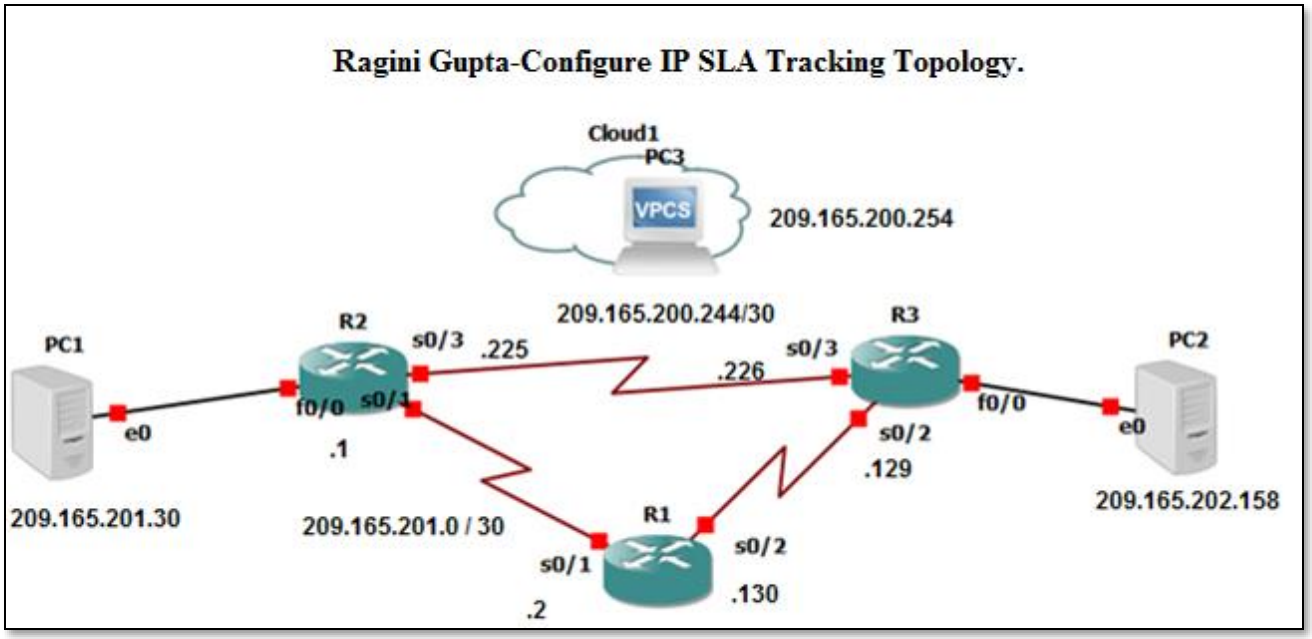
INDEX

Practical-1

Aim- Configure IP SLA Tracking Topology.



Objectives-

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

Background-

You want to experiment with the Cisco IP Service Level Agreement (SLA) feature to study how it could be of value to your organization.
At times, a link to an ISP could be operational, yet users cannot connect to any other outside Internet resources. The problem might be with the ISP or downstream from them. Although policy-based routing (PBR) can be implemented to alter path control, you will implement the Cisco IOS SLA feature to monitor these behaviors and intervene by injecting another default route to a backup ISP. To test this, you have set up a three-router topology in a lab environment.
Router R1 represents a branch office connected to two different ISPs. ISP1 is the preferred connection to the Internet, while ISP2 provides a backup link. ISP1 and ISP2 can also interconnect, and both can reach the web server.
To monitor ISP1 for failure, you will configure IP SLA probes to track the reachability to the ISP1 DNS server. If connectivity to the ISP1 server fails, the SLA probes detect the failure and alter the default static route to point to the ISP2 server.

Required Resources-

- 3 routers.
- Serial and console cables.

Step 1-Prepare the routers and configure the router hostname and interface addresses and Cable. The network as shown in the topology diagram.

A. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to them as well as the serial interfaces on R1, ISP1, and ISP2.

Router R1-

```
R1(config)#hostname R1
R1(config)#interface loopback0
R1(config-if)#description Ragini Gupta R1 LAN
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface serial0/0
R1(config-if)#description Ragini Gupta R1-->ISP1
R1(config-if)#ip address 209.165.201.2 255.255.255.252
R1(config-if)#clock rate 128000
R1(config-if)#bandwidth 128
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:03:14.807: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R1(config-if)#
*Mar  1 00:03:15.811: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
R1(config-if)#exit
R1(config)#interface serial0/1
R1(config-if)#description Ragini Gupta R1-->ISP2
R1(config-if)#
*Mar  1 00:03:43.859: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to down
R1(config-if)#ip address 209.165.202.130 255.255.255.252
R1(config-if)#bandwidth 128
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:04:15.995: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
R1(config-if)#
```

Router ISP1 (R2)-

```
R1    R2
R2(config)#hostname ISP1
ISP1(config)#interface loopback0
ISP1(config-if)#description Ragini Gupta Simulated Internet Web Server
ISP1(config-if)#ip address 209.165.200.254 255.255.255.255
ISP1(config-if)#exit
ISP1(config)#interface loopback1
ISP1(config-if)#descrip
*Mar  1 00:10:35.263: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1
, changed state to up
ISP1(config-if)#description Ragini Gupta ISP1 DNS Server
ISP1(config-if)#ip address 209.165.201.30 255.255.255.255
ISP1(config-if)#exit
ISP1(config)#interface serial0/0
ISP1(config-if)#description Ragini Gupta ISP1-->R1
ISP1(config-if)#ip address 209.165.201.1 255.255.255.252
ISP1(config-if)#bandwidth 128
ISP1(config-if)#no shutdown
ISP1(config-if)#exit
ISP1(config)#
*Mar  1 00:12:11.739: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
*Mar  1 00:12:12.763: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
ISP1(config)#interface serial 0/1
ISP1(config-if)#description Ragini Gupta ISP1-->ISP2
ISP1(config-if)#ip address 209.165.200.225 255.255.255.252
ISP1(config-if)#clock rate 128000
ISP1(config-if)#bandwidth 128
ISP1(config-if)#no shutdown
ISP1(config-if)#
```

ISP2 (R3)-

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname ISP2
ISP2(config)#interface loopback0
ISP2(config-if)#descrip
*Mar  1 00:15:24.095: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
ISP2(config-if)#description Ragini Gupta Simulated Internet Web Server
ISP2(config-if)#ip address 209.165.200.254 255.255.255.255
ISP2(config-if)#exit
ISP2(config)#interface loopback1
ISP2(config-if)#descrip
*Mar  1 00:16:19.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1
, changed state to up
ISP2(config-if)#description Ragini Gupta ISP2 DNS Server
ISP2(config-if)#ip address 209.165.202.158 255.255.255.255
ISP2(config-if)#exit
ISP2(config)#interface serial0/0
ISP2(config-if)#description Ragini Gupta ISP2-->R1
ISP2(config-if)#ip address 209.165.202.129 255.255.255.252
ISP2(config-if)#clock rate 128000
ISP2(config-if)#bandwidth 128
ISP2(config-if)#no shutdown
ISP2(config-if)#
*Mar  1 00:17:59.103: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
ISP2(config-if)#
*Mar  1 00:18:00.107: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
ISP2(config-if)#exit
```

```
ISP2(config)#interface serial0/1
ISP2(config-if)#description Ragini Gupta ISP2-->ISP1
ISP2(config-if)#ip address 209.165.200.226 255.255.255.252
ISP2(config-if)#bandwidth 128
ISP2(config-if)#no shutdown
ISP2(config-if)#
*Mar  1 00:19:19.271: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
ISP2(config-if)#
*Mar  1 00:19:20.275: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
ISP2(config-if)#
```

B. Verify the configuration by using the **show interfaces description** command. The output from router R1 is shown here as an example.
You can see that Se0/0, Se0/1, Lo0 status are changed from down to **up.**

```
R1#show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Se0/0                          up             up       Ragini Gupta R1-->ISP1
Fa0/1                          admin down     down
Se0/1                          up             up       Ragini Gupta R1-->ISP2
Se1/0                          admin down     down
Se1/1                          admin down     down
Se1/2                          admin down     down
Se1/3                          admin down     down
Lo0                            up             up       Ragini Gupta R1 LAN
R1#
```

All three interfaces should be active. Troubleshoot if necessary.

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

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

Branch (R1)-

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.201.1
R1(config)#
```

ISP1 (R2)-

```
ISP1(config)#router eigrp 1
ISP1(config-router)#network 209.165.200.224 0.0.0.3
ISP1(config-router)#network 209.165.201.0 0.0.0.31
ISP1(config-router)#no auto-summary
ISP1(config-router)#ip route 192.168.1.0 255.255.255.0 209.165.201.2
ISP1(config)#
```

ISP2 (R3)-

```
ISP2(config)#router eigrp 1
ISP2(config-router)#network 209.165.200.224 0.0.0.3
ISP2(config-router)#ne
*Mar  1 00:29:32.299: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 209.165.200.22
5 (Serial0/1) is up: new adjacency
ISP2(config-router)#network 209.165.202.128 0.0.0.31
ISP2(config-router)#no auto-summary
ISP2(config-router)#ip
*Mar  1 00:30:00.899: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 209.165.200.22
5 (Serial0/1) is resync: summary configured
ISP2(config-router)#ip route 192.168.1.0 255.255.255.0 209.165.202.130
ISP2(config)#
```

Step 2- Verify server reachability.

a. Before implementing the Cisco ISO 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. Write the following TCL script into R1.

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

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.254, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/55/80 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.30, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/52/76 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.158, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/46/84 ms
R1(tcl)#
```

b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server. Write the following Tcl script into R1.

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

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.201.1 4 msec 72 msec 40 msec
Type escape sequence to abort.
Tracing the route to 209.165.201.30

  1 209.165.201.1 52 msec 36 msec 72 msec
Type escape sequence to abort.
Tracing the route to 209.165.202.158

  1 209.165.201.1 16 msec 80 msec 52 msec
  2 209.165.200.226 112 msec 112 msec 36 msec
R1(tcl)#
```

Step 3- Configure IP SLA probes.

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

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

Note

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip sla monitor 11
R1(config-sla-monitor)#type echo protocol ipIcmpEcho 209.165.201.30
R1(config-sla-monitor-echo)#frequency 10
R1(config-sla-monitor-echo)#exit
R1(config)#ip sla monitor schedule 11 life forever start-time now
R1(config)#
R1(config)#exit
R1#
*Mar  1 01:00:16.923: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

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

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

Note

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

```
R1#show ip sla monitor configuration 11
IP SLA Monitor, Infrastructure Engine-II.
Entry number: 11
Owner:
Tag:
Type of operation to perform: echo
Target address: 209.165.201.30
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Operation frequency (seconds): 10
Next Scheduled Start Time: Start Time already passed
Group Scheduled : FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: None
Enhanced History:

 --More--
```

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

```
R1#
R1#show ip sla monitor statistics
Round trip time (RTT)    Index 11
        Latest RTT: 32 ms
Latest operation start time: *01:04:52.691 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 30
Number of failures: 0
Operation time to live: Forever


R1#
```

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip sla monitor 22
R1(config-sla-monitor)#type echo protocol ipIcmpEcho 209.165.202.158
R1(config-sla-monitor-echo)#frequency 10
R1(config-sla-monitor-echo)#exit
R1(config)#ip sla monitor schedule 22 life forever start-time now
R1(config)#
R1(config)#exit
R1#
*Mar  1 01:08:54.619: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

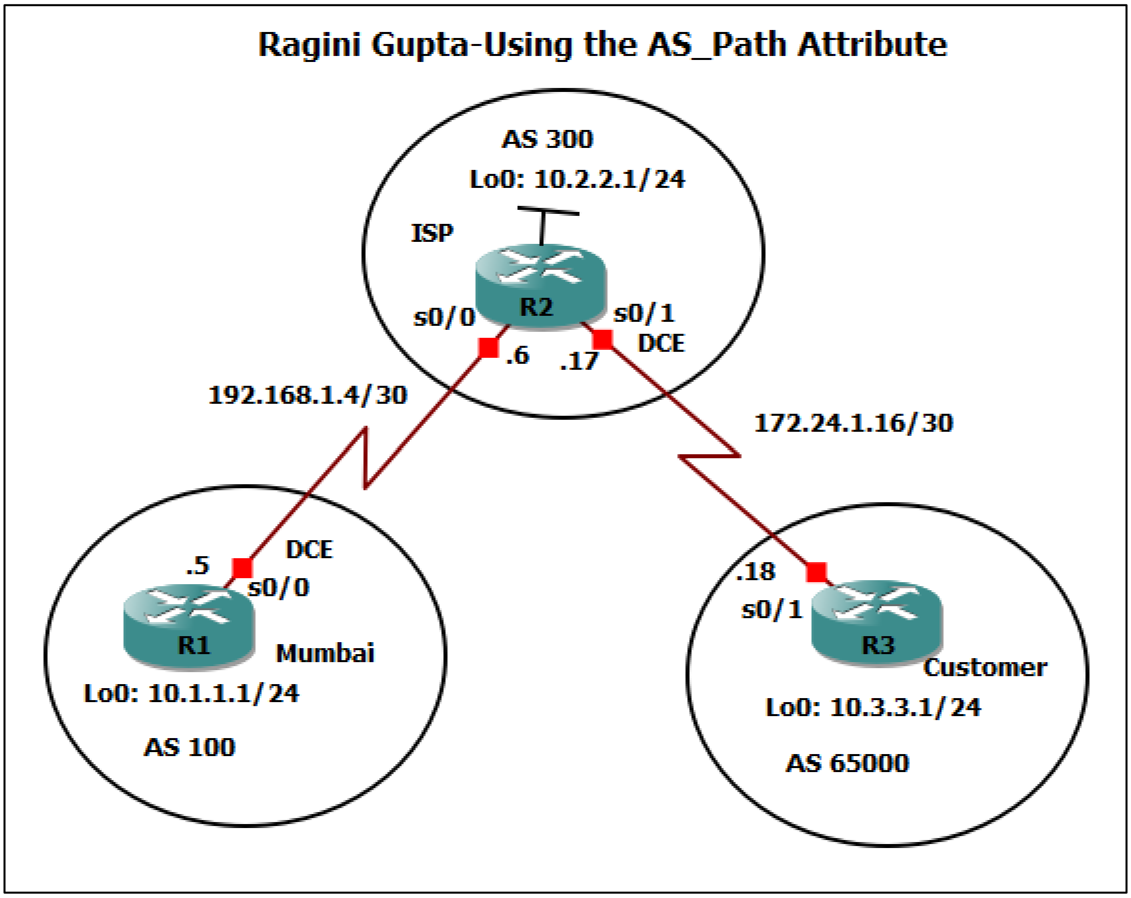
e. Verify the new probe using the show **ip sla monitor configuration** and **show ip sla monitor statistics** commands.

```
R1    R2    R3
R1#show ip sla monitor configuration 22
IP SLA Monitor, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: echo
Target address: 209.165.202.158
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Operation frequency (seconds): 10
Next Scheduled Start Time: Start Time already passed
Group Scheduled : FALSE
Life (seconds): Forever
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 1
Statistic distribution interval (milliseconds): 20
Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: None
Enhanced History:


R1#
```

```
R1#show ip sla monitor statistics 22
Round trip time (RTT)    Index 22
        Latest RTT: 128 ms
Latest operation start time: *01:15:09.803 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 39
Number of failures: 0
Operation time to live: Forever


R1#
```

Practical-2

Aim- Using the AS_PATH Attribute.



Objectives-

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

Background-

The International Travel Agency's ISP has been assigned an AS number of 300. This provider uses BGP to exchange routing information with several customer networks. Each customer network is assigned an AS number from the private range, such as AS 65000.

Configure the ISP router to remove the private AS numbers from the AS Path information of Customer. In addition, the ISP would like to prevent its customer networks from receiving route Information from International Travel Agency's AS 100. Use the AS_PATH attribute to implement this policy.

Note

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

Required Resources-

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

Step 1- Prepare the routers for the lab. Cable the network as shown in the topology diagram.

Step 2- Configure the hostname and interface addresses and use ping command to test connectivity.

Router R1 (Hostname Mumbai)-

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname Mumbai
Mumbai(config)#interface Loopback0
Mumbai(config-if)#descr
*Mar  1 00:00:50.991: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
Mumbai(config-if)#description Ragini Gupta
Mumbai(config-if)#ip address 10.1.1.1 255.255.255.0
Mumbai(config-if)#interface serial0/0
Mumbai(config-if)#ip address 192.168.1.5 255.255.255.252
Mumbai(config-if)#clock rate 128000
Mumbai(config-if)#no shutdown
Mumbai(config-if)#
*Mar  1 00:01:57.643: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Mumbai(config-if)#
*Mar  1 00:01:58.647: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
Mumbai(config-if)#
```

Router R2 (hostname ISP)-

```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname ISP
ISP(config)#interface Loopback0
ISP(config-if)#desc
*Mar  1 00:00:55.927: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
ISP(config-if)#description Ragini Gupta
ISP(config-if)#ip address 10.2.2.1 255.255.255.0
ISP(config-if)#interface serial0/0
ISP(config-if)#ip address 192.168.1.6 255.255.255.252
ISP(config-if)#no shutdown
ISP(config-if)#interface
*Mar  1 00:02:03.203: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
ISP(config-if)#interface ser
*Mar  1 00:02:04.207: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
ISP(config-if)#interface serial 0/1
ISP(config-if)#ip address 172.24.1.17 255.255.255.252
ISP(config-if)#clock
```

Router R3 (hostname Customer)-

```
Customer(config-if)#exit
Customer(config)#hostname Customer
Customer(config)#interface Loopback0
Customer(config-if)#ip ad
*Mar  1 00:02:35.739: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
Customer(config-if)#ip address 10.3.3.1 255.255.255.0
Customer(config-if)#interface serial0/1
Customer(config-if)#ip address 172.24.1.18 255.255.255.252
Customer(config-if)#no shutdown
Customer(config-if)#
```

Note

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

Step 3- Configure BGP.

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

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

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 300
ISP(config-router)#neigh
*Mar  1 00:00:32.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to down
*Mar  1 00:00:33.003: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to down
ISP(config-router)#neighbor 192.168.1.5 remote-as 100
ISP(config-router)#neighbor 172.24.1.18 remote-as 65000
ISP(config-router)#network 10.2.2.0 mask 255.255.255.0
ISP(config-router)#exit
ISP(config)#exit
ISP#
*Mar  1 00:01:36.467: %SYS-5-CONFIG_I: Configured from console by console
ISP#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
ISP#
```

```
Customer#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Customer(config)#router bgp 65000
Customer(config-router)#neighbor 172
*Mar  1 00:00:32.795: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to down
Customer(config-router)#neighbor 172.24.1.17 remote-as 300
Customer(config-router)#network 10.3.3.0 mask 255.255.255.0
Customer(config-router)#
```

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

```
ISP#show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 65000, external link
  BGP version 4, remote router ID 10.3.3.1
  BGP state = Established, up for 00:00:18
  Last read 00:00:18, last write 00:00:18, 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          1
    Keepalives:             3          3
    Route Refresh:          0          0
    Total:                  6          5
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 3/0
 Output queue size : 0
  Index 1, Offset 0, Mask 0x2
 --More--
```

Step 4- Remove the private AS.

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

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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 192.168.1.6, 00:01:09
B       10.2.2.0 [20/0] via 192.168.1.6, 00:02:29
C       10.1.1.0 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0
Mumbai#
```

b. Ping the 10.3.3.1 address from Mumbai. Why does this fail?

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

```
Mumbai#ping
Protocol [ip]:
Target IP address: 10.3.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/102/116 ms
Mumbai#
```

Note

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

```
Mumbai#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 = 76/104/116 ms
Mumbai#
```

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

```
Mumbai#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      192.168.1.6              0             0 300 i
*> 10.3.3.0/24      192.168.1.6                            0 300 65000 i
Mumbai#
```

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

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 300
ISP(config-router)#neighbor 192.168.1.5 remove-private-as
ISP(config-router)#
```

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

Note

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

Does Mumbai still have a route to 10.3.3.0?

Mumbai should be able to ping 10.3.3.1 using its loopback O interface as the source of the ping.

```
Mumbai#ping 10.3.3.1 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 80/109/148 ms
Mumbai#
```

g. Now check the BGP table on Mumbai. The AS_PATH to the 10.3.3.0 network should be AS 300.

It no longer has the private AS in the path.

```
Mumbai#show ip bgp
BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              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      192.168.1.6              0             0 300 i
*> 10.3.3.0/24      192.168.1.6                            0 300 i
Mumbai#
```

Step 5- Use the AS_PATH attribute to filter routes.

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

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

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

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

The first command uses the ^ character to indicate that the AS path must begin with the given number 100. The $ character indicates that the AS_PATH attribute must also end with 100.

Essentially, this statement matches only paths that are sourced from AS 100. Other paths, which might include AS 100 along the way, will not match this list.

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

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

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

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

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

Note

To force the local router to resend its BGP table, a less disruptive option is to use the **clear ip bgp * out** or **clear ip bgp * soft** command (the second command performs both outgoing and incoming route resync).

```
ISP#show ip route
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

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected, Serial0/1
     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 172.24.1.18, 00:19:34
C       10.2.2.0 is directly connected, Loopback0
B       10.1.1.0 [20/0] via 192.168.1.5, 00:20:24
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0
ISP#
```

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

```
Customer#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

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected, Serial0/1
     10.0.0.0/24 is subnetted, 2 subnets
C       10.3.3.0 is directly connected, Loopback0
B       10.2.2.0 [20/0] via 172.24.1.17, 00:20:36
Customer#
```

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

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

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      192.168.1.5              0             0 100 i
ISP#
```

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

Run the following TCI script on all routers to verify whether there is connectivity. All pings from ISP should be successful.

Mumbai should not be able to ping the Customer loopback 10.3.3.1 or the WAN link 172.24.1.16/30.

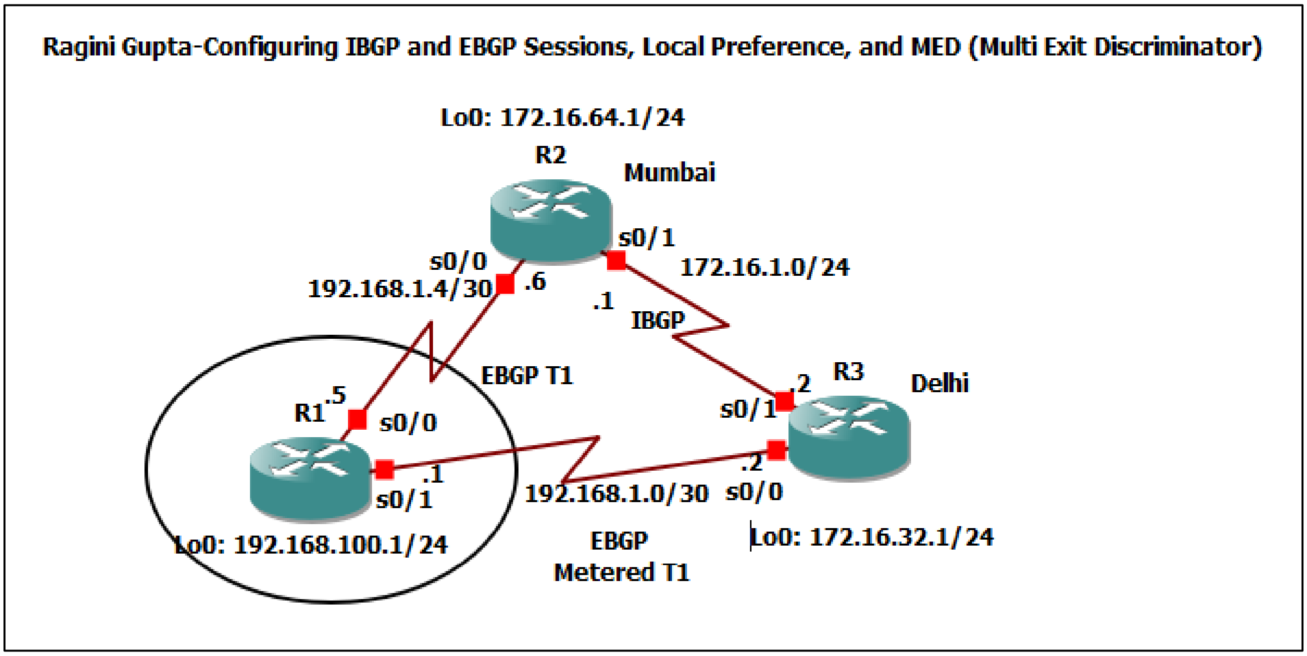
Customer should not be able to ping the Mumbai loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

```
ISP(tcl)#tclsh

ISP(tcl)#foreach address { 10.1.1.1
+>10.2.2.1
+>10.3.3.1
+>192.168.1.5
+>192.168.1.6
+>172.24.1.17
+>172.24.1.18
+>} {
+>ping $address }
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/57/80 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 = 1/2/4 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 = 28/63/132 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/52/76 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 104/125/176 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 = 76/108/156 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 = 36/66/100 ms
ISP(tcl)#
```

Practical-3

Aim- Configuring IBGP and EBGP Sessions, Local Preference, and MED (Multi Exit Discriminator).



Objectives-

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

Background-

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

Note

This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features.

Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources-

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

Step 1-Prepare the routers for the lab. Cable the network as shown in the topology diagram.

Step 2-Configure the hostname and interface addresses and use ping to test the connectivity between the directly connected routers.

Both Mumbai and Delhi routers should be able to ping each other and their local ISP serial link IP address. The ISP router cannot reach the segment between Mumbai and Delhi.

Router R1 (hostname ISP)-

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname ISP
ISP(config)#interface Loopback0
ISP(config-if)#
*Mar  1 00:02:34.447: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
ISP(config-if)#description Ragini Gupta
ISP(config-if)#ip address 192.168.100.1 255.255.255.0
ISP(config-if)#interface Serial 0/0
ISP(config-if)#ip address 192.168.1.5 255.255.255.252
ISP(config-if)#clock rate 128000
ISP(config-if)#no shutdown
ISP(config-if)#
*Mar  1 00:03:57.483: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
ISP(config-if)#
```

```
ISP(config-if)#interface Serial0/1
ISP(config-if)#ip address 172.16.1.1 255.255.255.0
ISP(config-if)#clock rate 128000
ISP(config-if)#no shutdown
ISP(config-if)#
```

Router R2 (hostname Mumbai)-

```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#hostname Mumbai
Mumbai(config)#interface Loopback0
Mumbai(config-if)#de
*Mar  1 00:00:37.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
Mumbai(config-if)#description Ragini Gupta
Mumbai(config-if)#ip address 172.16.64.1 255.255.255.0
Mumbai(config-if)#interface Serial 0/0
Mumbai(config-if)#ip address 192.168.1.6 255.255.255.252
Mumbai(config-if)#no shutdown
Mumbai(config-if)#
```

```
Mumbai(config-if)#exit
Mumbai(config)#interface Serial0/1
Mumbai(config-if)#ip address 172.16.1.1 255.255.255.0
Mumbai(config-if)#clock rate 128000
Mumbai(config-if)#no shutdown
Mumbai(config-if)#
*Mar  1 00:03:13.531: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
Mumbai(config-if)#
*Mar  1 00:03:14.535: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
Mumbai(config-if)#exit
Mumbai(config)#exit
Mumbai#cop
*Mar  1 00:03:24.479: %SYS-5-CONFIG_I: Configured from console by console
Mumbai#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Mumbai#
*Mar  1 00:03:42.887: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to down
Mumbai#
```

Router R3 (hostname Delhi)-

```
R3(config)#hostname Delhi
Delhi(config)#interface Loopback0
Delhi(config-if)#de
*Mar  1 00:00:57.659: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
Delhi(config-if)#description Ragini Gupta
Delhi(config-if)#ip address 172.16.32.1 255.255.255.0
Delhi(config-if)#interface Serial0/0
Delhi(config-if)#ip address 192.168.1.2 255.255.255.252
Delhi(config-if)#clock rate 128000
Delhi(config-if)#no shutdown
Delhi(config-if)#inter
*Mar  1 00:02:00.587: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
Delhi(config-if)#interface
*Mar  1 00:02:01.591: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
Delhi(config-if)#interface Serial0/1
Delhi(config-if)#ip address 172.16.1.2 255.255.2
*Mar  1 00:02:22.727: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to down
Delhi(config-if)#ip address 172.16.1.2 255.255.255.0
Delhi(config-if)#no shutdown
Delhi(config-if)#
*Mar  1 00:02:32.151: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
*Mar  1 00:02:33.151: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
Delhi(config-if)#
```

Step 3- Configure EIGRP between the Mumbai and Delhi routers.

```
Mumbai(config)#router eigrp 64512
Mumbai(config-router)#no auto-summary
Mumbai(config-router)#network 172.16.0.0
Mumbai(config-router)#
```

```
Delhi(config)#router eigrp 64512
Delhi(config-router)#no auto-summary
Delhi(config-router)#network 172.16.0.0
Delhi(config-router)#
```

Step 4-Configure IBGP and verify BGP neighbors.

a. Configure IBGP between the Mumbai and Delhi routers. On the Mumbai router, enter the following configuration.

```
Mumbai#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Mumbai(config)#router bgp 64512
Mumbai(config-router)#neighbor 172.16.32.1 remote-as 64512
Mumbai(config-router)#neighbor 172.16.32.1 update-source lo0
Mumbai(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 Loopback0 as the source IP address for all BGP messages sent to that neighbor.

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

```
Delhi#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Delhi(config)#router bgp 64512
Delhi(config-router)#nei
*Mar  1 00:00:32.819: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to down
*Mar  1 00:00:32.843: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to down
Delhi(config-router)#neighbor 172.16.64.1 remote-as 64512
Delhi(config-router)#neighbor 172.16.64.1 update-source lo0
Delhi(config-router)#
```

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

```
Delhi#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 00:02:54
  Last read 00:00:54, last write 00:00:54, 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:             4          4
    Route Refresh:          0          0
    Total:                  5          5
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 1/0
 Output queue size : 0
 --More--
```

The link between Mumbai and Delhi should be identified as an internal link, as shown in the output.

Step 5- Configure EBGP and verify BGP neighbors.

a. Configure ISP to run EBGP with Mumbai and Delhi. Enter the following commands on ISP.

```
ISP#config t
Enter configuration commands, one per line.  End with CNTL/Z.
ISP(config)#router bgp 200
ISP(config-router)#neighbor 192.168.1.6 remote-as 64512
ISP(config-router)#neighbor 192.168.1.2 remote-as 64512
ISP(config-router)#network 192.168.100.0
ISP(config-router)#
```

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

b. Configure Mumbai as an EBGP peer to ISP.

```
Mumbai#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Mumbai(config)#ip route 172.16
*Mar  1 00:00:33.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to down
*Mar  1 00:00:33.359: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to down
Mumbai(config)#ip route 172.16.0.0 255.255.0.0 null0
Mumbai(config)#router bgp 64512
Mumbai(config-router)#neighbor 192.168.1.5 remote-as 200
Mumbai(config-router)#network 172.16.0.0
Mumbai(config-router)#
```

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

```
Mumbai#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 00:07:42
  Last read 00:00:42, last write 00:00:40, 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:                1          0
    Keepalives:             9          9
    Route Refresh:          0          0
    Total:                 11         10
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  BGP table version 2, neighbor version 2/0
 Output queue size : 0
 --More--
```

d. Configure Delhi as an EBGP peer to ISP.

```
Delhi#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Delhi(config)#ip route 172.16.0.0 255.255.0.0 null 0
Delhi(config)#router bgp 64512
Delhi(config-router)#neighbor 192.168.1.1 remote-as 200
Delhi(config-router)#network 172.16.0.0
```

Step 6-View BGP summary output.

In Step 5, the show ip bgp neighbors command was used to verify that Mumbai and ISP had reached the established state. A useful alternative command is show ip bgp summary. The output should be similar to the following.

```
Delhi#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 234 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
BGP using 1086 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/PfxRc
d
172.16.64.1     4 64512      18      18        5    0    0 00:13:17        2
192.168.1.1     4   200       6       5        5    0    0 00:00:50        1
Delhi#
```

Step 7- Verify which path the traffic takes.

a. Clear the IP BGP conversation with the clear ip bgp * command on ISP. Wait for the conversations to re-establish with both Mumbai and Delhi router.
b. Test whether ISP can ping the loopback 0 address of 172.16.64.1 on Mumbai and the serial link between Mumbai and Delhi,172.16.1.1.
c. Now ping from ISP to the loopback 0 address of 172.16.32.1 on Delhi and the serial link between Mumbai and Delhi, 172.16.1.2.

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

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

```
ISP#show ip bgp
BGP table version is 3, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

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

Notice that ISP has two valid routes to the 172.16.0.0 network, as indicated by the *. However, the link to Delhi has been selected as the best path. Why did the ISP prefer the link to Delhi over Mumbai?

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

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

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

```
ISP#ping 172.16.1.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/48/76 ms
ISP#
```

```
ISP#ping 172.16.32.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/67/144 ms
ISP#
```

```
ISP#ping 172.16.1.2  source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 80/108/136 ms
ISP#
```

```
ISP#ping 172.16.64.1  source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/52/104 ms
ISP#
```

```
ISP#ping
Protocol [ip]:
Target IP address: 172.16.64.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.100.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/43/68 ms
ISP#
```

Complete reachability has been demonstrated between the ISP router and both Mumbai and Delhi.

Step 8-Configure the BGP next-hop-self feature.

Mumbai is unaware of the link between ISP and Delhi, and Delhi is unaware of the link between ISP and Mumbai. Before ISP can successfully ping all the internal serial interfaces of AS 64512, these serial links should be advertised via BGP on the ISP router. This can also be resolved via EIGRP on both Mumbai and Delhi router. The preferred method is for ISP to advertise these links.

a. Issue the following commands on the ISP router.

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

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

```
ISP#show ip bgp
BGP table version is 5, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

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

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

```
Delhi#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, 00:28:50, Serial0/1
     192.168.1.0/30 is subnetted, 2 subnets
C       192.168.1.0 is directly connected, Serial0/0
B       192.168.1.4 [20/0] via 192.168.1.1, 00:03:16
B    192.168.100.0/24 [20/0] via 192.168.1.1, 00:13:11
Delhi#
```

The next issue to consider is BGP policy routing between autonomous systems. The next-hop attribute of a route in a different AS is set to the IP address of the border router in the next AS toward the destination, and this attribute is not modified by default when advertising this route through IBGP.

Therefore, for all IBGP peers, it is either necessary to know the route to that border router (in a different neighboring AS), or our own border router needs to advertise the foreign routes using the next-hop self-feature, overriding the next-hop address with its own IP address.

The Delhi router is passing a policy to Mumbai and vice versa. The policy for routing from AS 64512 to AS 200 is to forward packets to the 192.168.1.1 interface. Mumbai has a similar yet opposite policy: it forwards requests to the 192.168.1.5 interface.

If either WAN link fails, it is critical that the opposite router become a valid gateway. This is achieved if the next-hop-self command is configured on Mumbai and Delhi.

d. View the output before the next-hop-self command is issued.

```
Delhi#show ip bgp
BGP table version is 8, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
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
r> 192.168.1.0/30   192.168.1.1              0             0 200 i
r i                 192.168.1.5              0    100      0 200 i
*> 192.168.1.4/30   192.168.1.1              0             0 200 i
* i                 192.168.1.5              0    100      0 200 i
*> 192.168.100.0    192.168.1.1              0             0 200 i
* i                 192.168.1.5              0    100      0 200 i
Delhi#
```

Issue the next-hop-self command on Mumbai and Delhi.

```
Mumbai#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Mumbai(config)#router bgp 64512
Mumbai(config-router)#neighbor 172.16.32.1 next-hop-self
Mumbai(config-router)#router bgp 64512
Mumbai(config-router)#neighbor 172.16.64.1 next-hop-self
% Cannot configure the local system as neighbor
Mumbai(config-router)#
```

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

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

g. After the routers have returned to established BGP speakers, issue the show ip bgp command to validate that the next hop has also been corrected.

```
Delhi#clear ip bgp * soft
Delhi#show ip bgp
BGP table version is 8, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
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
r> 192.168.1.0/30   192.168.1.1              0             0 200 i
r i                 172.16.64.1              0    100      0 200 i
*> 192.168.1.4/30   192.168.1.1              0             0 200 i
* i                 172.16.64.1              0    100      0 200 i
*> 192.168.100.0    192.168.1.1              0             0 200 i
* i                 172.16.64.1              0    100      0 200 i
Delhi#
```

Step 9- Set BGP local preference.

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

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

```
Mumbai#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Mumbai(config)#route-map PRIMARY_T1_IN permit 10
Mumbai(config-route-map)#set local-preference 150
Mumbai(config-route-map)#exit
Mumbai(config)#router bgp 64512
Mumbai(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_IN in
Mumbai(config-router)#
```

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

b. 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 Mumbai and Delhi.

```
Mumbai#clear ip bgp * soft
Mumbai#
Mumbai#show ip bgp
BGP table version is 9, 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
*> 192.168.1.0/30   192.168.1.5              0    150      0 200 i
r> 192.168.1.4/30   192.168.1.5              0    150      0 200 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
Mumbai#
```

```
Delhi#clear ip bgp * soft
Delhi#show ip bgp
BGP table version is 11, 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
r  192.168.1.0/30   192.168.1.1              0    125      0 200 i
r>i                 172.16.64.1              0    150      0 200 i
*  192.168.1.4/30   192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
*  192.168.100.0    192.168.1.1              0    125      0 200 i
*>i                 172.16.64.1              0    150      0 200 i
Delhi#
```

This now indicates that routing to the loopback segment for ISP 192.168.100.0 /24 can be reached only through the link common to Mumbai and ISP.

Step 10- Set BGP MED (Multi Exit Discriminator).

How will traffic return from network 192.168.100.0 /24? Will it be routed through Mumbai or Delhi?

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

a. Use an extended **ping** command in this situation. Specify the record option and compare your output to the following.

```
Delhi#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]: 2
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 2, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
```

```
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)

Reply to request 0 (84 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
```

```
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 1 (116 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
```

```
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Success rate is 100 percent (2/2), round-trip min/avg/max = 84/100/116 ms
Delhi#
```

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

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

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

```
Mumbai#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Mumbai(config)#route-map PRIMARY_T1_MED_OUT permit 10
Mumbai(config-route-map)#set Metric 50
Mumbai(config-route-map)#exit
Mumbai(config)#router bgp 64512
Mumbai(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out
Mumbai(config-router)#
Mumbai(config-router)#exit
Mumbai(config)#
```

```
Delhi#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Delhi(config)#route-map SECONDARY_T1_MED_OUT permit 10
Delhi(config-route-map)#set Metric 75
Delhi(config-route-map)#exit
Delhi(config)#router bgp 64512
Delhi(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
Delhi(config-router)#
```

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

```
Mumbai#clear ip bgp * soft
Mumbai#show ip bgp
BGP table version is 9, 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
*> 192.168.1.0/30   192.168.1.5              0    150      0 200 i
r> 192.168.1.4/30   192.168.1.5              0    150      0 200 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
Mumbai#
```

d. Reissue an extended ping command with the record command.

```
Delhi#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]: 2
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 2, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
```

```
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)

Reply to request 0 (100 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 1 (104 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
```

```
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Success rate is 100 percent (2/2), round-trip min/avg/max = 100/102/104 ms
Delhi#
```

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

There might not be a chance to use Telnet to the ISP router and to issue the show ip bgp command. However, the command on the opposite side of the newly configured policy MED is clear, showing that the lower value is considered best. The ISP now prefers the route with the lower MED value to AS 64512. This is just opposite from the local-preference command configured earlier.

```
ISP#show ip bgp
BGP table version is 7, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.2             75             0 64512 i
*>                  192.168.1.6             50             0 64512 i
*> 192.168.1.0/30   0.0.0.0                  0         32768 i
*> 192.168.1.4/30   0.0.0.0                  0         32768 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
ISP#
```

Step 11- Establish a default network.

The final step is to establish a default route that uses a policy statement that adjusts to changes in the network. Configure both Mumbai and Delhi to use the 192.168.100.0 /24 network as the default network. The following steps configure the Mumbai router. Do the same on the Delhi router.

a. View the routing table prior to issuing the ip default-network statement.

```
Mumbai#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
D       172.16.32.0/24 [90/2297856] via 172.16.1.2, 01:18:24, 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
     192.168.1.0/30 is subnetted, 2 subnets
B       192.168.1.0 [20/0] via 192.168.1.5, 00:30:44
C       192.168.1.4 is directly connected, Serial0/0
B    192.168.100.0/24 [20/0] via 192.168.1.5, 01:07:23
Mumbai#
```

```
Mumbai#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Mumbai(config)#ip default-network 192.168.100.0
Mumbai(config)#
```

Note

The above command works well only with remotely-learned classful networks. It should not be used with classless networks. An alternative to using the ip default-network command on Mumbai is issuing the neighbor X.X.X.X default-originate configuration on the ISP router.

b. View the routing table after issuing the ip default-network statement.

```
Mumbai#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 192.168.1.5 to network 192.168.100.0

     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, 01:21:35, 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
     192.168.1.0/30 is subnetted, 2 subnets
B       192.168.1.0 [20/0] via 192.168.1.5, 00:33:55
C       192.168.1.4 is directly connected, Serial0/0
B*   192.168.100.0/24 [20/0] via 192.168.1.5, 01:10:39
Mumbai#
```

What would be required to add a future T3 link on Delhi and for it to have preference for incoming and outgoing traffic?

A newly added route is as easy as adding another route map for local preference with a value of 175 and a route map referencing a MED (metric) value of 35.

Note

By default, the MED is compared only when the route is being received from the same neighboring AS, although advertised by different border routers. The nondefault behavior of comparing the MED regardless of the AS advertising the route can be activated using the bgp always-compare- med command, however, the results of this command have to be carefully considered.

Note Because the MED is an optional attribute, it might not be present in BGP updates. RFC 4271 requires that a missing MED is equivalent to having the MED set to 0. However, a missing MED can also be considered to be the worst possible MED, which is activated using the bgp best path med missing- as-worst command.

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

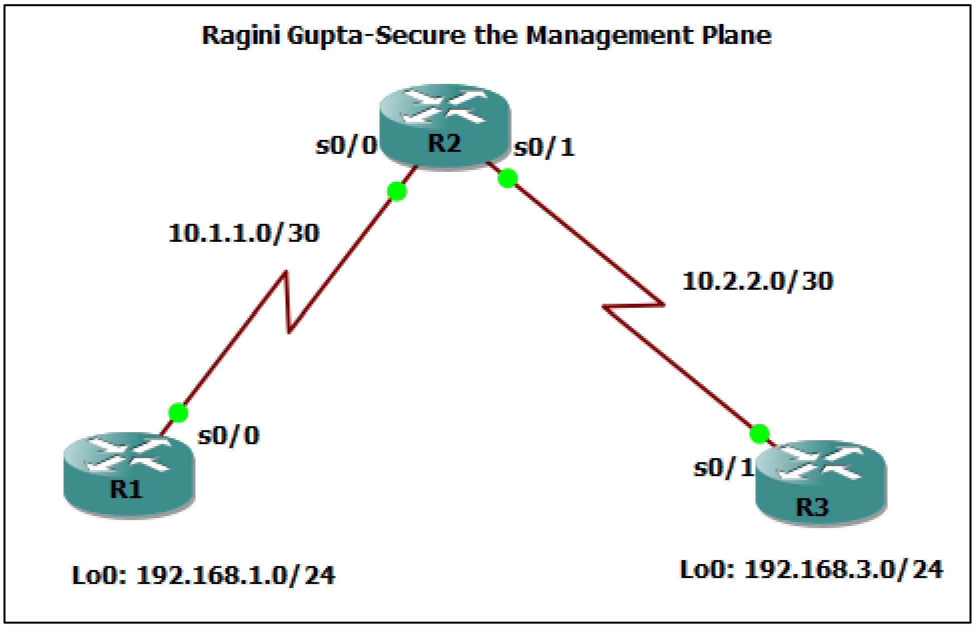
```
ISP#tclsh
ISP(tcl)#foreach address { 192.168.100.1
+>172.16.64.1
+>172.16.32.1
+>192.168.1.1
+>192.168.1.2
+>192.168.1.5
+>192.168.1.6
+>172.16.1.1
+>172.16.1.2
+>} {
+>ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, 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 172.16.64.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/48/72 ms
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 = 40/133/220 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/106/136 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/71/120 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/53/88 ms
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 = 44/61/84 ms
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 = 36/123/172 ms
ISP(tcl)#
```

Practical-4

Aim-Secure the Management Plane.



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.

R1 router-

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#interface loopback0
R1(config-if)#description R1 LAN Ragini
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#!
R1(config)#interface serial0/0
R1(config-if)#description R1-->R2
R1(config-if)#ip address 10.1.1.1 255.255.255.252
R1(config-if)#clock rate 128000
R1(config-if)#no shutdown
R1(config-if)#exit
*Mar  1 00:05:24.143: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R1(config)#
*Mar  1 00:05:25.147: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
R1(config)#end
R1#
*Mar  1 00:05:33.451: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

R2 Router-

```
R2(config)#hostname R2
R2(config)#interface serial0/0
R2(config-if)#description R2-->R1
R2(config-if)#ip address 10.1.1.2 255.255.255.252
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
*Mar  1 00:29:07.995: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R2(config)#
*Mar  1 00:29:08.999: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
R2(config)#interface serial0/1
R2(config-if)#description R2-->R3
R2(config-if)#ip address 10.2.2.1 255.255.255.252
R2(config-if)#clock rate 128000
R2(config-if)#no shutdown
R2(config-if)#exit
R2(config)#
```

R3 Router-

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname R3
R3(config)#interface loopback0
R3(config-if)#descr
*Mar  1 00:10:34.807: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R3(config-if)#description R3 LAN Ragini
R3(config-if)#ip address 192.168.3.1 255.255.255.0
R3(config-if)#exit
R3(config)#interface serial0/1
R3(config-if)#description R3-->R2
R3(config-if)#ip address 10.2.2.2 255.255.255.252
R3(config-if)#no shutdown
R3(config-if)#ex
*Mar  1 00:11:59.583: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
R3(config-if)#exit
*Mar  1 00:12:00.587: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
R3(config-if)#exit
R3(config)#
```

Step 2- Configure static routes.

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

```
R1#config 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
R1(config)#
```

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

```
R3#config 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)#
```

c. On R2, configure two static routes.

```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.2
R2(config)#
```

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

```
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#10.1.1.1
+>(tcl)#10.1.1.2
+>(tcl)#10.2.2.1
+>(tcl)#10.2.2.2
+>(tcl)#192.168.3.1
+>(tcl)# }{ping $address }
extra characters after close-brace
R1(tcl)#tclsh

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

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 76/106/120 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 = 28/51/76 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 = 40/58/80 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 = 36/124/180 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/111/168 ms
R1(tcl)#
```

Step 3- Secure management access.

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#security password min-length 10
R1(config)#
```

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

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

c. Configure a console password and enable login for routers. For additional security, the exec-timeout command causes the line to log out after 5 minutes of inactivity. The logging synchronous command prevents console messages from interrupting command entry. Note: To avoid repetitive logins during this lab, the exec-timeout command can be set to 0 0, which prevents it from expiring. However, this is not considered a good security practice.

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

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

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

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

```
R1(config)#line aux 0
R1(config-line)#no exec
R1(config-line)#end
R1#
*Mar  1 00:44:44.039: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

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

```
R1#show run
Building configuration...

Current configuration : 1516 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
security passwords min-length 10
enable secret 5 $1$eE9/$e2CbTpLNIIzDuPUELacPw1
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
```

```
ip tcp synwait-time 5
!
!
!
interface Loopback0
 description R1 LAN Ragini
 ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0
 description R1-->R2
 ip address 10.1.1.1 255.255.255.252
 clock rate 128000
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 --More--
```

```
!
line con 0
 exec-timeout 5 0
 privilege level 15
 password ciscoconpass
 logging synchronous
 login
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 no exec
line vty 0 4
 exec-timeout 5 0
 password ciscovtypass
 login
!
!
end
```

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#service password-encryption
R1(config)#
```

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

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

```
R1#show run
Building configuration...

Current configuration : 1545 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
security passwords min-length 10
enable secret 5 $1$eE9/$e2CbTpLNIIzDuPUELacPw1
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
 --More--
```

```
!
line con 0
 exec-timeout 5 0
 privilege level 15
 password 7 00071A150754080901314D5D1A
 logging synchronous
 login
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 no exec
line vty 0 4
 exec-timeout 5 0
 password 7 1511021F07253D303123343100
 login
!
!
end
 --More--
```

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#banner motd $Unauthorized accedd strictly prohibited!$
R1(config)#exit
R1#
*Mar  1 00:53:00.367: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

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

```
R1#show run
Building configuration...

Current configuration : 1600 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R1
!
boot-start-marker
```

```
!
banner motd ^CUnauthorized accedd strictly prohibited!^C
!
line con 0
 exec-timeout 5 0
 privilege level 15
 password 7 00071A150754080901314D5D1A
 logging synchronous
 login
```

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

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

```
R3(config)#
R3(config)#security password min-length 10
R3(config)#enable secret class12345
R3(config)#line console 0
R3(config-line)#password cisconopass
R3(config-line)#exec-timeout 5 0
R3(config-line)#login
R3(config-line)#logging synchronous
R3(config-line)#exit
R3(config)#line vty 0 4
R3(config-line)#password ciscovtypass
R3(config-line)#exec-timeout 5 0
R3(config-line)#login
R3(config-line)#exit
R3(config)#line aux 0
R3(config-line)#no exec
R3(config-line)#end
R3#
*Mar  1 01:02:03.075: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

```
R3#show run
Building configuration...

Current configuration : 1516 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
security passwords min-length 10
enable secret 5 $1$gTEl$/PWFpfUtS8IO5ysBZJxuA0
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
 --More--
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#service password-encryption
R3(config)#
```

```
R3#show run
Building configuration...

Current configuration : 1544 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname R3
!
boot-start-marker
boot-end-marker
!
security passwords min-length 10
enable secret 5 $1$gTEl$/PWFpfUtS8IO5ysBZJxuA0
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
```

```
!
line con 0
 exec-timeout 5 0
 privilege level 15
 password 7 110A1016141D0503142B3837
 logging synchronous
 login
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 no exec
line vty 0 4
 exec-timeout 5 0
 password 7 070C285F4D060F110E020A1F17
 login
!
!
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#banner motd $Unauthorized access strictly prohibited$
R3(config)#exit
R3#
*Mar  1 01:06:51.999: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

Step 4-Configure enhanced username password security.

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

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

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

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

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

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

```
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#end
R1#
*Mar  1 01:12:35.871: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

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

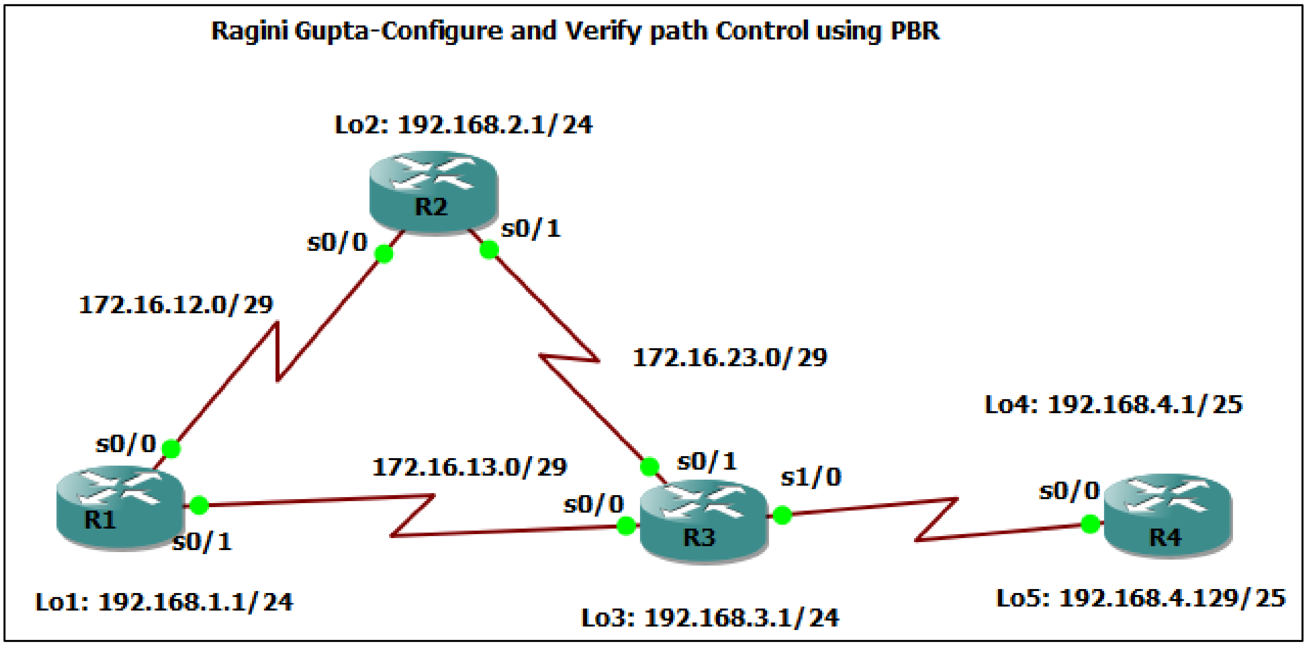
```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#username RG-ADMIN secret class12345
R3(config)#username ADMIN secret class54321
R3(config)#line console 0
R3(config-line)#login local
R3(config-line)#exit
R3(config)#line vty 0 4
R3(config-line)#login local
R3(config-line)#end
R3#
*Mar  1 01:14:40.475: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

e. To verify the configuration, telnet to R3 from R1 and login using the ADMIN local database account. (Username-ADMIN, Password-class54321)

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

User Access Verification

Username: ADMIN
Password:
R3>
```

Practical-5

Aim-Configuring and Verifying Path control using PBR.



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

Step 2- Configure router hostname and interface addresses.

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

R1 Router-

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#interface Lo1
R1(config-if)#de
*Mar  1 00:06:03.623: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1
, changed state to up
R1(config-if)#description R1 LAN Ragini Gupta
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#
R1(config-if)#exit
R1(config)#interface serial0/0
R1(config-if)#description Ragini Gupta R1-->R2
R1(config-if)#ip address 172.16.12.1 255.255.255.248
R1(config-if)#clock rate 128000
R1(config-if)#bandwidth 128
```

```
R1(config-if)#no shutdown
R1(config-if)#
*Mar  1 00:08:15.843: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R1(config-if)#
*Mar  1 00:08:16.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
R1(config-if)#exit
R1(config)#interface serial0/1
R1(config-if)#description Ragini Gupta R1-
*Mar  1 00:08:43.855: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to down
R1(config-if)#description Ragini Gupta R1-->R3
R1(config-if)#ip address 172.16.13.1 255.255.255.248
R1(config-if)#bandwidth 64
R1(config-if)#no shutdown
R1(config-if)#end
R1#
*Mar  1 00:09:31.275: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
R1#
*Mar  1 00:09:32.323: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:09:32.323: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
R1#
*Mar  1 00:09:53.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to down
R1#
```

R2 Router-

```
R2(config)#hostname R2
R2(config)#interface lo2
R2(config-if)#description Ragini Gupta R2 LAN
R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#
R2(config-if)#exit
R2(config)#interface serial0/0
R2(config-if)#description Ragini Gupta R2-->R1
R2(config-if)#ip address 172.16.12.2 255.255.255.248
R2(config-if)#bandwidth 128
R2(config-if)#no shutdown
R2(config-if)#
*Mar  1 00:40:06.867: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R2(config-if)#
*Mar  1 00:40:07.871: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state
R2(config-if)#exit
R2(config)#interface serial0/1
R2(config-if)#description Ragini Gupta R2-->R3
R2(config-if)#ip address 172.16.23.2 255.255.255.248
R2(config-if)#clock rate 128000
R2(config-if)#bandwidth 128
R2(config-if)#no shutdown
R2(config-if)#end
R2#
*Mar  1 00:41:29.911: %SYS-5-CONFIG_I: Configured from console by console
R2#
```

R3 Router-

```
R1    R2    R3
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface serial0/0
R3(config-if)#description Ragini Gupta R3-->R1
R3(config-if)#ip address 172.16.13.3 255.255.255.248
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
R3(config-if)#
*Mar  1 00:17:50.367: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R3(config-if)#
*Mar  1 00:17:51.371: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
R3(config-if)#exit
R3(config)#interface serial0/1
R3(config-if)#description Ragini Gupta R3-->R2
R3(config-if)#ip address 172.16.23.3 255.255.255.248
R3(config-if)#bandwidth 128
R3(config-if)#no shutdown
R3(config-if)#
*Mar  1 00:18:58.639: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
R3(config-if)#
*Mar  1 00:18:59.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
R3(config-if)#exit
R3(config)#interface serial1/0
R3(config-if)#description Ragini Gupta R3-->R4
R3(config-if)#ip address 172.16.34.3 255.255.255.248
R3(config-if)#clock rate 64000
```

```
R3(config-if)#no shutdown
R3(config-if)#
*Mar  1 00:18:58.639: %LINK-3-UPDOWN: Interface Serial0/1, changed state to up
R3(config-if)#
*Mar  1 00:18:59.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1
, changed state to up
R3(config-if)#exit
R3(config)#interface serial1/0
R3(config-if)#description Ragini Gupta R3-->R4
R3(config-if)#ip address 172.16.34.3 255.255.255.248
R3(config-if)#clock rate 64000
R3(config-if)#bandwidth 64
R3(config-if)#no shutdown
R3(config-if)#
*Mar  1 00:20:16.291: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
R3(config-if)#end
*Mar  1 00:20:17.295: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0
, changed state to up
R3(config-if)#end
R3#
*Mar  1 00:20:19.235: %SYS-5-CONFIG_I: Configured from console by console
R3#
*Mar  1 00:20:44.095: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0
, changed state to down
R3#
```

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface lo3
R3(config-if)#description Ragini Gupta R3 LAN
R3(config-if)#ip address 192.168.3.1 255.255.255.0
R3(config-if)#
```

R4 Router-

```
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#hostname R4
R4(config)#interface lo4
R4(config-if)#descript
*Mar  1 00:22:49.819: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback4
, changed state to up
R4(config-if)#description Ragini Gupta R4 LAN A
R4(config-if)#ip address 192.168.4.1 255.255.255.128
R4(config-if)#exit
R4(config)#
R4(config)#interface lo5
R4(config-if)#descript
*Mar  1 00:23:44.543: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback5
, changed state to up
R4(config-if)#description Ragini Gupta R4 LAN B
R4(config-if)#ip address 192.168.4.129 255.255.255.128
R4(config-if)#exit
```

```
R4(config)#
R4(config)#interface serial0/0
R4(config-if)#description Ragini Gupta R4-->R3
R4(config-if)#ip address 172.16.34.4 255.255.255.248
R4(config-if)#bandwidth 64
R4(config-if)#no shutdown
R4(config-if)#end
R4#
*Mar  1 00:25:48.015: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
R4#
*Mar  1 00:25:49.091: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:25:49.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0
, changed state to up
R4#
```

b. Verify the configuration with the show ip interface brief, show protocols, and show interfaces description commands.

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Pro
tocol
FastEthernet0/0            unassigned      YES unset  administratively down dow
n
Serial0/0                  172.16.12.1     YES manual up                    dow
n
FastEthernet0/1            unassigned      YES unset  administratively down dow
n
Serial0/1                  172.16.13.1     YES manual up                    up

Serial1/0                  unassigned      YES unset  administratively down dow
n
Serial1/1                  unassigned      YES unset  administratively down dow
n
Serial1/2                  unassigned      YES unset  administratively down dow
n
Serial1/3                  unassigned      YES unset  administratively down dow
n
Loopback1                  192.168.1.1     YES manual up                    up

R1#
```

```
R1#
R1#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 down
  Internet address is 172.16.12.1/29
FastEthernet0/1 is administratively down, line protocol is down
Serial0/1 is up, line protocol is up
  Internet address is 172.16.13.1/29
Serial1/0 is administratively down, line protocol is down
Serial1/1 is administratively down, line protocol is down
Serial1/2 is administratively down, line protocol is down
Serial1/3 is administratively down, line protocol is down
Loopback1 is up, line protocol is up
  Internet address is 192.168.1.1/24
R1#
```

```
R1#show interface description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Se0/0                          up             down     Ragini Gupta R1-->R2
Fa0/1                          admin down     down
Se0/1                          up             up       Ragini Gupta R1-->R3
Se1/0                          admin down     down
Se1/1                          admin down     down
Se1/2                          admin down     down
Se1/3                          admin down     down
Lo1                            up             up       R1 LAN Ragini Gupta
R1#
```

```
R2#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES unset  administratively down down
Serial0/0                  172.16.12.2     YES manual up                    up
FastEthernet0/1            unassigned      YES unset  administratively down down
Serial0/1                  172.16.23.2     YES manual up                    up
Serial1/0                  unassigned      YES unset  administratively down down
Serial1/1                  unassigned      YES unset  administratively down down
Serial1/2                  unassigned      YES unset  administratively down down
Serial1/3                  unassigned      YES unset  administratively down down
Loopback2                  192.168.2.1     YES manual up                    up
R2#
```

```
R2#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 172.16.12.2/29
FastEthernet0/1 is administratively down, line protocol is down
Serial0/1 is up, line protocol is up
  Internet address is 172.16.23.2/29
Serial1/0 is administratively down, line protocol is down
Serial1/1 is administratively down, line protocol is down
Serial1/2 is administratively down, line protocol is down
Serial1/3 is administratively down, line protocol is down
Loopback2 is up, line protocol is up
  Internet address is 192.168.2.1/24
R2#
```

```
R2#show interface description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Se0/0                          up             up       Ragini Gupta R2-->R1
Fa0/1                          admin down     down
Se0/1                          up             up       Ragini Gupta R2-->R3
Se1/0                          admin down     down
Se1/1                          admin down     down
Se1/2                          admin down     down
Se1/3                          admin down     down
Lo2                            up             up       Ragini Gupta R2 LAN
R2#
```

```
R3#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
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
Serial1/0                  172.16.34.3     YES manual up                    up
Serial1/1                  unassigned      YES unset  administratively down down
Serial1/2                  unassigned      YES unset  administratively down down
Serial1/3                  unassigned      YES unset  administratively down down
Loopback3                  192.168.3.1     YES manual up                    up
R3#
```

```
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
Serial1/0 is up, line protocol is up
  Internet address is 172.16.34.3/29
Serial1/1 is administratively down, line protocol is down
Serial1/2 is administratively down, line protocol is down
Serial1/3 is administratively down, line protocol is down
Loopback3 is up, line protocol is up
  Internet address is 192.168.3.1/24
R3#
```

```
R3#show interface description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Se0/0                          up             up       Ragini Gupta R3-->R1
Fa0/1                          admin down     down
Se0/1                          up             up       Ragini Gupta R3-->R2
Se1/0                          up             up       Ragini Gupta R3-->R4
Se1/1                          admin down     down
Se1/2                          admin down     down
Se1/3                          admin down     down
Lo3                            up             up       Ragini Gupta R3 LAN
R3#
```

```
R4#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES unset  administratively down down
Serial0/0                  172.16.34.4     YES manual up                    up
FastEthernet0/1            unassigned      YES unset  administratively down down
Serial0/1                  unassigned      YES unset  administratively down down
Serial1/0                  unassigned      YES unset  administratively down down
Serial1/1                  unassigned      YES unset  administratively down down
Serial1/2                  unassigned      YES unset  administratively down down
Serial1/3                  unassigned      YES unset  administratively down down
Loopback4                  192.168.4.1     YES manual up                    up
Loopback5                  192.168.4.129   YES manual up                    up
R4#
```

```
R4#show protocol
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 172.16.34.4/29
FastEthernet0/1 is administratively down, line protocol is down
Serial0/1 is administratively down, line protocol is down
Serial1/0 is administratively down, line protocol is down
Serial1/1 is administratively down, line protocol is down
Serial1/2 is administratively down, line protocol is down
Serial1/3 is administratively down, line protocol is down
Loopback4 is up, line protocol is up
  Internet address is 192.168.4.1/25
Loopback5 is up, line protocol is up
  Internet address is 192.168.4.129/25
R4#
R4#show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Se0/0                          up             up       Ragini Gupta R4-->R3
Fa0/1                          admin down     down
Se0/1                          admin down     down
Se1/0                          admin down     down
Se1/1                          admin down     down
Se1/2                          admin down     down
Se1/3                          admin down     down
Lo4                            up             up       Ragini Gupta R4 LAN A
Lo5                            up             up       Ragini Gupta R4 LAN B
R4#
```

Step 3- Configure basic EIGRP.

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

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router eigrp 1
R1(config-router)#network 192.168.1.0
R1(config-router)#network 172.16.12.0 0.0.0.7
R1(config-router)#network 172.16.13.0 0.0.0.7
R1(config-router)#no auto-summary
R1(config-router)#
```

```
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router eigrp 1
R2(config-router)#network 192.168.2.0
R2(config-router)#network 172.16.12.0 0.0.0.7
R2(config-router)#
*Mar  1 01:09:01.235: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.1 (S
R2(config-router)#network 172.16.23.0 0.0.0.7
R2(config-router)#no auto-summary
R2(config-router)#
*Mar  1 01:09:36.443: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.1 (S
R2(config-router)#no auto-summary
R2(config-router)#
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router eigrp 1
R3(config-router)#network 192.168.3.0
R3(config-router)#network 172.16.13.0 0.0.0.7
R3(config-router)#n
*Mar  1 01:11:06.775: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.1 (Serial0/0
R3(config-router)#network 172.16.23.0 0.0.0.7
R3(config-router)#
*Mar  1 01:11:26.311: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.23.2 (Serial0/1
R3(config-router)#network 172.16.34.0 0.0.0.7
R3(config-router)#no auto-summary
R3(config-router)#
*Mar  1 01:11:53.331: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.13.1 (Serial0/0
*Mar  1 01:11:53.335: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.23.2 (Serial0/1
R3(config-router)#
```

```
R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#router eigrp 1
R4(config-router)#network 192.168.4.0
R4(config-router)#network 172.16.34.0 0.0.0.7
R4(config-router)#no
*Mar  1 01:12:53.267: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Seria
R4(config-router)#no auto-summary
R4(config-router)#
*Mar  1 01:13:00.831: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.34.3 (Seria
R4(config-router)#
```

Step 4- Verify EIGRP connectivity.

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

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.13.3             Se0/1             12 00:02:56   75  2280  0  27
0   172.16.12.2             Se0/0             10 00:05:01   77  1140  0  24
R1#
```

```
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 00:03:39   65  1140  0  28
0   172.16.12.1             Se0/0             11 00:06:04   81  1140  0  26
R2#
```

```
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             Se1/0             11 00:02:43  166  2280  0  7
1   172.16.23.2             Se0/1             11 00:04:29   79  1140  0  25
0   172.16.13.1             Se0/0             12 00:04:49   72  2280  0  27
R3#
```

```
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             11 00:03:21  137  2280  0  32
R4#
```

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

```
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#192.168.4.129
+>(tcl)# } {
+>(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 = 40/128/168 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 = 4/79/212 ms
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 172/243/316 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 = 36/84/176 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 220/266/324 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/88/144 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 108/148/200 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/205/296 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.129, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 224/268/408 ms
R1(tcl)#
```

```
R2#tclsh
R2(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#192.168.4.129
+>(tcl)# } {
+>(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 = 68/93/136 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 = 144/223/288 ms
```

```
R3#tclsh
R3(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#192.168.4.129
+>(tcl)# } {
+>(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 = 116/175/244 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 = 80/111/140 ms
```

```
R4#tclsh
R4(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#192.168.4.129
+>(tcl)# } {
+>(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 = 176/305/404 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 = 156/240/332 ms
```

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

Step 5- Verify the current path.

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

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

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter 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

     172.16.0.0/29 is subnetted, 4 subnets
D       172.16.34.0 [90/41024000] via 172.16.13.3, 00:20:27, Serial0/1
D       172.16.23.0 [90/21024000] via 172.16.12.2, 00:20:48, Serial0/0
C       172.16.12.0 is directly connected, Serial0/0
C       172.16.13.0 is directly connected, Serial0/1
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:18:53, Serial0/1
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:18:53, Serial0/1
C    192.168.1.0/24 is directly connected, Loopback1
D    192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:20:51, Serial0/0
D    192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:20:51, Serial0/0
R1#
```

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

```
R4#traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 80 msec 136 msec 240 msec
  2 172.16.23.2 180 msec 112 msec 264 msec
  3 172.16.12.1 448 msec 260 msec 380 msec
R4#
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 232 msec 112 msec 176 msec
  2 172.16.23.2 216 msec 296 msec 416 msec
  3 172.16.12.1 284 msec 280 msec 280 msec
R4#
```

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

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

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter 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

     172.16.0.0/29 is subnetted, 4 subnets
C       172.16.34.0 is directly connected, Serial1/0
C       172.16.23.0 is directly connected, Serial0/1
D       172.16.12.0 [90/21024000] via 172.16.23.2, 00:23:44, Serial0/1
C       172.16.13.0 is directly connected, Serial0/0
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/40640000] via 172.16.34.4, 00:21:50, Serial1/0
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:21:50, Serial1/0
D    192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:23:53, Serial0/1
D    192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:23:53, Serial0/1
C    192.168.3.0/24 is directly connected, Loopback3
R3#
```

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

```
R3#show interface s0/0
Serial0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Description: Ragini Gupta R3-->R1
  Internet address is 172.16.13.3/29
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:03, output 00:00:02, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 48 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1074 packets input, 72652 bytes, 0 no buffer
     Received 604 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     961 packets output, 66616 bytes, 0 underruns
     0 output errors, 0 collisions, 7 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
 --More--
```

R1 R2 R3 R4

```
R3#
R3#show interface s0/1
Serial0/1 is up, line protocol is up
  Hardware is GT96K Serial
  Description: Ragini Gupta R3-->R2
  Internet address is 172.16.23.3/29
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:03, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 96 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1061 packets input, 74126 bytes, 0 no buffer
     Received 574 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1031 packets output, 73043 bytes, 0 underruns
     0 output errors, 0 collisions, 7 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

R3#
```

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

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

```
R3#show ip eigrp topology 192.168.1.0
IP-EIGRP (AS 1): Topology entry for 192.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000
  Routing Descriptor Blocks:
  172.16.23.2 (Serial0/1), from 172.16.23.2, Send flag is 0x0
      Composite metric is (21152000/20640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
  172.16.13.1 (Serial0/0), from 172.16.13.1, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
R3#
```

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

Step 6- Configure PBR to provide path control.

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

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

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

The steps required to implement path control include the following:

• Choose the path control tool to use. Path control tools manipulate or bypass the IP routing table. For PBR, route-map commands are used.

• Implement the traffic-matching configuration, specifying which traffic will be manipulated. The match commands are used within route maps.

• Define the action for the matched traffic using set commands within route maps.

- Apply the route map to incoming traffic.
- As a test, you will configure the following policy on router R3:

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

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

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

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

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

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

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

```
R3(config)#interface s1/0
R3(config-if)#ip policy route-map R3-to-R1
R3(config-if)#end
R3#
*Mar  1 01:43:16.819: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

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

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

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

Step 7- Test the policy.

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

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

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#access-list 1 permit 192.168.4.0 0.0.0.255
R3(config)#exit
R3#
*Mar  1 01:45:14.283: %SYS-5-CONFIG_I: Configured from console by console
R3#
```

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

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

R3#debug ip policy ?
  dynamic  dynamic PBR

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

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

```
R4#traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 108 msec 224 msec 112 msec
  2 172.16.23.2 320 msec 252 msec 344 msec
  3 172.16.12.1 248 msec 360 msec 208 msec
R4#
```

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

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

```
R3#
*Mar  1 01:47:57.483: IP: s=192.168.4.1 (Serial1/0), d=192.168.1.1, len 28, FIB policy rejected(r
ing
*Mar  1 01:47:57.859: IP: s=192.168.4.1 (Serial1/0), d=192.168.1.1, len 28, FIB policy rejected(r
ing
*Mar  1 01:47:58.091: IP: s=192.168.4.1 (Serial1/0), d=192.168.1.1, len 28, FIB policy rejected(r
ing
*Mar  1 01:47:58.375: IP: s=192.168.4.1 (Serial1/0), d=192.168.1.1, len 28, FIB policy rejected(r
ing
R3#
*Mar  1 01:47:58.707: IP: s=192.168.4.1 (Serial1/0), d=192.168.1.1, len 28, FIB policy rejected(r
ing
*Mar  1 01:47:59.027: IP: s=192.168.4.1 (Serial1/0), d=192.168.1.1, len 28, FIB policy rejected(r
ing
R3#
```

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

```
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 240 msec 252 msec 124 msec
  2 172.16.13.1 392 msec 220 msec 188 msec
R4#
```

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

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

```
R3#
*Mar  1 01:49:59.107: IP: s=192.168.4.129 (Serial1/0), d=192.168.1.1, len 28, FIB policy ma
*Mar  1 01:49:59.107: IP: s=192.168.4.129 (Serial1/0), d=192.168.1.1, g=172.16.13.1, len 28
*Mar  1 01:49:59.515: IP: s=192.168.4.129 (Serial1/0), d=192.168.1.1, len 28, FIB policy ma
*Mar  1 01:49:59.515: IP: s=192.168.4.129 (Serial1/0), d=192.168.1.1, g=172.16.13.1, len 28
*Mar  1 01:49:59.699: IP: s=192.168.4.129 (Serial1/0), d=192.168.1.1, len 28, FIB policy ma
*Mar  1 01:49:59.699: IP: s=192.168.4.129 (Serial1/0), d=192.168.1.1, g=172.16.13.1, len 28
R3#
```

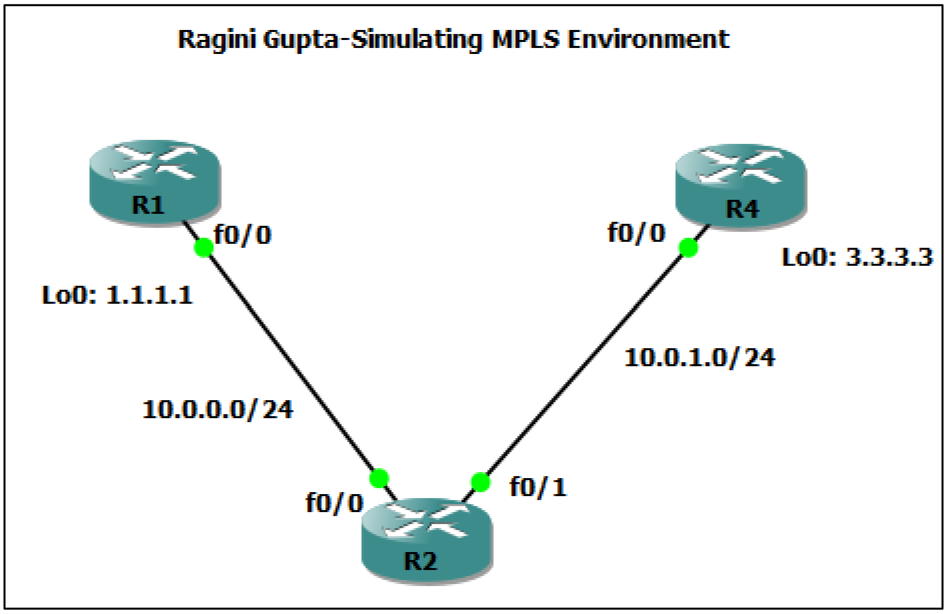
e. On R3, display the policy and matches using the show route-map command.

```
R3#show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 3 packets, 96 bytes
R3#
```

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

Practical-6

Aim- Simulating MPLS Environment.



Steps 1-IP addressing of MPLS Core and OSPF.

First bring 3 routers into your topology R1, R2, R4 position them as below. We are going to address the routers and configure ospf to ensure loopback to loopback connectivity between R1 and R4.

R1 Router-

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#interface loopback0
R1(config-if)#descr
*Mar  1 00:02:22.115: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R1(config-if)#description Ragini Gupta
R1(config-if)#ip address 1.1.1.1 255.255.255.255
R1(config-if)#ip ospf 1 area 0
R1(config-if)#interface f0/0
R1(config-if)#ip address 10.0.0.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#ip os
*Mar  1 00:03:31.687: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Mar  1 00:03:32.687: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/0, changed state to up
R1(config-if)#ip ospf 1 area 0
R1(config-if)#
```

R2 Router-

```
R2(config)#hostname R2
R2(config)#interface Loopback 0
R2(config-if)#ip address 2.2.2.2 255.255.255.255
R2(config-if)#ip ospf 1 area 0
R2(config-if)#exit
R2(config)#interface f0/0
R2(config-if)#ip address 10.0.0.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#ip ospf 1 area 0
R2(config-if)#exit
R2(config)#interface f0/1
R2(config-if)#ip address 10.0.1.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#
*Mar  1 00:21:05.119: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state
to up
*Mar  1 00:21:06.119: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/1, changed state to up
R2(config-if)#ip ospf 1 area 0
R2(config-if)#
*Mar  1 00:21:21.463: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on FastEthernet0/1
 from LOADING to FULL, Loading Done
R2(config-if)#
```

R4 Router-

```
R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#hostname R4
R4(config)#interface loopback 0
R4(config-if)#ip address 3.3.3.3 255.255.255.255
R4(config-if)#ip ospf 1 area 0
R4(config-if)#exit
R4(config)#interface f0/0
R4(config-if)#ip address 10.0.1.3 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#ip ospf 1 area 0
R4(config-if)#
```

You should now have full ip connectivity between R1, R2, R4 to verify this we need to see if we can ping between the loopbacks of R1 and R4.

```
R1#ping 3.3.3.3 source loopback 0

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 = 284/358/464 ms
R1#
```

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

Step 2 – Configure LDP on all the interfaces in the MPLS Core In order to run MPLS you need to enable it, there are two ways to do this.

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

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#mpls ldp autoconfig
R1(config-router)#
```

```
R2#
*Mar  1 00:29:04.323: %SYS-5-CONFIG_I: Configured from console by console
R2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#mpls ldp autoconfig
R2(config-router)#
*Mar  1 00:29:31.919: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
R2(config-router)#
```

```
R4#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#router ospf 1
R4(config-router)#mpls ldp autoconfig
R4(config-router)#
*Mar  1 00:30:07.439: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is UP
R4(config-router)#
```

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.

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
R2#
```

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

```
R2#sh mpls ldp neigh
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 1.1.1.1.646 - 2.2.2.2.60096
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:02:36
        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: 3.3.3.3.38788 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 9/9; Downstream
        Up time: 00:01:41
        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
R2#
```

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

```
R1#trace 3.3.3.3

Type escape sequence to abort.
Tracing the route to 3.3.3.3

  1 10.0.0.2 [MPLS: Label 17 Exp 0] 220 msec 268 msec 480 msec
  2 10.0.1.3 524 msec 424 msec 488 msec
R1#
```

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 R4. We need to establish a Multi-Protocol BGP session between R1 and R4 this is done by configuring the vpnv4 address family as below.

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router bgp 1
R1(config-router)#neighbor 3.3.3.3 remote-as 1
R1(config-router)#neighbor 3.3.3.3 update-source loopback 0
R1(config-router)#no auto-summary
R1(config-router)#
R1(config-router)#address-family vpnv4
R1(config-router-af)#neighbor 3.3.3.3 activate
R1(config-router-af)#
```

```
R4(config)#router bgp 1
R4(config-router)#neighbor 1.1.1.1 remote-as 1
R4(config-router)#neighbor 1.1.1.1 update-source
*Mar  1 00:46:19.751: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
R4(config-router)#neighbor 1.1.1.1 update-source loopback 0
R4(config-router)#no auto-summary
R4(config-router)#address-family vpnv4
R4(config-router-af)#neighbor 1.1.1.1 activate
R4(config-router-af)#
*Mar  1 00:47:04.331: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down Address family ac
tivated
R4(config-router-af)#
*Mar  1 00:47:07.247: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
R4(config-router-af)#
```

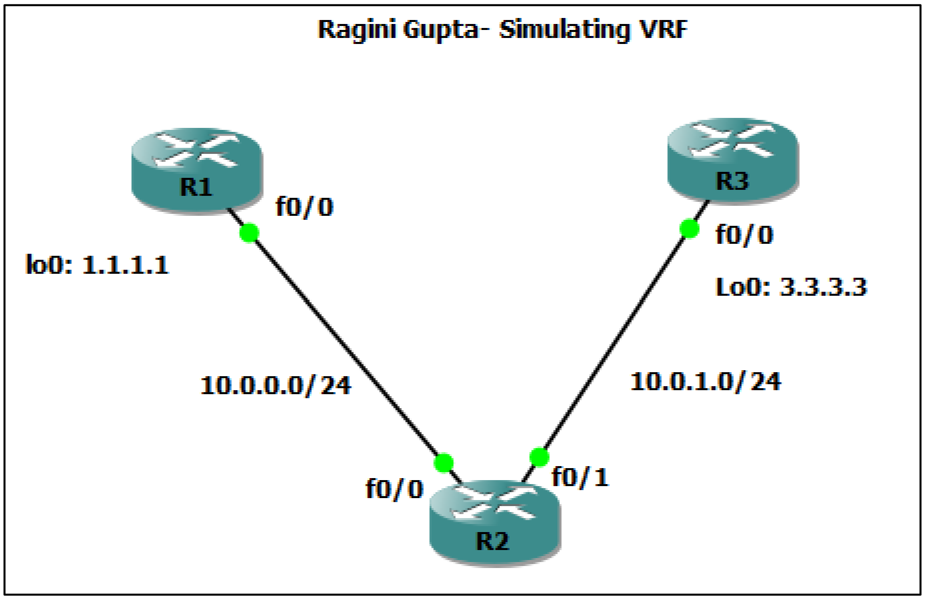
You should see log messages showing the BGP sessions coming up. To verify the BGP session between R1 and R4 issue the command sh bgp vpnv4 unicast all summary.

```
R1#sh bgp vpnv4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRc
d
3.3.3.3         4     1       9       9        1    0    0 00:01:15        0
R1#
```

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

Practical-7

Aim-Simulating VRF.



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 Router-

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#hostname R1
R1(config)#interface loopback 0
R1(config-if)#ip
*Mar  1 00:01:51.499: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R1(config-if)#ip address 1.1.1.1 255.255.255.255
R1(config-if)#ip ospf 1 area 0
R1(config-if)#exit
R1(config)#interface f0/0
R1(config-if)#ip address 10.0.0.1 255.255.255.0
R1(config-if)#no shutdown
R1(config-if)#ip o
*Mar  1 00:02:50.491: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Mar  1 00:02:51.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/0, changed state to up
R1(config-if)#ip ospf 1 area 0
R1(config-if)#
```

R2 Router-

```
R2(config)#hostname R2
R2(config)#interface loopback0
R2(config-if)#ip a
*Mar  1 00:04:15.667: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R2(config-if)#ip address 2.2.2.2 255.255.255.255
R2(config-if)#ip ospf 1 area 0
R2(config-if)#exit
R2(config)#interface f0/0
R2(config-if)#ip address 10.0.0.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#ip os
*Mar  1 00:05:17.527: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Mar  1 00:05:18.527: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/0, changed state to up
R2(config-if)#ip ospf 1 area 0
R2(config-if)#exit
R2(config)#inte
*Mar  1 00:05:27.483: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0
 from LOADING to FULL, Loading Done
R2(config)#interface f0/1
R2(config-if)#ip address 10.0.1.2 255.255.255.0
R2(config-if)#no shutdown
R2(config-if)#
*Mar  1 00:05:56.595: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state
to up
*Mar  1 00:05:57.595: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
```

```
R2(config-if)#no shutdown
R2(config-if)#
*Mar  1 00:05:56.595: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state
to up
*Mar  1 00:05:57.595: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/1, changed state to up
R2(config-if)#ip ospf 1 area 0
R2(config-if)#
```

R3 Router-

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#hostname R3
R3(config)#interface loopback0
R3(config-if)#ip a
*Mar  1 00:08:46.687: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R3(config-if)#ip address 3.3.3.3 255.255.255.255
R3(config-if)#ip ospf 1 area 0
R3(config-if)#exit
R3(config)#interface f0/0
R3(config-if)#ip address 10.0.1.3 255.255.255.0
R3(config-if)#no shutdown
R3(config-if)#ip ospf
*Mar  1 00:09:42.455: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Mar  1 00:09:43.455: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/0, changed state to up
R3(config-if)#ip ospf 1 area 0
R3(config-if)#
```

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

```
R1#ping 3.3.3.3 source loopback0

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 = 148/237/372 ms
R1#
```

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

Step 2 – Configure LDP on all the interfaces in the MPLS Core. In order to run MPLS you need to enable it, there are two ways to do this.

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

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 1
R1(config-router)#mpls ldp autoconfig
R1(config-router)#
```

```
R2(config-if)#exit
R2(config)#router ospf 1
R2(config-router)#mpls ldp autoconfig
R2(config-router)#
*Mar  1 00:13:37.491: %LDP-5-NBRCHG: LDP Neighbor 1.1.1.1:0 (1) is UP
R2(config-router)#
```

```
R3(config)#router ospf 1
R3(config-router)#mpls ldp autoconfig
R3(config-router)#
*Mar  1 00:14:32.739: %LDP-5-NBRCHG: LDP Neighbor 2.2.2.2:0 (1) is UP
R3(config-router)#
```

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

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
R2#
```

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

```
R2#sh mpls ldp neigh
    Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 1.1.1.1.646 - 2.2.2.2.32474
        State: Oper; Msgs sent/rcvd: 10/10; Downstream
        Up time: 00:02:14
        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: 3.3.3.3.39355 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 9/9; Downstream
        Up time: 00:01:19
        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
R2#
```

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

```
R1#trace 3.3.3.3

Type escape sequence to abort.
Tracing the route to 3.3.3.3

  1 10.0.0.2 [MPLS: Label 17 Exp 0] 96 msec 220 msec 184 msec
  2 10.0.1.3 156 msec 268 msec 232 msec
R1#
```

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 vpnv4 address family as below.

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router bgp 1
R1(config-router)#neighbor 3.3.3.3 remote-as 1
R1(config-router)#neighbor 3.3.3.3 update-source loopback0
R1(config-router)#no auto-summary
R1(config-router)#
R1(config-router)#address-family vpnv4
R1(config-router-af)#neighbor 3.3.3.3 activate
R1(config-router-af)#
```

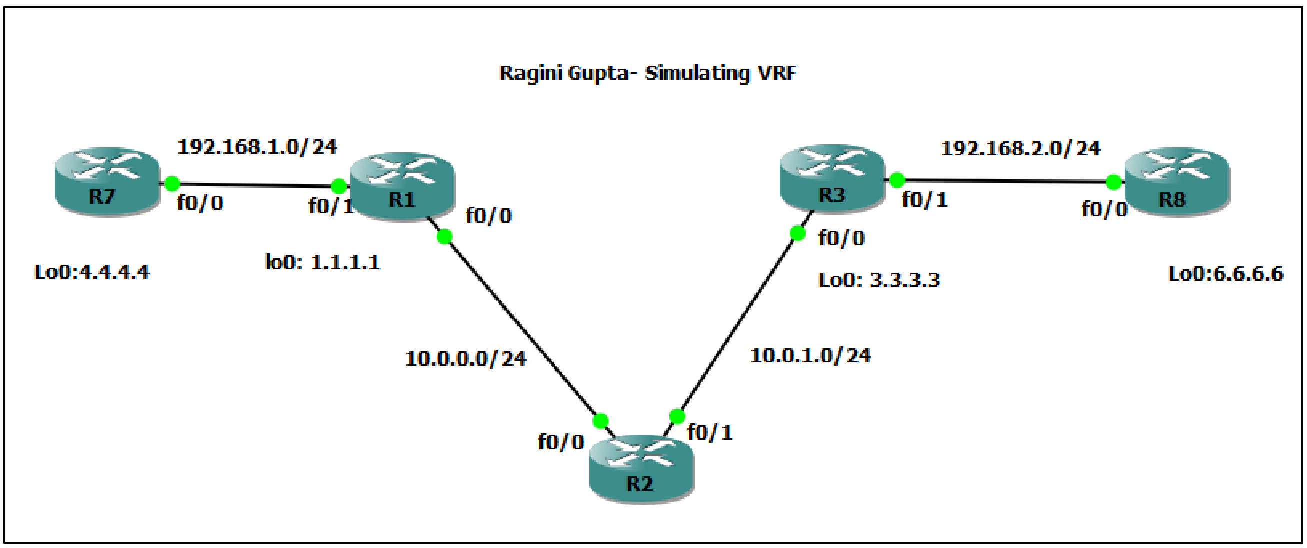
```
R3(config)#router bgp 1
R3(config-router)#neighbor 1.1.1.1 remote-as 1
R3(config-router)#neighbor 1.1.1.1 update-source loopback0
R3(config-router)#
*Mar  1 00:20:34.443: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
R3(config-router)#no auto-summary
R3(config-router)#
R3(config-router)#address-family vpnv4
R3(config-router-af)#neighbor 1.1.1.1 activate
R3(config-router-af)#
*Mar  1 00:21:12.307: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Down Address family a
tivated
R3(config-router-af)#
*Mar  1 00:21:14.607: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
R3(config-router-af)#
```

You should see log messages showing the BGP sessions coming up. To verify the BGP session between R1 and R3 issue the command sh bgp vpnv4 unicast all summary.

```
R1#sh bgp vpnv4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRc
d
3.3.3.3         4     1       8       8        1    0    0 00:01:26        0
R1#
```

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

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.



Note-In Place of R4 router I am assume R7 And Instead of R5 is R8.

```
R7#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R7(config)#interface loopback0
R7(config-if)#i
*Mar  1 00:01:48.915: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R7(config-if)#ip address 4.4.4.4 255.255.255.255
R7(config-if)#ip ospf 2 area 2
R7(config-if)#exit
R7(config)#interface f0/0
R7(config-if)#ip address 192.168.1.4 255.255.255.0
R7(config-if)#ip ospf 2 area 2
R7(config-if)#no shutdown
R7(config-if)#
*Mar  1 00:02:56.943: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Mar  1 00:02:57.943: %LINEPROTO-5-UPDOWN: Line protocol
```

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface f0/1
R1(config-if)#no shutdown
R1(config-if)#ip addr
*Mar  1 00:05:13.415: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state
to up
*Mar  1 00:05:14.415: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/1, changed state to up
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#
```

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 192.168.1.0/24 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.

```
R1(config)#interface f0/1
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
R1(config-vrf)#
```

The RD and route-target do not need to be the same.

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

```
R1(config)#interface f0/1
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
R1(config-vrf)#exit
R1(config)#interface f0/1
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
R1(config-vrf)#exit
R1(config)#interface f0/1
R1(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.1.1 removed due to enabling VRF
RED
R1(config-if)#
```

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

You just need to re-apply it.

```
R1(config)#interface f0/1
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#ip vrf RED
R1(config-vrf)#rd 4:4
R1(config-vrf)#route-target both 4:4
R1(config-vrf)#exit
R1(config)#interface f0/1
R1(config-if)#ip vrf forwarding RED
R1(config-if)#exit
R1(config)#interface f0/1
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#
```

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

```
R1#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
 ip vrf forwarding RED
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
end

R1#
```

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

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

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

```
R1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA 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, 00:14:34, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/21] via 10.0.0.2, 00:14:34, 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, 00:14:34, 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#sh ip route vrf red
% IP routing table red does not exist
R1#
```

Note The VRF name is case sensitive!

```
R1#sh ip route vrf red
% IP routing table red does not exist
R1#
R1#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA 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    192.168.1.0/24 is directly connected, FastEthernet0/1
R1#
```

We just need to enable OSPF on this interface and get the loopback address for R4 in the VRF RED routing table before proceeding. You should see a log message showing the OSPF neighbor come up.

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface f0/1
R1(config-if)#ip ospf 2 area 2
R1(config-if)#
*Mar  1 00:20:25.951: %OSPF-5-ADJCHG: Process 2, Nbr 4.4.4.4 on FastEthernet0/1
 from LOADING to FULL, Loading Done
R1(config-if)#
```

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

```
R1#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA 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
O       4.4.4.4 [110/11] via 192.168.1.4, 00:00:35, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/1
R1#
```

We now need to repeat this process for R3 & R5. Router 5 will peer OSPF using process number 2 to a VRF configured on R3. It will use the local site addressing of 192.168.2.0/24.

```
R8#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R8(config)#interface loopback0
R8(config-if)#ip
*Mar  1 00:23:43.563: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0
, changed state to up
R8(config-if)#ip address 6.6.6.6 255.255.255.255
R8(config-if)#ip ospf 2 area 2
R8(config-if)#exit
R8(config)#interface f0/0
R8(config-if)#ip address 192.168.2.6 255.255.255.0
R8(config-if)#ip ospf 2 area 2
R8(config-if)#no shutdown
R8(config-if)#
*Mar  1 00:25:04.575: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state
to up
*Mar  1 00:25:05.575: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEther
net0/0, changed state to up
R8(config-if)#
```

```
R3(config)#interface f0/1
R3(config-if)#no shutdown
R3(config-if)#ip address 192.168.2.3 255.255.255.0
```

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

```
R3(config)#interface f0/1
R3(config-if)#no shutdown
R3(config-if)#ip ad
*Mar  1 00:26:32.867: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed sta
to up
*Mar  1 00:26:33.867: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEt
net0/1, changed state to up
R3(config-if)#ip address 192.168.2.3 255.255.255.0
R3(config-if)#exit
R3(config)#interface f0/1
R3(config-if)#no shutdown
R3(config-if)#ip address 192.168.2.3 255.255.255.0
R3(config-if)#ip vrf RED
R3(config-vrf)#rd 4:4
R3(config-vrf)#route-target both 4:4
```

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

```
R3(config)#interface f0/1
R3(config-if)#ip address 192.168.2.3 255.255.255.0
R3(config-if)#ip vrf RED
R3(config-vrf)#rd 4:4
R3(config-vrf)#route-target both 4:4
R3(config-vrf)#
R3(config-vrf)#interface f0/1
R3(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.2.3 removed due to enabling V
RED
R3(config-if)#exit
R3(config)#interface f0/1
R3(config-if)#ip vrf forwarding RED
R3(config-if)#ip address 192.168.2.1 255.255.255.0
R3(config-if)#exit
```

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

```
R3(config)#interface f0/1
R3(config-if)#ip address 192.168.2.1 255.255.255.0
R3(config-if)#exit
R3(config)#exit
R3#
*Mar  1 00:39:58.071: %SYS-5-CONFIG_I: Configured from console by console
R3#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
 ip vrf forwarding RED
 ip address 192.168.2.1 255.255.255.0
 duplex auto
 speed auto
end

R3#
```

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

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#interface f0/1
R3(config-if)#ip ospf 2 area 2
R3(config-if)#
*Mar  1 00:51:04.283: %OSPF-5-ADJCHG: Process 2, Nbr 6.6.6.6 on FastEthernet0/1
 from LOADING to FULL, Loading Done
R3(config-if)#
```

Check the routes in vrf RED.

```
R3#sh ip route vrf lred
% IP routing table lred does not exist
R3#
R3#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

     6.0.0.0/32 is subnetted, 1 subnets
O       6.6.6.6 [110/11] via 192.168.2.6, 00:01:27, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/1
R3#
```

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 let's do some verifications. Check the routes on R4.

```
R7#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

     4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
R7#
```

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#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 rou
       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, 00:54:24, FastEthernet0/0
     3.0.0.0/32 is subnetted, 1 subnets
O       3.3.3.3 [110/21] via 10.0.0.2, 00:54:24, 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, 00:54:24, FastEthernet0/0
R1#
```

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

```
R1#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA 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
O       4.4.4.4 [110/11] via 192.168.1.4, 00:38:16, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/1
R1#
```

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

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

Redistribute OSPF into MP-BGP on R1.

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router bgp 1
R1(config-router)#address-family ipv4 vrf RED
R1(config-router-af)#redistribute ospf 2
R1(config-router-af)#
```

Redistribute OSPF into MP-BGP on R3.

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router bgp 1
R3(config-router)#address-family ipv4 vrf RED
R3(config-router-af)#redistribute ospf 2
R3(config-router-af)#
```

This has enabled redistribution of the OSPF routes into BGP. We can check the routes from R4 and R6 are now showing in the BGP table for their VRF with this command **sh ip bgp vpnv4 vrf RED**.

```
R1#sh ip bgp vpnv4 vrf RED
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, 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       192.168.1.4             11         32768 ?
*>i6.6.6.6/32       3.3.3.3                 11    100      0 ?
*> 192.168.1.0      0.0.0.0                  0         32768 ?
*>i192.168.2.0      3.3.3.3                  0    100      0 ?
R1#
```

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

```
R3#sh ip bgp vpnv4 vrf RED
BGP table version is 9, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
*>i4.4.4.4/32       1.1.1.1                 11    100      0 ?
*> 6.6.6.6/32       192.168.2.6             11         32768 ?
*>i192.168.1.0      1.1.1.1                  0    100      0 ?
*> 192.168.2.0      0.0.0.0                  0         32768 ?
R3#
```

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

```
R1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#router ospf 2
R1(config-router)#redistribute bgp 1 subnets
R1(config-router)#
```

```
R3#config t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#router ospf 2
R3(config-router)#redistribute bgp 1 subnets
R3(config-router)#
```

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

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

```
R7#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

     4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Loopback0
     6.0.0.0/32 is subnetted, 1 subnets
O IA    6.6.6.6 [110/21] via 192.168.1.1, 00:02:48, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O IA 192.168.2.0/24 [110/11] via 192.168.1.1, 00:02:48, FastEthernet0/0
R7#
```

Great we have 6.6.6.6 in there **Also check the routing table on R5.**

```
R8#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

     4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/21] via 192.168.2.1, 00:01:42, FastEthernet0/0
     6.0.0.0/32 is subnetted, 1 subnets
C       6.6.6.6 is directly connected, Loopback0
O IA 192.168.1.0/24 [110/11] via 192.168.2.1, 00:01:42, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/0
R8#
```

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

```
R7#
R7#ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 876/1000/1124 ms
R7#
```

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

```
R7#trace 6.6.6.6

Type escape sequence to abort.
Tracing the route to 6.6.6.6

  1 192.168.1.1 172 msec 228 msec 300 msec
  2 10.0.0.2 [MPLS: Labels 17/19 Exp 0] 1076 msec 1080 msec 1316 msec
  3 192.168.2.1 [MPLS: Label 19 Exp 0] 664 msec 888 msec 716 msec
  4 192.168.2.6 808 msec 988 msec 876 msec
R7#
```