hostname Virar)

```

Virar(config) #interface Loopback 0
Virar(config-if) #ip address 172.16.64.1 255.255.255.0
Virar(config) #exit
Virar(config) #interface Serial 0/0/0
Virar(config-if) #ip address 223.168.1.6 255.255.255.252
Virar(config-if) #no shutdown
Virar(config-if) #exit

```

```

Virar(config) interface Serial 0/0/1
Virar(config-if) #ip address 172.16.1.1 255.255.255.0
Virar(config-if) #no shutdown
Virar(config-if) #end

```

```

Virar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Virar(config)#interface Loopback 0
Virar(config-if)#
*Mar 1 00:03:08.551: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
Virar(config-if) #ip address 172.16.64.1 255.255.255.0
Virar(config-if) #exit
Virar(config)#
Virar(config)#int s0/0
Virar(config-if) #ip address 223.168.1.6 255.255.255.252
Virar(config-if) #no shutdown
Virar(config-if) #exit
*Mar 1 00:04:55.299: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Virar(config-if) #exit
Virar(config)#
Virar(config)#
*Mar 1 00:04:56.303: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
  changed state to up
Virar(config)#int s0/1
Virar(config-if) #ip address 172.16.1.1 255.255.255.0
Virar(config-if) #no shutdown

```

Router R3(hostname Vasai)

```

Vasai(config) #interface Loopback 0
Vasai(config-if) #ip address 172.16.32.1 255.255.255.0
Vasai(config-if) #exit
Vasai(config) #interface Serial 0/0/0
Vasai(config-if) #ip address 223.168.1.2 255.255.255.252
Vasai(config-if) #no shutdown
Vasai(config-if) #exit
Vasai(config) #interface Serial 0/0/1
Vasai(config-if) #ip address 172.16.1.2 255.255.255.0
Vasai(config-if) #no shutdown
Vasai(config-if) #end

```

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config)#interface Loopback 0
Vasai(config-if)#
Vasai(config-if)#
*Mar 1 00:05:19.075: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
Vasai(config-if)#ip address 172.16.32.1 255.255.255.0
Vasai(config-if)#exit
Vasai(config)#
Vasai(config)#int s0/0
Vasai(config-if)#ip address 223.168.1.2 255.255.255.252
Vasai(config-if)#no shutdown
Vasai(config-if)#exit
Vasai(config)#
*Mar 1 00:06:34.007: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Vasai(config)#
*Mar 1 00:06:35.011: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to up
Vasai(config)#int s0/1
Vasai(config-if)#ip address 172.16.1.2 255.255.255.0
Vasai(config-if)#no shutdown
```

Step 2: Configure EIGRP.

Configure EIGRP between the Virar and Vasai routers. (Note: if using an IOS prior to 15.0, use the no auto-summary router configuration command to disable automatic summary this command is the default beginning with IOS 15)

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

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

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

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

Step 3: Configure IBGP and Verify BGP neighbors.

- Configure IBGP between the Virar and Vasai routers. On the Virar router, enter the following configuration.

```
Virar(config) #router bgp 64512
Virar(config-router) #neighbor 172.16.32.1 remote-as 64512
```

Virar(config-router) #neighbor 172.16.32.1 update-source Lo0

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

If multiple pathways to the BGP neighbor exist, the router can use multiple IP interfaces to communicate with the neighbor. The source IP address therefore depends on the outgoing interface. The update-source Lo0 command instructs the router to use the IP address of the interface Loopback 0 as the source IP address for all BGP messages sent to that neighbor.

- b. Complete the IBGP configuration on Vasai using the Following commands.

Vasai(config) #router bgp 64512

Vasai(config-router) # neighbor 172.16.64.1 remote-as 64512

Vasai(config-router) #neighbor 172.16.64.1 update-source Lo0

```
Vasai(config)#router bgp 64512
Vasai(config-router)#neighbor 172.16.64.1 remote-as 64512
Vasai(config-router)#neighbor
*Mar 1 00:19:55.075: %BGP-5-ADJCHANGE: neighbor 172.16.64.1 Up
Vasai(config-router)#neighbor 172.16.64.1 update-source lo0
Vasai(config-router)#[
```

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

```
Vasai#show ip bgp neighbors
BGP neighbor is 172.16.64.1, remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.64.1
  BGP state = Established, up for 00:03:37
  Last read 00:00:36, last write 00:00:36, hold time is 180, keepalive interval
  is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
          Sent          Rcvd
  Opens:           1           1
  Notifications:  0           0
  Updates:        0           0
  Keepalives:     5           5
  Route Refresh:  0           0
  Total:          6           6
```

Step 4: Configure EBGP and Verify BGP neighbors.

- d. Configure ISP to run EBGP with Virar and Vasai. Enter the following commands on ISP.

```
ISP (config) #router bgp 200
ISP (config-router) #neighbor 223.168.1.6 remote-as 64512
ISP (config-router) #neighbor 223.168.1.2 remote-as 64512
ISP (config-router) #network 223.168.100.0
```

```
ISP(config)#
ISP(config) #router bgp 200
ISP(config-router) #neighbor 223.168.1.6 remote-as 64512
ISP(config-router) #neighbor 223.168.1.2 remote-as 64512
ISP(config-router) #network 223.168.100.0
ISP(config-router) #
```

- e. 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.

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

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

- f. Configure Virar as an EBGP peer to ISP

```
Virar(config) #router bgp 64512
Virar(config-router) #neighbor 223.168.1.5 remote-as 200
Virar(config-router) #network 172.16.0.0
```

```
Virar(config) #router bgp 64512
Virar(config-router) #neighbor 223.168.1.5 remote-as 200
Virar(config-router) #network 172.16.0.0
*Mar 1 00:39:55.947: %BGP-5-ADJCHANGE: neighbor 223.168.1.5 Up
Virar(config-router) #network 172.16.0.0
Virar(config-router) #exit
```

- g. Use the show ip bgp neighbors command to verify that Virar and ISP have reached the established state. Troubleshoot if necessary.

Virar# show ip bgp neighbors

```
Virar#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 00:21:28
  Last read 00:00:27, last write 00:00:27, hold time is 180, keepalive interval
is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
          Sent      Rcvd
  Opens:          1          1
  Notifications: 0          0
  Updates:        2          0
  Keepalives:     23         23
  Route Refresh: 0          0
  Total:         26         24
```

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

```
Vasai(config) #ip route 172.16.0.0 255.255.0.0 null0
Vasai(config) #router bgp 64512
Vasai(config-router) #neighbor 223.168.1.1 remote-as 200
Vasai(config-router) #network 172.16.0.0
```

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config)#ip route 172.16.0.0 255.255.0.0 null0
Vasai(config)#router bgp 64512
Vasai(config-router)#neighbor 223.168.1.1 remote-as 200
Vasai(config-router)#network
*Mar 1 00:47:00.451: %BGP-5-ADJCHANGE: neighbor 223.168.1.1 Up
Vasai(config-router)#network 172.16.0.0
```

Step 5: View BGP summary output.

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

Vasai# show ip bgp summary

```
Vasai#show ip bgp summary
BGP router identifier 172.16.32.1, local AS number 64512
BGP table version is 5, main routing table version 5
2 network entries using 240 bytes of memory
4 path entries using 208 bytes of memory
5/2 BGP path/bestpath attribute entries using 620 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 2 (at peak 3) using 64 bytes of memory
BGP using 1156 total bytes of memory
BGP activity 2/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor      V   AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.64.1    4  64512     34      34       5     0    0 00:29:29          2
223.168.1.1    4   200       8       8       5     0    0 00:02:24          1
Vasai#
```

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 conversation to re-establish with each Virar router.

ISP# clear ip bgp *

```
ISP#clear ip bgp *
ISP#
*Mar  1 00:53:35.899: %BGP-5-ADJCHANGE: neighbor 223.168.1.2 Down User reset
*Mar  1 00:53:35.903: %BGP-5-ADJCHANGE: neighbor 223.168.1.6 Down User reset
*Mar  1 00:53:36.435: %BGP-5-ADJCHANGE: neighbor 223.168.1.6 Up
*Mar  1 00:53:36.791: %BGP-5-ADJCHANGE: neighbor 223.168.1.2 Up
ISP#
```

- g. Test whether ISP can ping the Loopback 0 address of 172.16.64.1 on Virar and the serial link between Virar and Vasai, 172.16.1.1

ISP# ping 172.16.64.1

```
ISP#
ISP#ping 172.16.64.1

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

ISP# ping 172.16.1.1

```
ISP#
ISP#ping 172.16.1.1

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

- h. Now ping from ISP to the Loopback 0 address of 172.16.32.1 on Vasai and the Serial link between Virar and Vasai, 172.16.1.2.

ISP# ping 172.16.32.1

```
ISP#ping 172.16.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms
```

ISP# ping 172.16.1.2

```
ISP#ping 172.16.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/16 ms
ISP#
```

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

ISP# show ip bgp

```
ISP#show ip bgp
BGP table version is 11, local router ID is 223.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0        223.168.1.2          0          0 64512 i
*>  223.168.1.6        223.168.1.6          0          0 64512 i
*>  223.168.100.0     0.0.0.0           0          32768 i
```

- i. At this point, the ISP router should be able to get to each network connected to Virar and Vasai from the Loopback address 223.168.100.1. Use the extended ping command and specify the source address of ISP Lo0 to test.

ISP# ping 172.16.1.1 source 223.168.100.1

```
ISP#ping 172.16.1.1 source 223.168.100.1
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 223.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/16 ms
ISP#
```

ISP# ping 172.16.32.1 source 223.168.100.1

```
ISP#ping 172.16.32.1 source 223.168.100.1
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 223.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/12 ms
ISP#
```

ISP# ping 172.16.1.2 source 223.168.100.1

```
ISP#ping 172.16.1.2 source 223.168.100.1
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 223.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
ISP#
```

ISP# ping 172.16.64.1 source 223.168.100.1

```
ISP#ping 172.16.64.1 source 223.168.100.1
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 223.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/12 ms
ISP#
```

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

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

```
ISP (config) # router bgp 200
ISP (config-router) # network 223.168.1.0 mask 255.255.255.252
ISP (config-router) # network 223.168.1.4 mask 255.255.255.252
```

```
ISP#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#network 223.168.1.0 mask 255.255.255.252
ISP(config-router)#network 223.168.1.4 mask 255.255.255.252
ISP(config-router)#exit
```

- k. Issue the show ip bgp command to verify that ISP is Correctly injecting its own WAN links int BGP.

ISP# show ip bgp

```
ISP#show ip bgp
BGP table version is 5, local router ID is 223.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 172.16.0.0      223.168.1.2        0          0 64512 i
*          223.168.1.6        0          0 64512 i
*> 223.168.1.0/30   0.0.0.0          0          32768 i
*> 223.168.1.4/30   0.0.0.0          0          32768 i
*> 223.168.100.0    0.0.0.0          0          32768 i
```

- I. Verify on Virar and Vasai that opposite WAN link is included in the routing table.
The output from Vasai is as follows.

Vasai# show ip route

```
Vasai#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial0/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 01:24:13, Serial0/1
```

- m. 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 Vasai. The only possible path from to ISP's 223.168.100.0/24 through Virar.

ISP (config) #router bgp 200

ISP (config-router) #no network 223.168.1.0 mask 255.255.255.252

ISP (config-router) #no network 223.168.1.4 mask 255.255.255.252

ISP (config-router) #exit

ISP (config) #interface serial 0/0/1

ISP (config-if) #shutdown

```

ISP#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#no network 223.168.1.0 mask 255.255.255.252
ISP(config-router)#no network 223.168.1.4 mask 255.255.255.252
ISP(config-router)#exit
ISP(config)#int s0/1
ISP(config-if)#shutdown
ISP(config-if)#
*Mar 1 01:34:41.871: %BGP-5-ADJCHANGE: neighbor 223.168.1.2 Down Interface flap
ISP(config-if)#
*Mar 1 01:34:43.863: %LINK-5-CHANGED: Interface Serial0/1, changed state to adm
inistratively down
*Mar 1 01:34:44.863: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1,
changed state to down

```

- n. Display Vasai BGP table using the **show ip bgp** command and the IPv4 routing table with **show ip route**.

Vasai# show ip bgp

```

Vasai#show ip bgp
BGP table version is 15, 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 172.16.0.0      0.0.0.0              0        32768 i
*   i                 172.16.64.1          0       100      0 i
*   i 223.168.100.0  223.168.1.5          0       100      0 200 i

```

Vasai# show ip route

```

Vasai#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    223.168.100.0/24 [20/0] via 223.168.1.1, 00:34:46
     223.168.1.0/30 is subnetted, 2 subnets
B      223.168.1.4 [20/0] via 223.168.1.1, 00:04:58
C      223.168.1.0 is directly connected, Serial0/0
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.16.32.0/24 is directly connected, Loopback0
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial0/1
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 01:14:18, Serial0/1

```

Virar(config) #router bgp 64512

Virar(config-router) #neighbor 172.16.32.1 next-hop-self

```
Virar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Virar(config)#router bgp 64512
Virar(config-router)#neighbor 172.16.32.1 next-hop-self
Virar(config-router)#exit
```

Vasai(config) # router bgp 64512

Vasai(config-router) #neighbor 172.16.64.1 next-hop-self

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config)#router bgp 64512
Vasai(config-router)#
Vasai(config-router)#neighbor 172.16.64.1 next-hop-self
Vasai(config-router)#[
```

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

Virar# clear ip bgp *

```
Virar#clear ip bgp *
Virar#
*Mar  1 01:56:41.131: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Down User reset
*Mar  1 01:56:41.131: %BGP-5-ADJCHANGE: neighbor 223.168.1.5 Down User reset
*Mar  1 01:56:41.199: %BGP-5-ADJCHANGE: neighbor 172.16.32.1 Up
*Mar  1 01:56:42.023: %BGP-5-ADJCHANGE: neighbor 223.168.1.5 Up
Virar#
```

Vasai# clear ip bgp *

```
Vasai#clear ip bgp *
Vasai#
*Mar  1 01:54:37.299: %BGP-5-ADJCHANGE: neighbor 172.16.64.1 Down User reset
*Mar  1 01:54:37.835: %BGP-5-ADJCHANGE: neighbor 172.16.64.1 Up
Vasai#[
```

- p. After the routers have returned to established BGP Speakers, issues the **show ip bgp** command on Vasai and notice that the next hop is now Virar instead of ISP.

```
Vasai#show ip bgp
BGP table version is 15, 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      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
* i223.168.100.0   223.168.1.5          0       100      0 200 i
```

- q. The **show ip route** command on Vasai now displays the 223.168.100.0/24 network because Virar is the next hop, 172.16.64.1, which is reachable from Vasai.

Vasai# **show ip route**

```
Vasai#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    223.168.100.0/24 [200/0] via 172.16.64.1, 00:03:40
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C    172.16.32.0/24 is directly connected, Loopback0
S    172.16.0.0/16 is directly connected, Null0
C    172.16.1.0/24 is directly connected, Serial0/1
D    172.16.64.0/24 [90/2297856] via 172.16.1.1, 01:46:56, Serial0/1
```

- r. Before configuring the next BGP attribute, restore the WAN link between ISP and Vasai. This will change the BGP table and routing table on both routers. For example, Vasai routing table shows 223.168.100.0/24 will now have a better path through ISP.

ISP(config)# interface serial 0/0/1

ISP (config-if) #no shutdown

Vasai# show ip route

```
Vasai#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 223.168.1.1 to network 0.0.0.0

B    223.168.100.0/24 [20/0] via 223.168.1.1, 00:21:28
      223.168.1.0/30 is subnetted, 1 subnets
C      223.168.1.0 is directly connected, Serial0/0
      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C          172.16.32.0/24 is directly connected, Loopback0
S          172.16.0.0/16 is directly connected, Null0
C          172.16.1.0/24 is directly connected, Serial0/1
D          172.16.64.0/24 [90/2297856] via 172.16.1.1, 03:40:49, Serial0/1
B*   0.0.0.0/0 [20/0] via 223.168.1.1, 00:21:30
```

Step 8: Set BGP local Preference.

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

Virar(config) # route-map PRIMARY_T1_IN permit 10

Virar(config-route-map) # set local-preference 150

Virar(config-route-map) #exit

Virar(config) # router bgp 64512

Virar(config-router) #neighbor 223.168.1.5 route-map PRIMARY_T1_IN in

```
Virar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Virar(config)#route-map PRIMARY_T1_IN permit 10
Virar(config-route-map)#set local-preference 150
Virar(config-route-map)#exit
Virar(config)#router bgp 64512
Virar(config-router)#neighbor 223.168.1.5 route-map PRIMARY_T1_IN in
Virar(config-router) #
```

```
Vasai(config) # route-map SECONDARY_T1_IN permit 10
Vasai(config-route-map) # set local-preference 125
Vasai(config-router-map) #exit
Vasai(config) # router bgp 64512
Vasai(config-router) #neighbor 223.168.1.1 route-map SECONDARY_T1_IN in
```

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config) #route-map SECONDARY_T1_IN permit 10
Vasai(config-route-map) #set local-preference 125
Vasai(config-route-map) #exit
Vasai(config) #router bgp 64512
Vasai(config-router) #neighbor 223.168.1.1 route-map SECONDARY_T1_IN in
Vasai(config-router) #
```

- t. Use the clear ip bgp * soft command after configuring this new policy. When the Conversations have been re-established, issue the show ip bgp command on Virar and Vasai.

Virar# clear ip bgp * soft

```
Virar#
Virar#clear ip bgp * soft
Virar#
```

Vasai# clear ip bgp * soft

```
Vasai#
*Mar 1 02:20:39.671: %SYS-5-CONFIG_I: Configured from console by console
Vasai#clear ip bgp * soft
Vasai#
```

Virar# show ip bgp

```
Virar#show 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0        172.16.32.1        0       100      0 i
*>                 0.0.0.0             0           32768 i
*> 223.168.100.0    223.168.1.5        0       150      0 200 i
```

Vasai# show ip bgp

```
Vasai#show ip bgp
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      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
*   223.168.100.0    223.168.1.1          0       125      0 200 i
*>i                 172.16.64.1          0       150      0 200 i
```

Step 9: BGP MED.

- u.** In this previous step we saw that Virar and Vasai will route traffic for 223.168.100.0/24 using the link between Virar and ISP. Examine what the return path ISP takes to reach AS 64512. Notice that the path is different from the original path. This is known as asymmetric routing and is not necessarily an unwanted trait.

ISP# show ip bgp

```
ISP#show ip bgp
BGP table version is 11, local router ID is 223.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*   172.16.0.0      223.168.1.2          0        0 64512 i
*->                    223.168.1.6          0        0 64512 i
*> 223.168.100.0    0.0.0.0              0        32768  i
```

ISP# show ip route.

```
ISP#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    223.168.100.0/24 is directly connected, Loopback0
      223.168.1.0/30 is subnetted, 2 subnets
C        223.168.1.4 is directly connected, Serial0/0
C        223.168.1.0 is directly connected, Serial0/1
B    172.16.0.0/16 [20/0] via 223.168.1.6, 00:34:06
```

- a.** Use an extended ping command to verify this situation. Specify the record option and compare your output to following. Notice the return path using the exit interface 223.168.1.1 to Vasai.

```
Vasai#
Vasai#ping
Protocol [ip]:
Target IP address: 223.168.100.1
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 223.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options: Total option bytes= 39, padded length=40
Record route: <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
Reply to request 4 (1 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(223.168.1.6)
(223.168.100.1)
(223.168.1.5)
(172.16.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/24 ms
```

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

1. A ping that is sourced from 172.16.32.1 exits Vasai through s0/0/1, 172.16.1.2. then it at the s0/0/1 interface for Virar.

2. Virar s0/0/0, 223.168.1.6, routes the packet out to arrive at the s0/0/0 interface of ISP.
3. The target of 223.168.100.1 is reached: 223.168.100.1.
4. The packet is next forwarded out the s0/0/1, s0/0/1, 223.168.1.1 interface for ISP and arrives at the s0/0/0 interface for Vasai.
5. Vasai then forwards the packet out the last interface, loopback 0, 172.16.32.1.

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

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

```
Virar(config) #route-map PRIMARY_T1_MED_OUT permit 10
```

```
Virar(config-route-map) #set Metric 50
```

```
Virar(config-router-map) # exit
```

```
Virar(config) #router bgp 64512
```

```
Virar(config-router) #neighbor 223.168.1.5 route-map PRIMARY_T1_MED_OUT out
```

```
Virar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Virar(config) #route-map PRIMARY_T1_MED_OUT permit 10
Virar(config-route-map) #set Metric 50
Virar(config-route-map) #exit
Virar(config) #router bgp 64512
Virar(config-router) #neighbor 223.168.1.5 route-map PRIMARY_T1_MED_OUT out
```

```
Vasai(config) #route-map SECONDARY_T1_MED_OUT permit 10
```

```
Vasai(config-route-map) # set metric 75
```

```
Vasai(config-route-map) # exit
```

```
Vasai(config) #router bgp 64512
```

```
Vasai(config-router) #neighbor 223.168.1.1 route-map SECONDARY_T1_MED_OUT
out
```

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config) #route-map SECONDARY_T1_MED_OUT permit 10
Vasai(config-route-map) #set Metric 75
Vasai(config-route-map) #exit
Vasai(config) #router bgp 64512
Vasai(config-router) #neighbor 223.168.1.1 route-map SECONDARY_T1_MED_OUT out
% Specify remote-as or peer-group commands first
Vasai(config-router) #
```

- v. Use the clear ip bgp * soft command after issuing this new policy. Issuing the show ip bgp command as follows on Virar or Vasai does not indicate anything about this newly defined policy.

Virar# show ip bgp * soft

```
Virar#
Virar#clear ip bgp * soft
Virar#
```

Vasai# show ip bgp * soft

```
Vasai#
*Mar  1 02:20:39.671: %SYS-5-CONFIG_I: Configured from console by console
Vasai#clear ip bgp * soft
Vasai#
```

Virar# show ip bgp

```
Virar#show 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.32.1        0      100      0 i
*>                 0.0.0.0             0          32768 i
*> 223.168.100.0   223.168.1.5        0      150      0 200 i
```

Vasai# show ip bgp

```
Vasai#show ip bgp
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      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
* 223.168.100.0   223.168.1.1         0      125      0 200 i
*>i
```

Reissue an extended **ping** command with the record command. Notice the change in return path using the exit interface 223.168.1.5 to Virar.

```
Vasai#ping
Protocol [ip]:
Target IP address: 223.168.100.1
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 223.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options: Total option bytes= 39, padded length=40
Record route: <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
Reply to request 4 (1 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(223.168.1.6)
(223.168.100.1)
(223.168.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/8/24 ms
```

ISP# show ip bgp

```
ISP#show ip bgp
BGP table version is 12, local router ID is 223.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
*-> 172.16.0.0      223.168.1.2        0          0 64512 i
*          223.168.1.6        50          0 64512 i
*> 223.168.100.0    0.0.0.0          0          32768 i
```

Step 10: Establish a default route.

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

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

```
ISP (config)# router bgp 200
ISP (config-router) #neighbor 223.168.1.6 default-originate
ISP (config-router) #neighbor 223.168.1.2 default-originate
ISP (config-router) # exit
ISP (config) # interface Loopback 10
ISP (config-if) #ip address 10.0.0.1 255.255.255.0
```

```
ISP#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router) #neighbor 223.168.1.6 default-originate
                  ^
% Invalid input detected at '^' marker.

ISP(config-router) #neighbor 223.168.1.6 default-originate
ISP(config-router) #neighbor 223.168.1.2 default-originate
ISP(config-router) #exit
```

```
ISP(config)#interface loopback 10
ISP(config-if)#i
*Mar  1 03:13:58.167: %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)#+exit
```

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

Virar# show ip route

```
Virar#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

B    223.168.100.0/24 [200/0] via 172.16.32.1, 00:02:02
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D    172.16.32.0/24 [90/2297856] via 172.16.1.2, 03:22:07, Serial0/1
S    172.16.0.0/16 is directly connected, Null0
C    172.16.1.0/24 is directly connected, Serial0/1
C    172.16.64.0/24 is directly connected, Loopback0
B*   0.0.0.0/0 [200/0] via 172.16.32.1, 00:02:02
```

Vasai# show ip route

```
Vasai#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 223.168.1.1 to network 0.0.0.0

B    223.168.100.0/24 [20/0] via 223.168.1.1, 00:21:28
    223.168.1.0/30 is subnetted, 1 subnets
C    223.168.1.0 is directly connected, Serial0/0
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C    172.16.32.0/24 is directly connected, Loopback0
S    172.16.0.0/16 is directly connected, Null0
C    172.16.1.0/24 is directly connected, Serial0/1
D    172.16.64.0/24 [90/2297856] via 172.16.1.1, 03:40:49, Serial0/1
B*   0.0.0.0/0 [20/0] via 223.168.1.1, 00:21:30
```

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

Vasai# show ip bgp

```
Vasai#show ip bgp
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      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
*   223.168.100.0    223.168.1.1      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 Virar.

Vasai# traceroute 10.0.0.1

```
Vasai#traceroute 10.0.0.1

Type escape sequence to abort.
Tracing the route to 10.0.0.1

 1 172.16.1.1 16 msec 0 msec 0 msec
 2 223.168.1.5 [AS 200] 0 msec 12 msec 0 msec
```

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

ISP(config)# interface serial 0/0/0

```
ISP(config)#int s0/0
ISP(config-if)#shutdown
ISP(config-if)#+
```

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

Virar# show ip route

```
Virar#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

B    223.168.100.0/24 [200/0] via 172.16.32.1, 00:02:02
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D    172.16.32.0/24 [90/2297856] via 172.16.1.2, 03:22:07, Serial0/1
S    172.16.0.0/16 is directly connected, Null0
C    172.16.1.0/24 is directly connected, Serial0/1
C    172.16.64.0/24 is directly connected, Loopback0
B*   0.0.0.0/0 [200/0] via 172.16.32.1, 00:02:02
```

Vasai# show ip route

```
Vasai#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 223.168.1.1 to network 0.0.0.0

B    223.168.100.0/24 [20/0] via 223.168.1.1, 00:21:28
    223.168.1.0/30 is subnetted, 1 subnets
C    223.168.1.0 is directly connected, Serial0/0
    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C    172.16.32.0/24 is directly connected, Loopback0
S    172.16.0.0/16 is directly connected, Null0
C    172.16.1.0/24 is directly connected, Serial0/1
D    172.16.64.0/24 [90/2297856] via 172.16.1.1, 03:40:49, Serial0/1
B*   0.0.0.0/0 [20/0] via 223.168.1.1, 00:21:30
```

- g. Verify the new path using the traceroute command to 10.0.0.1 from Virar. Notice default route is now through Vasai.

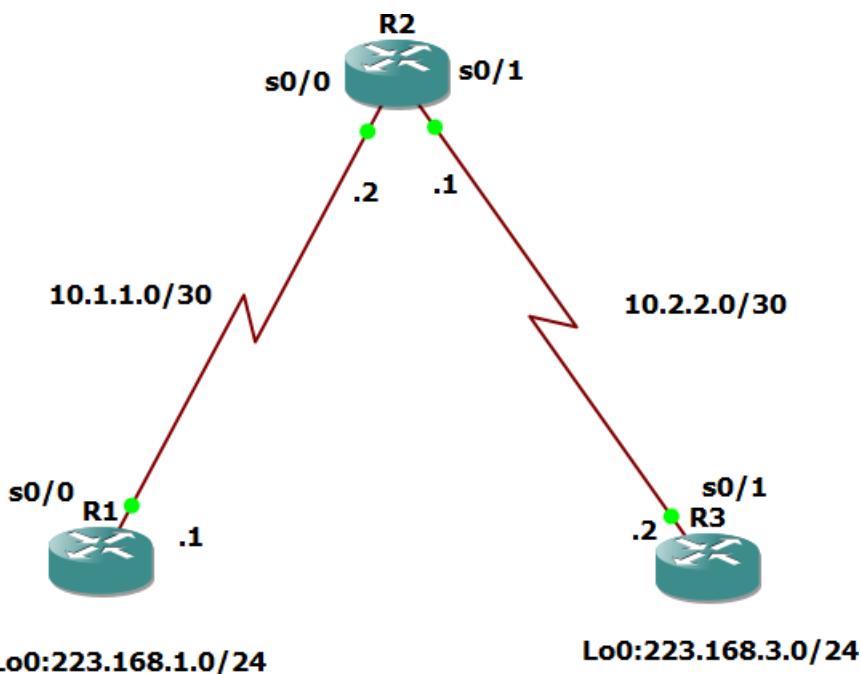
Vasai# trace 10.0.0.1

```
Vasai#traceroute 10.0.0.1
Type escape sequence to abort.
Tracing the route to 10.0.0.1
 1 223.168.1.1 [AS 200] 16 msec 12 msec 0 msec
```

Practical No 4

Aim: Secure Management Plane

Topology:



Objectives:

Secure management access.

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

Step 1: Configure loopbacks and assign addresses.

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

Router 1

```

interface Loopback 0
ip address 223.168.1.1 255.255.255.0
exit
interface Serial0/0/0
ip address 10.1.1.1 255.255.255.252
no shutdown
exit
end

```

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface loopback 0
R1(config-if)#
*Mar 1 00:02:25.839: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
R1(config-if)#ip address 223.168.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#int s0/0
R1(config-if)#ip address 10.1.1.1 255.255.255.252
R1(config-if)#no shutdown
R1(config-if)#exit

```

Router R2

```

interface Serial0/0/0
ip address 10.1.1.2 255.255.255.252
no shutdown
exit interface Serial0/0/1
ip address 10.2.2.1 255.255.255.252
no shutdown
exit
end

```

```

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int s0/0
R2(config-if)#ip address 10.1.1.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit

```

Router R3

```

interface Loopback0
ip address 223.168.3.1 255.255.255.0
exit
interface Serial0/0/1
ip address 10.2.2.2 255.255.255.252
no shutdown
exit
end

```

```

R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#interface Loopback 0
R3(config-if)#ip
*Mar 1 00:05:29.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
R3(config-if)#ip address 223.168.3.1 255.255.255.0
R3(config-if)#exit
R3(config)#
R3(config)#interface s0/1
R3(config-if)#ip address 10.2.2.2 255.255.252
% Incomplete command.

R3(config-if)#ip address 10.2.2.2 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#exit

```

Step 2: Configure static routes.

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

```

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 10.1.1.2

```

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

```

R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#ip route 0.0.0.0 0.0.0.0 10.2.2.1
R3(config)#

```

```
R2#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R2(config)#ip route 223.168.1.0 255.255.255.0 10.1.1.1  
R2(config)#ip route 223.168.3.0 255.255.255.0 10.2.2.2  
R2(config) #
```

```
R2(config)# ip route 223.168.1.0 255.255.255.0 10.1.1.1
```

```
R2(config)# ip route 223.168.3.0 255.255.255.0 10.2.2.2
```

```
foreach address {
```

```
223.168.1.1
```

```
10.1.1.1
```

```
10.1.1.2
```

```
10.2.2.1
```

```
10.2.2.2
```

```
223.168.3.1}
```

```
{ping $address}
```

```
R1#tclsh  
R1(tcl)#foreach address {  
+>(tcl)#223.168.1.1  
+>(tcl)#10  
+>(tcl)#10.1.1.1  
+>(tcl)#10.1.1.2  
+>(tcl)#10.2.2.1  
+>(tcl)#10.2.2.2  
+>(tcl)#223.168.3.1  
+>(tcl)#{ ping $address }
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 223.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms% Unrecognized host or address, or protocol not running.

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 = 1/9/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/16 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 = 1/1/1 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 = 1/6/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 223.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/28 ms
R1(tcl) #
```

Step 3: Secure management access.

1. On R1, use the security passwords command to set a minimum password length of 10 characters.
2. **R1(config)# security passwords min-length 10**
3. 2. Configure the enable secret encrypted password on both routers.
R1(config)# enable secret class12345.
4. 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)#line console 0
R1(config-line)#password cisconpass
R1(config-line)#exec-timeout 5 0
R1(config-line)#login
R1(config-line)#logging synchronous
R1(config-line)#exit
```

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

5. 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
```

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

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

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

```
R1(config)#service password-encryption
R1(config)#banner motd $Unauthorized access strickly prohibited!$%
R1(config)#exit
```

Step 4: Configure enhanced username password security.

1. 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:
2. R1(config)# username Shivam secret class12345
3. R1(config)# username Shivam secret class5432

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#username Shivam secret class12345
R1(config)#username Shivam secret class54321
R1(config)#{
```

4. 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)#
R1(config)#line console 0
R1(config-line)#login local
R1(config-line)#exit
```

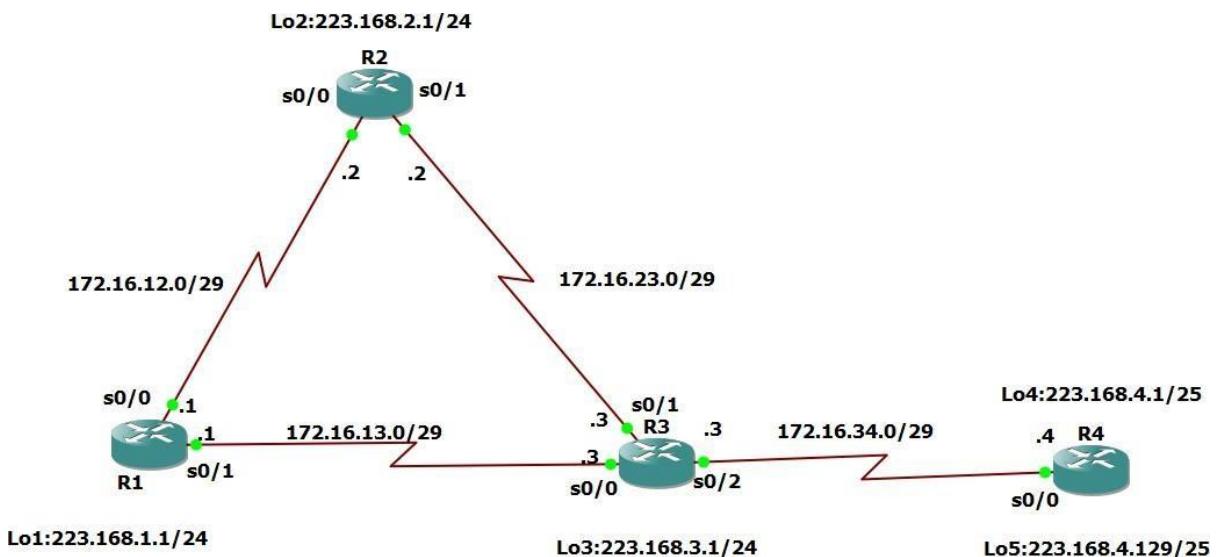
```
R1(config)# line vty 0 4
R1(config-line) # login local
R1(config-line) # end
```

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

Practical No-5

Aim: Configure and Verify Path Control Using PBR

Topology:



Objectives:

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

Step 1: Configure loopbacks and assign addresses.

- x. Cable the network as shown in the topology diagram. Erase the startup configuration, and reload each router to clear previous configurations.
- y. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to these and the serial interfaces on R1, R2, R3, and R4. On the serial interfaces connecting R1 to R3 and R3 to R4, specify the bandwidth as 64 Kb/s and set a clock rate on the DCE using the clock rate 64000 command. On the serial interfaces connecting R1 to R2 and R2 to R3, specify the bandwidth as 128 Kb/s and set a clock rate on the DCE using the clock rate 128000 command.

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

Router R1

Interface Lo1

```
ip address 223.168.1.1 255.255.255.0
interface Serial 0/0/0
ip address 172.16.12.1 255.255.255.248
no shutdown
interface Serial 0/0/1
ip address 172.16.13.1 255.255.255.248
no shutdown
end
```

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int Lo1
R1(config-if)#ip addre
*Mar 1 00:02:18.311: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,
changed state to up
R1(config-if)#ip address 223.168.1.1 255.255.255.0
R1(config-if)#int s0/0
R1(config-if)#ip address 172.16.12.1 255.255.255.248
R1(config-if)#no shutdown
R1(config-if)#exit
```

```
R1(config)#int s0/1
R1(config-if)#ip address 172.16.13.1
*Mar 1 00:04:03.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,
changed state to down
R1(config-if)#ip address 172.16.13.1 255.255.255.248
R1(config-if)#no shutdown
```

Router R2

Interface Lo2

```
ip address 223.168.2.1 255.255.255.0
interface Serial 0/0/0
ip address 172.16.12.2 255.255.255.248
no shutdown
interface Serial 0/0/1
ip address 172.16.23.2 255.255.255.248
no shutdown
end
```

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int Lo2
R2(config-if)#i
*Mar 1 00:03:50.395: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback2,
changed state to up
R2(config-if)#ip address 223.168.2.1 255.255.255.0
R2(config-if)#int s0/0
R2(config-if)#ip address 172.16.12.2 255.255.255.248
R2(config-if)#no shutdown
```

```
R2(config)#int s0/1
R2(config-if)#ip address 172.16.23.2 255.255.255.248
R2(config-if)#no shutdown
```

Router R3

Interface Lo3

```
ip address 223.168.3.1 255.255.255.0
interface Serial 0/0/0
ip address 172.16.13.3 255.255.255.248
no shutdown
interface Serial 0/0/1
ip address 172.16.23.3 255.255.255.248
interface Serial 0/1/0
ip address 172.16.34.3 255.255.255.248
no shutdown
end
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int Lo3
R3(config-if)#
*Mar 1 00:05:14.415: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback3,
changed state to up
R3(config-if)#ip address 223.168.3.1 255.255.255.0
R3(config-if)#int s0/0
R3(config-if)#ip address 172.16.13.3 255.255.255.248
R3(config-if)#no shutdown
```

```
R3(config-if)#int s0/1
R3(config-if)#ip address 172.16.23.3 255.255.255.248
R3(config-if)#no shutdown
```

```
R3(config)#int s0/2
R3(config-if)#ip address 172.16.34.3 255.255.255.248
R3(config-if)#no shutdown
R3(config-if)#exit
```

Router R4

Interface Lo4

ip address 223.168.4.1 255.255.255.128

interface Lo5

ip address 223.168.4.129 255.255.255.128

interface Serial 0/0/0

ip address 172.16.34.4 255.255.255.248

no shutdown

end

```
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#int Lo4
R4(config-if)#
*Mar 1 00:06:45.787: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback4,
    changed state to up
R4(config-if)#ip address 223.168.4.1 255.255.255.128
R4(config-if)#exit
```

```
R4(config)#interface Lo5
R4(config-if)#
*Mar 1 00:07:23.355: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback5,
    changed state to up
R4(config-if)#ip address 223.168.4.129 255.255.255.128
R4(config-if)#exit
R4(config)#int s0/0
R4(config-if)#ip address 172.16.34.4 255.255.255.248
R4(config-if)#no shutdown
```

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

R3# show ip interface brief

Interface ocol	IP-Address	OK? Method	Status	Prot
FastEthernet0/0	unassigned	YES unset	administratively down	down
Serial0/0	172.16.13.3	YES manual	up	up
FastEthernet0/1	unassigned	YES unset	administratively down	down
Serial0/1	172.16.23.3	YES manual	up	up
Serial0/2	172.16.34.3	YES manual	up	up
FastEthernet1/0	unassigned	YES unset	administratively down	down
Loopback3	223.168.3.1	YES manual	up	up

R3# show protocols

```
R3#show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
Serial0/0 is up, line protocol is up
  Internet address is 172.16.13.3/29
FastEthernet0/1 is administratively down, line protocol is down
Serial0/1 is up, line protocol is up
  Internet address is 172.16.23.3/29
Serial0/2 is up, line protocol is up
  Internet address is 172.16.34.3/29
FastEthernet1/0 is administratively down, line protocol is down
Loopback3 is up, line protocol is up
  Internet address is 223.168.3.1/24
```

R3# show interface description

Interface	Status	Protocol	Description
Fa0/0	admin down	down	
Se0/0	up	up	
Fa0/1	admin down	down	
Se0/1	up	up	
Se0/2	up	up	
Fa1/0	admin down	down	
Lo3	up	up	

Step 3: Configure basic EIGRP.

- a. Implement EIGRP AS 1 over the serial and loopback interfaces as you have configured it for the other EIGRP labs.
- b. Advertise networks 172.16.12.0/29, 172.16.13.0/29, 172.16.23.0/29, 172.16.34.0/29, 223.168.1.0/24, 223.168.2.0/24, 223.168.3.0/24, and 223.168.4.0/24 from their respective routers.

Router R1

```
Router eigrp 1
network 223.168.1.0
network 172.16.12.0 0.0.0.7
network 172.16.13.0 0.0.0.7
no auto-summary
```

```
R1(config)#router eigrp 1
R1(config-router)#network 223.168.1.0
R1(config-router)#network 172.16.12.0 0.0
% Incomplete command.

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)#

```

Router R2

```
Router eigrp 1
network 223.168.2.0
network 172.16.12.0 0.0.0.7
network 172.16.23.0 0.0.0.7
no auto-summary
```

```
R2(config)#router eigrp 1
R2(config-router)#network 223.168.2.0
R2(config-router)#network 172.16.12.0 0.0.0.7
R2(config-router)#n
*Mar 1 00:16:16.539: %DUAL-5-NBRCHANGE: IP-EIGRP(0)
rial0/0) is up: new adjacency
R2(config-router)#network 172.16.23.0 0.0.0.7
R2(config-router)#no auto-summary
R2(config-router)#

```

Router R3

```
Router eigrp 1
network 223.168.3.0
network 172.16.13.0 0.0.0.7
network 172.16.23.0 0.0.0.7
network 172.16.34.0 0.0.0.7
no auto-summary
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#router eigrp 1
R3(config-router)#network 223.168.3.0
R3(config-router)#network 172.16.13.0 0.0.0.7
R3(config-router)#
*Mar 1 00:16:38.731: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Serial0/0) is up: new adjacency
R3(config-router)#network 172.16.23.0 0.0.0.7
R3(config-router)#no
*Mar 1 00:17:16.239: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Serial0/1) is up: new adjacency
R3(config-router)#network 172.16.34.0 0.0.0.7
R3(config-router)#no auto-summary
```

Router R4

```
router eigrp 1
network 223.168.4.0
network 172.16.34.0 0.0.0.7
no auto-summary
```

```
R4(config)#router eigrp 1
R4(config-router)#network 223.168.4.0
R4(config-router)#network 172.16.34.0 0.0.0.7
R4(config-router)#no
*Mar 1 00:16:41.287: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Serial0/0) is up: new adjacency
R4(config-router)#no auto-summary
R4(config-router)#

```

Step 4: Verify EIGRP connectivity.

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

R1# show ip eigrp neighbors

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Seq
   (sec)          (ms)          Cnt Num
1   172.16.13.3       Se0/1        10 00:02:47  45     270   0   27
0   172.16.12.2       Se0/0        12 00:04:11  41     246   0   24
R1#
```

R2# show ip eigrp neighbors

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Seq
   (sec)          (ms)          Cnt Num
1   172.16.23.3       Se0/1        11 00:03:03  44     264   0   28
0   172.16.12.1       Se0/0        13 00:05:03  34     204   0   26
R2#
```

R3# show ip eigrp neighbors

```
R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Seq
   (sec)          (ms)          Cnt Num
2   172.16.34.4       Se0/2        11 00:02:04  55     330   0   7
1   172.16.23.2       Se0/1        10 00:03:26  42     252   0   25
0   172.16.13.1       Se0/0        14 00:04:03  36     216   0   27
R3#
```

R4# show ip eigrp neighbors

```
R4#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address           Interface      Hold Uptime    SRTT     RTO   Q   Seq
   (sec)          (ms)          Cnt Num
0   172.16.34.3       Se0/0        10 00:02:30  30     200   0   32
R4#
```

d. Run the following TCL Scripts on all routers to verify full connectivity.

```
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#
+>(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)#223.168.1.1
+>(tcl)#223.168.2.1
+>(tcl)#223.168.3.1
+>(tcl)#223.168.4.1
+>(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 = 1/1/1 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 = 1/8/20 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 = 1/9/32 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 = 1/1/1 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 = 1/3/12 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 = 1/1/1 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 = 1/4/20 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
```

Step 5: Verify the current path.

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

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

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

Gateway of last resort is not set

D  223.168.2.0/24 [90/2297856] via 172.16.12.2, 00:43:23, Serial0/0
D  223.168.3.0/24 [90/2297856] via 172.16.13.3, 00:43:23, Serial0/1
C  223.168.1.0/24 is directly connected, Loopback1
    223.168.4.0/25 is subnetted, 2 subnets
        D  223.168.4.0 [90/2809856] via 172.16.13.3, 00:41:54, Serial0/1
        D  223.168.4.128 [90/2809856] via 172.16.13.3, 00:41:54, Serial0/1
            172.16.0.0/29 is subnetted, 4 subnets
                D  172.16.34.0 [90/2681856] via 172.16.13.3, 00:43:06, Serial0/1
                D  172.16.23.0 [90/2681856] via 172.16.13.3, 00:43:25, Serial0/1
                    [90/2681856] via 172.16.12.2, 00:43:25, Serial0/0
            C  172.16.12.0 is directly connected, Serial0/0
            C  172.16.13.0 is directly connected, Serial0/1
```

R4# traceroute 223.168.1.1 source 223.168.4.1

```
R4#traceroute 223.168.1.1 source 223.168.4.1

Type escape sequence to abort.
Tracing the route to 223.168.1.1

 1 172.16.34.3 0 msec 0 msec 0 msec
 2 172.16.13.1 0 msec 0 msec 0 msec
```

R4# traceroute 223.168.1.1 source 223.168.4.129

```
R4#traceroute 223.168.1.1 source 223.168.4.129

Type escape sequence to abort.
Tracing the route to 223.168.1.1

 1 172.16.34.3 0 msec 0 msec 0 msec
 2 172.16.13.1 0 msec 0 msec 0 msec
```

On R3, use the show ip route command and note that the preferred route from R3 to R1 LAN 223.168.1.0/24 is via R2 using the R3 exit interface S0/0/1

R3# show ip route

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

D    223.168.2.0/24 [90/2297856] via 172.16.23.2, 00:47:21, Serial0/1
C    223.168.3.0/24 is directly connected, Loopback3
D    223.168.1.0/24 [90/2297856] via 172.16.13.1, 00:47:21, Serial0/0
     223.168.4.0/25 is subnetted, 2 subnets
D      223.168.4.0 [90/2297856] via 172.16.34.4, 00:45:50, Serial0/2
D      223.168.4.128 [90/2297856] via 172.16.34.4, 00:45:50, Serial0/2
     172.16.0.0/29 is subnetted, 4 subnets
C        172.16.34.0 is directly connected, Serial0/2
C        172.16.23.0 is directly connected, Serial0/1
D        172.16.12.0 [90/2681856] via 172.16.23.2, 00:47:23, Serial0/1
                  [90/2681856] via 172.16.13.1, 00:47:23, Serial0/0
C        172.16.13.0 is directly connected, Serial0/0
```

- f. On R3, use the **show interfaces serial 0/0/0** and **show interfaces s0/0/1** command.

```
R3#show int s0/0
Serial0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 172.16.13.3/29
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  CRC checking enabled
  Last input 00:00:01, output 00:00:01, 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 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    1189 packets input, 79137 bytes, 0 no buffer
    Received 445 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    1132 packets output, 75981 bytes, 0 underruns
    0 output errors, 0 collisions, 6 interface resets
    0 output buffer failures, 0 output buffers swapped out
    0 carrier transitions
  DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

- g. Confirm that R3 has a valid route to reach R1 from its serial 0/0/0 interface using the **show ip eigrp topology 223.168.1.0** command.

R3# **show ip eigrp topology 223.168.1.0**

```
R3#show ip eigrp topology 223.168.1.0
IP-EIGRP (AS 1): Topology entry for 223.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
  Routing Descriptor Blocks:
    172.16.13.1 (Serial0/0), from 172.16.13.1, Send flag is 0x0
      Composite metric is (2297856/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
    172.16.23.2 (Serial0/1), from 172.16.23.2, Send flag is 0x0
      Composite metric is (2809856/2297856), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

Step 6: Configure PBR to provide path control

The steps required to implement path control include the following:

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

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

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

- h.** 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 223.168.4.128 0.0.0.127
R3(config-std-nacl) #exit
```

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#ip access-list standard PBR-ACL
R3(config-std-nacl)#remark ACL matches R4 LAN B traffic
R3(config-std-nacl)#permit 223.168.4.128 0.0.0.127
R3(config-std-nacl)#exit
```

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

```
R3(config)# route-map R3-to-R1 permit
```

```
R3(config-route-map) # description RM to forward LAN B traffic to R1
```

```
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) #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
```

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

```
R3(config)# interface s0/1/0
```

```
R3(config-if) #ip policy route-map R3-to-R1
```

```
R3(config-if) # end
```

```
R3(config)#int s0/2
R3(config-if) #ip policy route-map R3-to-R1
R3(config-if) #end
```

- k.** 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
Policy routing matches: 0 packets, 0 bytes
```

Step 7: Test the Policy

- 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 223.168.4.0 0.0.0.255

R3(config)# exit

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#access-list 1 permit 223.168.4.0 0.0.0.255
R3(config)#exit
```

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

R3# debug ip policy?

```
R3#debug ip policy ?
  dynamic  dynamic PBR
```

R3# debug ip policy 1

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

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

R4# traceroute 223.168.1.1 source 223.168.4.1

```
R4#traceroute 223.168.1.1 source 223.168.4.1
Type escape sequence to abort.
Tracing the route to 223.168.1.1

 1 172.16.34.3 16 msec 8 msec 4 msec
 2 172.16.13.1 0 msec 0 msec 0 msec
```

```
*Mar 1 01:17:11.243: IP: s=223.168.4.1 (Serial0/2), d=223.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
*Mar 1 01:17:11.243: IP: s=223.168.4.1 (Serial0/2), d=223.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
*Mar 1 01:17:11.243: IP: s=223.168.4.1 (Serial0/2), d=223.168.1.1, len 28, FIB
policy rejected(no match) - normal forwarding
```

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

R4# traceroute 223.168.1.1 source 223.168.4.129

```
R4#traceroute 223.168.1.1 source 223.168.4.129
Type escape sequence to abort.
Tracing the route to 223.168.1.1

 1 172.16.34.3 28 msec 0 msec 0 msec
 2 172.16.13.1 0 msec 16 msec 0 msec
```

```
*Mar 1 01:19:54.267: IP: s=223.168.4.129 (Serial0/2), d=223.168.1.1, len 28, FI
B policy match
*Mar 1 01:19:54.267: IP: s=223.168.4.129 (Serial0/2), d=223.168.1.1, g=172.16.1
3.1, len 28, FIB policy routed
*Mar 1 01:19:54.267: IP: s=223.168.4.129 (Serial0/2), d=223.168.1.1, len 28, FI
B policy match
*Mar 1 01:19:54.267: IP: s=223.168.4.129 (Serial0/2), d=223.168.1.1, g=172.16.1
3.1, len 28, FIB policy routed
*Mar 1 01:19:54.283: IP: s=223.168.4.129 (Serial0/2), d=223.168.1.1, len 28, FI
B policy match
*Mar 1 01:19:54.283: IP: s=223.168.4.129 (Serial0/2), d=223.168.1.1, g=172.16.1
3.1, len 28, FIB policy routed
```

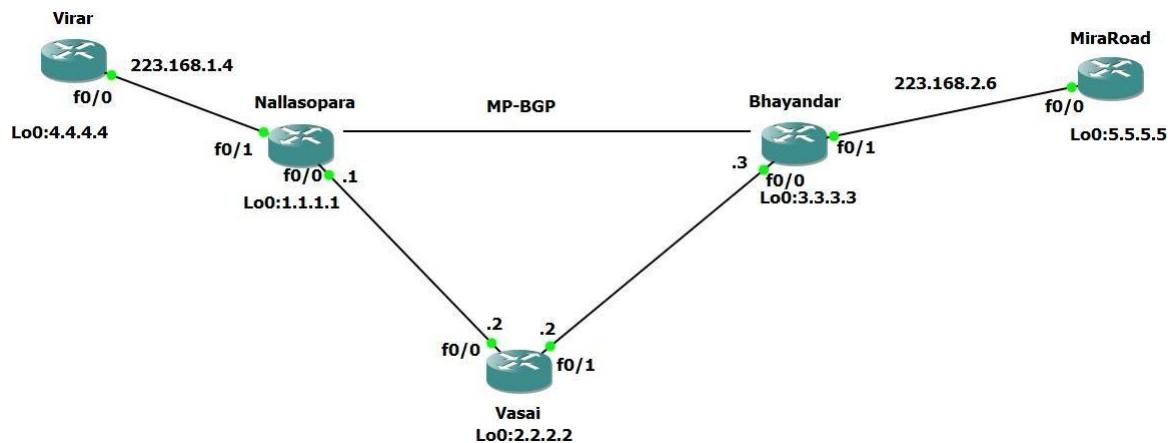
5. On R3, display the policy and matches using the show route-map command.

```
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
  Policy routing matches: 3 packets, 96 bytes
```

Practical No 6

Aim: Cisco MPLS Configuration

Topology:



Step 1 – IP addressing of MPLS Core and OSPF

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

```

Nallasopara#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Nallasopara(config)#int lo0
Nallasopara(config-if)#
*Mar 1 00:02:30.835: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
Nallasopara(config-if)#ip add 1.1.1.1 255.255.255.255
Nallasopara(config-if)#ip ospf 1 area 0
Nallasopara(config-if)#
Nallasopara(config-if)#int f0/0
Nallasopara(config-if)#ip add 10.0.0.1 255.255.255.0
Nallasopara(config-if)#no shut
Nallasopara(config-if)#ip ospf
*Mar 1 00:04:00.111: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:04:01.111: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Nallasopara(config-if)#ip ospf 1 area 0
Nallasopara(config-if)#
  
```

```
Vasai#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Vasai(config)#int lo0
Vasai(config-if)#
*Mar 1 00:03:11.943: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
Vasai(config-if)#ip add 2.2.2.2 255.255.255.255
Vasai(config-if)#ip ospf 1 area 0
Vasai(config-if)#
Vasai(config-if)#int f0/0
Vasai(config-if)#ip add 10.0.0.2 255.255.255.0
Vasai(config-if)#no shut
Vasai(config-if)#ip
*Mar 1 00:04:28.359: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:04:29.359: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Vasai(config-if)#ip ospf 1 area 0
Vasai(config-if)#
Vasai(config-if)#int f0/
*Mar 1 00:04:47.391: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0
from LOADING to FULL, Loading Done
Vasai(config-if)#int f0/1
Vasai(config-if)#ip add 10.0.1.2 255.255.255.0
Vasai(config-if)#no shut
Vasai(config-if)#ip ospf
*Mar 1 00:05:32.339: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:05:33.339: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
Vasai(config-if)#ip ospf 1 area 0
```

```
Bhayandar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Bhayandar(config)#int lo0
Bhayandar(config-if)#
*Mar 1 00:04:46.155: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
Bhayandar(config-if)#ip add 3.3.3.3 255.255.255.255
Bhayandar(config-if)#ip ospf 1 area 0
Bhayandar(config-if)#
Bhayandar(config-if)#int f0/0
Bhayandar(config-if)#ip add 10.0.1.3 255.255.255.0
Bhayandar(config-if)#no shut
Bhayandar(config-if)#ip ospf
*Mar 1 00:06:07.163: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:06:08.163: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Bhayandar(config-if)#ip ospf 1 area 0
```

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

```
Nallasopara#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 = 48/60/84 ms
```

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.

```
Nallasopara(config)#router ospf 1
Nallasopara(config-router)#mpls ldp autoconfig
Nallasopara(config-router)#End configuration
```

```
Vasai(config)#router ospf 1
Vasai(config-router)#mpls ldp autoconfig
Vasai(config-router)#router
*Mar 1 00:11:28.635: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
Vasai(config-router)#
*Mar 1 00:12:49.523: %LDP-5-NBRCHG: LDP Neighbor 3.3.3.3:0 (2) is UP
```

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

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

```
Bhayandar(config)#router ospf 1
Bhayandar(config-router)#mpls ldp autoconfig
Bhayandar(config-router)#End configuration
```

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

Interface	IP	Tunnel	Operational
FastEthernet0/0	Yes (ldp)	No	Yes
FastEthernet0/1	Yes (ldp)	No	Yes

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

```
Vasai#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 - 2.2.2.2.24585
  State: Oper; Msgs sent/rcvd: 11/11; Downstream
  Up time: 00:03:06
  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: 3.3.3.3.42628 - 2.2.2.2.646
  State: Oper; Msgs sent/rcvd: 10/10; Downstream
  Up time: 00:01:46
  LDP discovery sources:
    FastEthernet0/1, Src IP addr: 10.0.1.3
  Addresses bound to peer LDP Ident:
    10.0.1.3      3.3.3.3
```

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

```
Nallasopara#trace 3.3.3.3
Type escape sequence to abort.
Tracing the route to 3.3.3.3

 1 10.0.0.2 [MPLS: Label 17 Exp 0] 44 msec 76 msec 60 msec
 2 10.0.1.3 64 msec 72 msec 28 msec
```

Step 3 – MPLS BGP Configuration between R1 and R3

We need to establish a Multi Protocol BGP session between R1 and R3 this is done by configuring the vpng4 address family as below.

```
Nallasopara#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Nallasopara(config)#router bgp 1
Nallasopara(config-router)#neighbor 3.3.3.3 remote-as 1
Nallasopara(config-router)#neighbor 3.3.3.3 update-source Loopback0
Nallasopara(config-router)#no auto-summary
Nallasopara(config-router)#address-family vpng4
Nallasopara(config-router-af)#neighbor 3.3.3.3 activate
```

```
Bhayandar(config)#router bgp 1
Bhayandar(config-router)#neighbor 1.1.1.1 remote-as 1
Bhayandar(config-router)#neighbor 1.1.1.1
% Incomplete command.

Bhayandar(config-router)#
*Mar 1 00:18:56.079: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
Bhayandar(config-router)#neighbor 1.1.1.1 update-source loopback 0
Bhayandar(config-router)#no auto-summary
Bhayandar(config-router)#address-family vpng4
Bhayandar(config-router-af)#neighbor 1.1.1.1 activate
```

To verify the BGP session between R1 and R3 issue the command sh bgp vpng4 unicast all summary.

```
Nallasopara#sh bgp vpng4 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       8       8           1       0       0 00:00:59          0
Nallasopara#
```

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.

```
Virar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Virar(config)#int lo0
Virar(config-if)#
*Mar 1 00:11:25.043: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
changed state to up
Virar(config-if)#ip address 4.4.4.4 255.255.255.255
Virar(config-if)#ip ospf 2 area 2
Virar(config-if)#int f0/0
Virar(config-if)#ip address 223.168.1.4 255.255.255.0
Virar(config-if)#ip ospf 2 area 2
Virar(config-if)#no shut
```

```
Nallasopara#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Nallasopara(config)#int f0/1
Nallasopara(config-if)#no shut
Nallasopara(config-if)#
*Mar 1 00:29:01.831: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to
o up
*Mar 1 00:29:02.831: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
Nallasopara(config-if)#ip address 223.168.1.1 255.255.255.0
```

```
Nallasopara(config)#
Nallasopara(config)#ip vrf RED
Nallasopara(config-vrf)#rd 4:4
Nallasopara(config-vrf)#route-target both 4:4
Nallasopara(config-vrf)#
Nallasopara(config-vrf)#
Nallasopara(config-vrf)#int f0/1
Nallasopara(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 223.168.1.1 removed due to enabling VRF R
ED
Nallasopara(config-if)#ip add 223.168.1.1 255.255.255.0
Nallasopara(config-if)#
Nallasopara#
```

```
Nallasopara#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
  ip vrf forwarding RED
  ip address 223.168.1.1 255.255.255.0
  duplex auto
  speed auto
end
```

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.

```
Nallasopara#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      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 [110/11] via 10.0.0.2, 00:30:41, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/21] via 10.0.0.2, 00:26:56, FastEthernet0/0
      10.0.0.0/24 is subnetted, 2 subnets
C          10.0.0.0 is directly connected, FastEthernet0/0
O          10.0.1.0 [110/20] via 10.0.0.2, 00:29:46, FastEthernet0/0
Nallasopara#
```

```
Nallasopara(config)#
Nallasopara(config)#ip vrf RED
Nallasopara(config-vrf)#rd 4:4
Nallasopara(config-vrf)#route-target both 4:4
Nallasopara(config-vrf)#
Nallasopara(config-vrf)#
Nallasopara(config-vrf)#int f0/1
Nallasopara(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 223.168.1.1 removed due to enabling VRF R
ED
Nallasopara(config-if)#ip add 223.168.1.1 255.255.255.0
Nallasopara(config-if)#

```

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

```
Nallasopara(config)#int f0/1
Nallasopara(config-if)#ip ospf 2 area 2
Nallasopara(config-if)#

```

```
Nallasopara#sh ip route vrf RED

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

Gateway of last resort is not set

C    223.168.1.0/24 is directly connected, FastEthernet0/1
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/11] via 223.168.1.4, 00:00:20, FastEthernet0/1

```

We now need to repeat this process for R3 & R6 Router 6 will peer OSPF using process number 2 to a VRF configured on R3. It will use the local site addressing to 223.168.2.0/24.

```
Nallasopara#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      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 [110/11] via 10.0.0.2, 00:33:12, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O          3.3.3.3 [110/21] via 10.0.0.2, 00:29:27, FastEthernet0/0
      10.0.0.0/24 is subnetted, 2 subnets
C            10.0.0.0 is directly connected, FastEthernet0/0
O            10.0.1.0 [110/20] via 10.0.0.2, 00:32:17, FastEthernet0/0
Nallasopara#
```

```
MiraRoad#conf t
Enter configuration commands, one per line. End with CNTL/Z.
MiraRoad(config)#int lo0
MiraRoad(config-if)#ip
*Mar 1 00:17:16.419: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0,
  changed state to up
MiraRoad(config-if)#ip add 6.6.6.6 255.255.255.255
MiraRoad(config-if)#ip ospf 2 area 2
MiraRoad(config-if)#int f0/0
MiraRoad(config-if)#ip address 223.168.2.6 255.255.255.0
MiraRoad(config-if)#ip ospf 2 area 2
MiraRoad(config-if)#no shut
```

```
Bhayandar(config)#int f0/1
Bhayandar(config-if)#no shut
Bhayandar(config-if)#
Bhayandar(config-if)#ip add
*Mar 1 00:43:46.895: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to
  up
*Mar 1 00:43:47.895: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
Bhayandar(config-if)#ip add 223.168.2.3 255.255.255.0
```

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

Check the Router in VRF RED

```
Bhayandar(config-vrf)#int f0/1
Bhayandar(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 223.168.2.3 removed due to enabling VRF R
ED
Bhayandar(config-if)#ip add 223.168.2.1 255.255.255.0
```

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

```
Bhayandar#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
  ip vrf forwarding RED
  ip address 223.168.2.1 255.255.255.0
  duplex auto
  speed auto
end
```

```
Bhayandar#sh ip route vrf RED

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

Gateway of last resort is not set

C    223.168.2.0/24 is directly connected, FastEthernet0/1
```

```
Virar#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    223.168.1.0/24 is directly connected, FastEthernet0/0
        4.0.0.0/32 is subnetted, 1 subnets
C          4.4.4.4 is directly connected, Loopback0
```

As expected we have the local interface and the loopback address. When we are done we want to see 6.6.6.6 in there so we can ping across the MPLS Check the routes on R1.

```
Nallasopara#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      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 [110/11] via 10.0.0.2, 00:33:12, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O          3.3.3.3 [110/21] via 10.0.0.2, 00:29:27, FastEthernet0/0
      10.0.0.0/24 is subnetted, 2 subnets
C          10.0.0.0 is directly connected, FastEthernet0/0
O          10.0.1.0 [110/20] via 10.0.0.2, 00:32:17, FastEthernet0/0
Nallasopara#
```

```
Nallasopara#
Nallasopara#sh ip route vrf RED

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

Gateway of last resort is not set

C    223.168.1.0/24 is directly connected, FastEthernet0/1
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/11] via 223.168.1.4, 00:13:07, FastEthernet0/1
```

```
Nallasopara#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Nallasopara(config)#
Nallasopara(config)#router bgp 1
Nallasopara(config-router)#address-family ipv4 vrf RED
Nallasopara(config-router-af)#redistribute ospf 2
Nallasopara(config-router-af)#exit
Nallasopara(config-router)#end
```

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

```
Nallasopara#sh ip bgp vpnv4 vrf RED
BGP table version is 5, 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
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
*> 4.4.4.4/32        223.168.1.4           11      32768  ?
*> 223.168.1.0       0.0.0.0             0      32768  ?
```

```
Bhayandar#sh ip bgp vpnv4 vrf RED
BGP table version is 5, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
*>i4.4.4.4/32       1.1.1.1           11      100      0 ?
*>i223.168.1.0      1.1.1.1           0      100      0 ?
```

Which it is! 6.6.6.6 is now in the BGP table in VRF RED on R3 with a next hop of 192.168.2.6 (R6) and also 4.4.4 is in there as well with a next hop of 1.1.1.1 (which is the loopback of R1 – showing that it is going over the MPLS and R2 is not in the picture).

```
Nallasopara(config)#
Nallasopara(config-if)#int f0/1
Nallasopara(config-if)#ip ospf 2 area 2
Nallasopara(config-if)#

```

```
Bhayandar#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Bhayandar(config)#router ospf 2
Bhayandar(config-router)#redistribute bgp 1 subnets
Bhayandar(config-router)#

```

Before we do let's see what the routing table look like on R.

```
Virar#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    223.168.1.0/24 is directly connected, FastEthernet0/0
      4.0.0.0/32 is subnetted, 1 subnets
C        4.4.4.4 is directly connected, Loopback0
.
```

```
Bhayandar#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
O        1.1.1.1 [110/21] via 10.0.1.2, 04:33:27, FastEthernet0/0
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/11] via 10.0.1.2, 04:33:27, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
C        3.3.3.3 is directly connected, Loopback0
      10.0.0.0/24 is subnetted, 2 subnets
O        10.0.0.0 [110/20] via 10.0.1.2, 04:33:27, FastEthernet0/0
C        10.0.1.0 is directly connected, FastEthernet0/0
Bhayandar#
```

Do the Same Step in R6.

```
MiraRoad#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    223.168.2.0/24 is directly connected, FastEthernet0/0
      6.0.0.0/32 is subnetted, 1 subnets
C        6.6.6.6 is directly connected, Loopback0
.
```

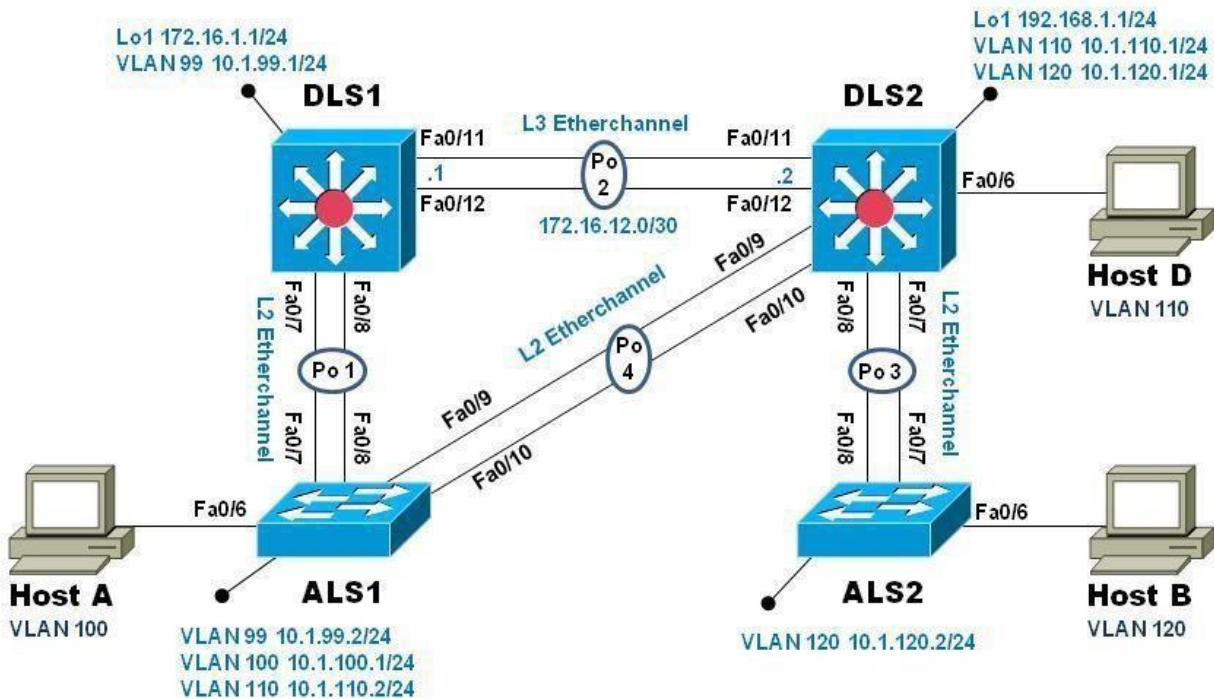
Nallasopara#ping 3.3.3.3 source lo0

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 = 48/60/84 ms

Practical No 7

Aim: Inter-VLAN Routing

Topology



Objectives

- Implement a Layer 3 EtherChannel
- Implement Static Routing
- Implement Inter-VLAN Routing

Required Resources

- 2 Cisco 2960 with the Cisco IOS Release 15.0(2)SE6 C2960-LANBASEK9-M or comparable
- 2 Cisco 3560v2 with the Cisco IOS Release 15.0(2)SE6 C3560-IPSERVICESK9-M or comparable
- Computer with terminal emulation software
- Ethernet and console cables
- 3 PCs with appropriate software

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. Thenload 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 223.168.1.1 255.255.255.0
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)#

```

The screenshot shows a Windows command prompt window titled "Windows IP Configuration". It displays the configuration for the "Ethernet adapter Local Area Connection". The adapter has the following details:

- 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

Below the adapter details, a "ping" command is run from the prompt "C:\Users\student>" to the IP address 192.168.1.1. The output shows four successful replies with a round-trip time of 1ms each.

```
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)#

```

Once you have configured both sides, verify that the EtherChannel link is up

```

DLS2# 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: 1
Number of aggregators:           1

Group  Port-channel  Protocol     Ports
-----+-----+-----+
2      Po2 (RU)       PAgP        Fa0/11 (P)   Fa0/12 (P)

DLS2# ping 172.16.12.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/3/9 ms
DLS2#

```

Step 5: Configure default routing between DLS switches

At this point, local routing is support at each distribution layer switch. Now to provide reachability across the layer 3 EtherChannel trunk, configure fully qualified static default routes at DLS1 and DLS2 that point to eachother. From DLS1:

```
DLS1(config)# ip route 0.0.0.0 0.0.0.0 port-channel 2
%Default route without gateway, if not a point-to-point interface, may impact
performance
DLS1(config)# ip route 0.0.0.0 0.0.0.0 port-channel 2 172.16.12.2
DLS1(config)#

```

Once done at both ends, verify connectivity by pinging from one switch to the other. In the example below, DLS2 pings the Loopback 1 interface at DLS1.

```
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-channel12
L        172.16.12.2/32 is directly connected, Port-channel12
          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 Vlans 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 inter-vlan 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) #
```

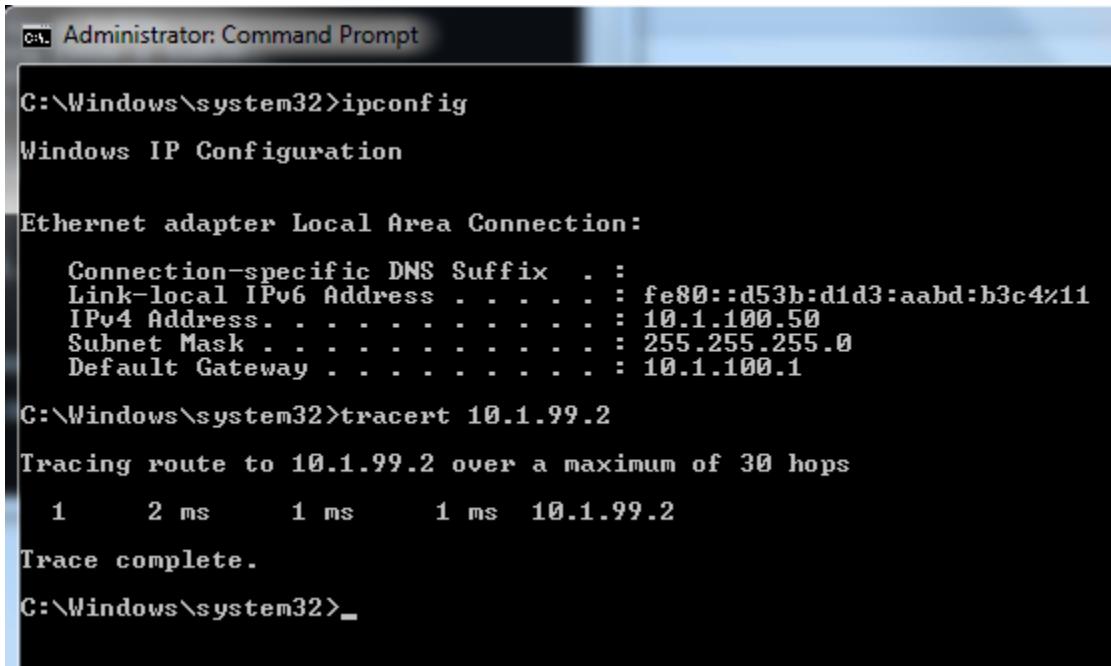
Step 2: **Configure and test Host Access**

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
  1      2 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, l - 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.

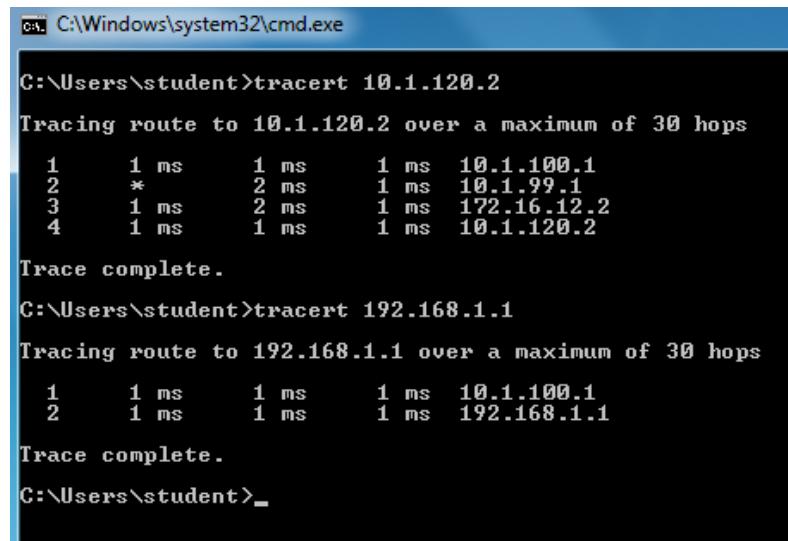
From ALS1, a traceroute to 10.1.120.2 should take three hops:

```
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
C:\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.

C:\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.

C:\Users\student>
```

Step 4: End of Lab

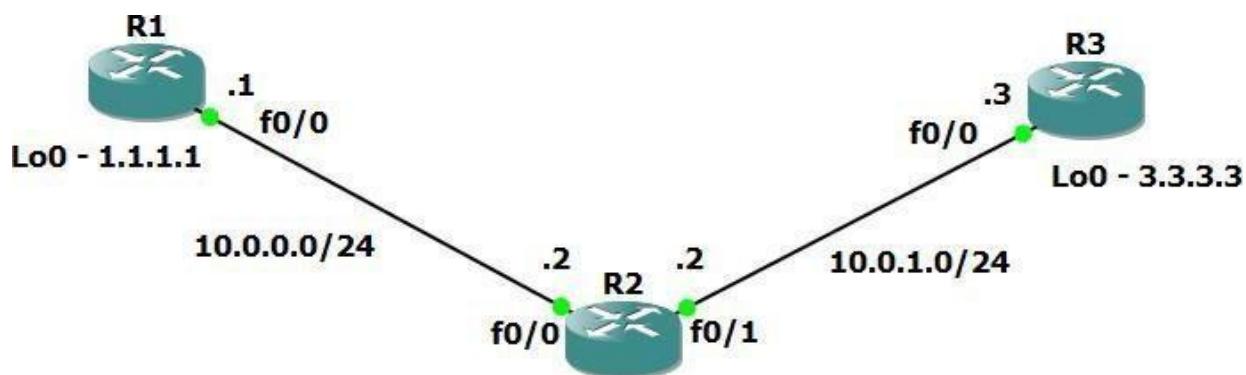
Save your configurations. The switches will be used as configured now for lab 5-2, DHCP.

Practical No 8

Aim: Cisco MPLS Configuration

Step 1 – IP addressing of MPLS Core and OSPF

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



```

R1
hostname
R1intlo0
ipaddr1.1.1.1255.255.255.255
ipospf1area0
intf0/0
ipaddr10.0.0.1255.255.255.0
no shut
ipospf1area0
  
```

```

R2
hostname
R2intlo0
  
```

```
ip add 2.2.2.2 255.255.255.255
```

```
ip ospf 1 area 0
```

```
int f0/0
```

```
ip add 10.0.0.2 255.255.255.0
```

```
no shut
```

```
ip ospf 1 area 0
```

```
int f0/1
```

```
ip add 10.0.1.2 255.255.255.0
```

```
no shut
```

```
ip ospf 1 area 0
```

```
R3
```

```
hostname R3 int lo0
```

```
ip add 3.3.3.3 255.255.255.255
```

```
ip ospf 1 area 0
```

```
int f0/0
```

```
ip add 10.0.1.3 255.255.255.0
```

```
no shut
```

```
ip ospf 1 area 0
```

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

```
R1#ping3.3.3.3source lo0

Type escape sequence to abort.

Sending 5, 100-byte ICMPEchoes to 3.3.3.3, timeout is 2 seconds:
Packets sent with a source address of 1.1.1.1
!!!!!

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

R1#
```

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

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

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

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

For this tutorial we will be using the second option, so go into 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
router ospf 1
mpls ldp autoconfig

R2
router ospf 1
```

```
R3  
router ospf 1  
mplsldpautoconfig
```

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

```
R2#  
*Mar1 00:31:53.643: %SYS-5-CONFIG_I: Configured fromconsole  
*Mar100:31:54.423:%LDP-5-NBRCHG:LDPNeighbor  
1.1.1.1:0 (1) is UPR2#  
*Mar100:36:09.951:%LDP-5-NBRCHG:LDPNeighbor  
3.3.3.3:0(2)isUP
```

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 interface  
  
Interface          IP          Tunnel  
Operational  
  
FastEthernet0/0    Yes (ldp)   No      Yes  
FastEthernet0/1    Yes (ldp)   No      Yes
```

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

```
R2#sh mpls ldpneigh  
  
PeerLDPIdent:1.1.1.1:0;LocalLDPIdent2.2.2.2:0TCPconnection:1.1.1.1.646  
-2.2.2.2.37909
```

State:Oper;Msgssent/rcvd:16/17;DownstreamUptim

e:00:07:46

LDPdiscoverysources:

FastEthernet0/0, SrcIPAddr:10.0.0.1 Addr

essesboundtopeerLDPIdent:

10.0.0.1 1.1.1.1

PeerLDPIdent:3.3.3.3:0;LocalLDPIdent2.2.2.2:0 TCPcon
nection:3.3.3.3.22155-2.2.2.2.646State:Oper;Msgssent/rcvd:12/11;DownstreamUptim
e:00:03:30

LDPdiscoverysources:

FastEthernet0/1, SrcIPAddr:10.0.1.3 Addr
essesboundtopeerLDPIdent:

10.0.1.3 3.3.3.3

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

R1#trace3.3.3.3

Type escape sequence to
abort.Tracingtherouteto3.3.3.3

110.0.0.2 [MPLS:Label17Exp0] 84 msec 72 msec 44 msec
210.0.1.3 68 msec 60 msec *

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

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

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

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

Step 3 – MPLS BGP Configuration between R1 and R3

We need to establish a Multi Protocol BGP session between R1 and R3 this is done by configuring the vpngv4 address family as below

```
R1#
router bgp 1
    neighbor 3.3.3.3 remote-as 1
    neighbor 3.3.3.3 update-source Loopback0 no auto-summary
!
address-family
    vpngv4
        neighbor 3.3.3.3 activate
    !
    exit-address-family

R3#
router bgp 1
    neighbor 1.1.1.1 remote-as 1
    neighbor 1.1.1.1 update-source Loopback0 no auto-summary
!
address-family
    vpngv4
        neighbor 1.1.1.1 activate
    !
    exit-address-family
```

```
*Mar100:45:01.047:%BGP-5-ADJCHANGE:neighbor1.1.1.1  
Up
```

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

To verify the BGP session between R1 and R3 issue the command **sh bgp vpng4 unicast all summary**

R1#sh bgp vpng4 unicast all summary
BGP table version is 1, main routing table version 1
Peer BGP Router Identifier 1.1.1.1, local AS number 1
Up/Down State/PfxRcd

NeighborOutQ V ASMsgRcvdMsgSent TblVerInQ
Up/Down State/PfxRcd

Step 4 - Add two more routers, create VRFs	3.3.3.3	4	1	218	218	1	0
We resolve add two more routers into the topology so it now looks like the final topology	0 03:17:48	0					

Router 4 will peer OSPF using process number 2 to a VRF configured on R1. It will use the local site addressing of 223.168.1.0/24.

```
R4  
  
int lo0  
  
ip add 4.4.4.4 255.255.255.255  
  
ip ospf 2 area  
  
2 int f0/0
```

```
ipaddr192.168.1.4255.255.255.0
```

```
ip ospf 2 area
```

```
2noshut
```

```
R1
```

```
int f0/1no
```

```
shut
```

```
ipaddr223.168.1.1255.255.255.0
```

Now at this point we have R4 peering to R1 but in the global routing table of R1 which is not what we want.

We are now going to start using VRF's

What is a VRF in networking?

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

As an example if R1 was a PE Provider Edge router of an ISP and it had two customers that were both addressed locally with the 223.168.1.0/24 address space it could accommodate both their routing tables in different VRFs – it distinguishes between the two of them using a Route Distinguisher

So back to the topology – we now need to create a VRF on R1 For this mpls tutorial I will

be using VRF RED

```
R1
```

```
ip vrf
```

```
REDrd4:4
```

```
route-targetboth4:4
```

The RD and route-target do not need to be the same – and for a full explanation please read this post on Route Distinguisher

[Route Distinguisher vs Route Target](#) before proceeding.

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

```
R1  
intf0/1  
ipvrfforwardingRED
```

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

```
R1(config-if)#ipvrffo  
R1(config-if)#ipvrfforwardingRED  
% Interface FastEthernet0/1 IP address  
223.168.1.1removedduetoenablingVRFRED
```

You just need to re-apply it

```
R1  
intf0/1  
ipaddress223.168.1.1255.255.255.0
```

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

```
R1  
R1#sh run int  
f0/1Buildingconfiguration...  
Currentconfiguration:119bytes
```

interface

```
FastEthernet0/1ipvrfforwardingRE
D
ipaddress223.168.1.1255.255.255.0
duplexautospeed
auto
end
```

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

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

If you issue the command **sh ip route** this shows the routes in the global table and you will notice that you do not see 223.168.1.0/24

```
R1#sh ip route
```

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

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2

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

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

o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

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

3.0.0.0/32 is subnetted, 1 subnets

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

10.0.0.0/24 is subnetted, 2 subnets

C 10.0.0.0 is directly connected, FastEthernet0/0
O 10.0.1.0 [110/20] via 10.0.0.2, 01:02:39,
FastEthernet0/0 R1#

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

```
R1#shiproutevrfred  
%IP routing table red does not exist
```

NOTE: The VRF name is case sensitive!

```
R1#shiproutevrfRED  
  
RoutingTable:RED  
  
Codes:C-connected, S-static, R - RIP, M-mobile, B  
-BGP  
  
D- EIGRP, EX- EIGRP external, O-OSPF, IA-OSPF  
interarea  
  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

E1-OSPFexternaltype1, E2-OSPFexternaltype2

i- IS-IS, su - IS-ISsummary, L1-IS-ISlevel-1, L2-
IS-IS level-2

ia - IS-IS inter area, * - candidate default, U - per-userstaticroute

o-ODR, P -periodicdownloadedstaticroute

Gatewayoflastresortisnotset

C223.168.1.0/24isdirectlyconnected, FastEthernet0/1R1#

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

R1

intf0/1
ip ospf 2 area2

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

R1(config-if)#
*Mar10 1:12:54.323: %OSPF-5-ADJCHG: Process 2, Nbr
4.4.4.4
onFastEthernet0/1 fromLOADINGtoFULL, LoadingDone

If we now check the routes in the VRF RED routing table you should see

4.4.4.4 in there as well.

```
R1#shiproutevrfRED
```

```
RoutingTable:RED
```

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

D- EIGRP, EX- EIGRP external, O-OSPF, IA-OSPF
interarea

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1-OSPF external type 1, E2-OSPF external type 2

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

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

o-ODR, P - periodic downloaded static route

Gateway of last resort is not set

4.0.0.0/32 is subnetted, 1 subnets

**O4.4.4.4[110/11] via 223.168.1.4, 00:00:22,
FastEthernet0/1**

C223.168.1.0/24 is directly connected, FastEthernet0/1 R1#

We now need to repeat this process for R3 & R6

Router 6 will peer OSPF using process number 2 to a VRF configured on R3. It will use the local site addressing of 223.168.2.0/24.

```
R6
int lo0
ipaddr6.6.6.6255.255.255.255
ip ospf 2 area
2intf0/0
ipaddr223.168.2.6255.255.255.0
ipospf2area2
```

```
no shut

R3

int f0/1
no
shut
ip add 223.168.2.3 255.255.255.0
```

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

```
R3

ip vrf
REDrd4:4
route-target both4:4
```

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

```
R3

int f0/1
ip vrf forwarding RED
```

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

```
R3 (config-if) # ip vrf forwarding RED
% Interface FastEthernet0/1 IP address
223.168.2.1 removed due to enabling VRF RED
```

You just need to re-apply it

```
R3
int f0/1
ip address 223.168.2.1 255.255.255.0
```

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

```
R3#sh run int  
f0/1Buildingconfiguration...  
  
Currentconfiguration:119bytes  
  
!  
  
interface  
FastEthernet0/1ipvrfforwardingRED  
ip address 223.168.2.1255.255.0  
duplex  
autospeed  
autoend
```

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

```
R3  
intf0/1  
ipospf2area2
```

Check the routes in vrf RED

```
R3  
R3#shiproutevrfRED  
RoutingTable:RED  
  
Codes:C-connected,S-static,R-RIP,M-mobile,B  
-BGP  
Gatewayoflastresortisnotset
```

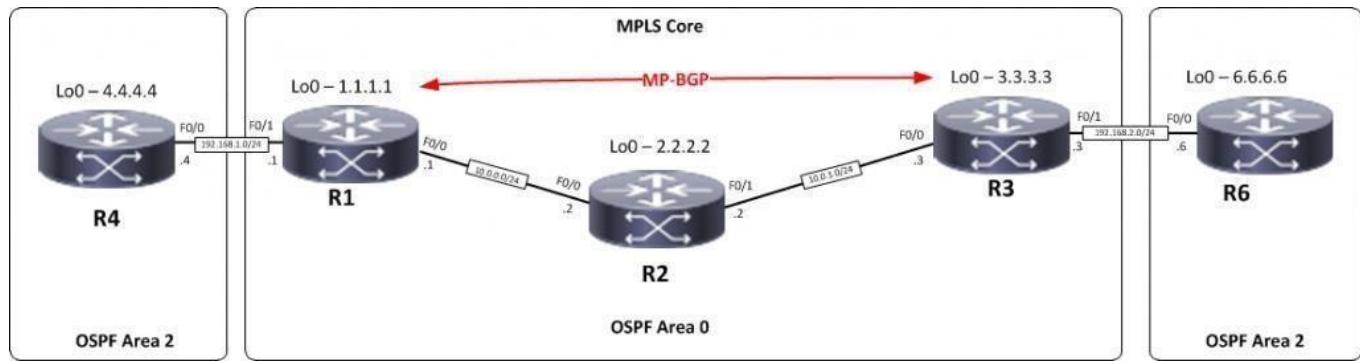
```
6.0.0.0/32 is subnetted, 1 subnets
```

```
O      6.6.6.6 [110/11] via 223.168.2.6, 00:02:44,  
FastEthernet0/1
```

```
C      223.168.2.0/24 is directly connected,  
FastEthernet0/1
```

R3#

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

```
R4#show route  
4.0.0.0/32 is subnetted, 1 subnets  
C 4.4.4.4 is directly connected, Loopback0
```

C223.168.1.0/24 is directly connected, FastEthernet0/0

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

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

Check the routes on R1

```
R1#shiproute

1.0.0.0/32 is subnetted, 1 subnets
C1.1.1.1 is directly connected, Loopback0
2.0.0.0/32 is subnetted
d, 1 subnets
O 2.2.2.2 [110/11] via 10.0.0.2, 00:01:04,
FastEthernet0/0
3.0.0.0/32 is subnetted, 1 subnets
O 3.3.3.3 [110/21] via 10.0.0.2, 00:00:54,
FastEthernet0/0
10.0.0.0/24 is subnetted, 2 subnets
C10.0.0.0 is directly connected, FastEthernet0/0
010.0.1.0 [110/20] via 10
.0.0.2, 00:00:54,
FastEthernet0/0
```

Remember we have a VRF configured on this router so this command will show routes in the global routing table (the MPLS Core) and it will not show the 223.168.1.0/24 route as that is in VRF RED – to see that we run the following command.

```
R1#shiproutevrfRED
RoutingTable:RED
4.0.0.0/32 is subnetted, 1 subnets
```

```
O 4.4.4.4[110/11]via192.168.1.4,00:02:32,  
FastEthernet0/1  
  
C223.168.1.0/24isdirectlyconnected,FastEthernet0/1
```

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

to do the following;

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

Redistribute OSPF into MP-BGP on R1

```
R1  
  
routerbgp1  
  
address-family ipv4 vrf  
REDredistributeospf2
```

Redistribute OSPF into MP-BGP on R3

```
R3  
  
routerbgp1  
  
address-family ipv4 vrf  
REDredistributeospf2
```

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

sh ip bgp vpng4 vrf RED

```
R1#sh ip bgp vpng4 vrf RED  
  
BGP table version is 9, local router ID is 1.1.1.1  
Status codes: s suppressed, d damped, h history, * valid, > best,
```

r RIB-failure, S Stale

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

Network	Next Hop	Metric	LocPrf	Weight	Path
---------	----------	--------	--------	--------	------

Route Distinguisher: 4:4 (default for vrf RED)

*> **4.4.4.4/32 223.168.1.4 11 32768 ?**

*>**i6.6.6/32 3.3.3.3 11 100 0 ?**

*> 223.168.1.0 0.0.0.0 0 32768 ?

*>i223.168.2.0 3.3.3.3 0 100 0 ?

Here we can see that 4.4.4.4 is now in the BGP table in VRF RED on R1 with a next hop of 223.168.1.4 (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 vpng4 vrf RED

BGP table version is 9, local router ID is 3.3.3.3

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

r RIB-failure, S Stale

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

Network Next Hop Metric LocPrf Weight Path Route Distinguisher: 4:4 (default for vrf RED)

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

*>**6.6.6/32 223.168.2.6 11 32768 ?**

*>i223.168.1.0 1.1.1.1 0 100 0 ?

*>223.168.2.00.0.0.0032768?

Which it is! 6.6.6.6 is now in the BGP table in VRF RED on R3 with a next hop of 223.168.2.6 (R6) and also 4.4.4 is in there as well with a next hop of 1.1.1.1 (which is the loopback of R1 – showing that it is going over the MPLS and R2 is not in the picture)

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

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

```
R1
router ospf 2 redistribute bgp 1 subnets
```

```
R3
router ospf 2 redistribute bgp 1 subnets
```

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

```
R4#show route
4.0.0.0/32 is subnetted, 1 subnets
C
    4.4.4.4 is directly connected, Loopba
    ck06.0.0.0/32 is subnetted, 1 subnets
O IA 6.6.6.6 [110/21] via 223.168.1.1, 00:01:31,
  FastEthernet0/0
```

O E2223.168.2.0/24[110/1]via223.168.1.1,00:01:31,
FastEthernet0/0

Great we have 6.6.6.6 in there

Also check the routing table on R6

```
R6#shiproute

4.0.0.0/32 is subnetted, 1 subnets
O IA4.4.4.4[110/21]via225.168.2.1,00:01:22,
FastEthernet0/0

6.0.0.0/32 is subnetted, 1 subnets
C

6.6.6.6 is directly connected, Loopback0
IA223.168.1.0/24[110
/11]via
223.168.2.1,00:01:22, FastEthernet0/0

C223.168.2.0/24 is directly connected, FastEthernet0/0
```

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

```
R4#ping6.6.6.6
Type escape sequence to abort.

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

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

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

```
R4#trace6.6.6.6
```

Type escape sequence to

abort.Tracingtherouteto6.6.6

223.168.1.120 msec 8 msec 8 msec

210.0.0.2 [MPLS:Labels17/20Exp0] 36 msec 40 msec
36 msec

223.168.2.1 [MPLS:Label20Exp0] 16 msec 40 msec 16 msec

223.168.2.6 44 msec 40 msec 56 msec R4#