University Of Mumbai
Institute of Distance & Open Learning

PRACTICAL JOURNAL

MODERN NETWORKING

SUBMITTED BY

DRIZZLE GONSALVES

SEAT NO:8166014

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY PART-I SEMESTER II

ACADEMIC YEAR
2024-2025

INSTITUTE OF DISTANCE AND OPEN LEARNING IDOL BUILDING, VIDYANAGARI, SANTACRUZ (EAST), MUMBAI-400 098

CONDUCTED AT
RIZVI COLLEGE OF ARTS, SCIENCE AND COMMERCE BANDRA (W), MUMBAI 400050

University Of Mumbai
Institute of Distance & Open Learning

Dr.Shankar Dayal Sharama Bhavan, Kalina, Vidanagari,
Santacruz (E), Mumbai-400 098.

PCP CENTRE :
RIZVI COLLEGE OF ARTS, SCIENCE AND COMMERCE, BANDRA(W), MUMBAI

Certificate

This is to certify that

Ms. *DRIZZLE GONSALVES* of M.Sc.-I.T. SEM-II, Seat No: _8166014_ has performed all the Practical of _MODERN NETWORKING_ for MSC.IT Part I. This Journal represents her Bonafide work during the academic year: 2024-2025.

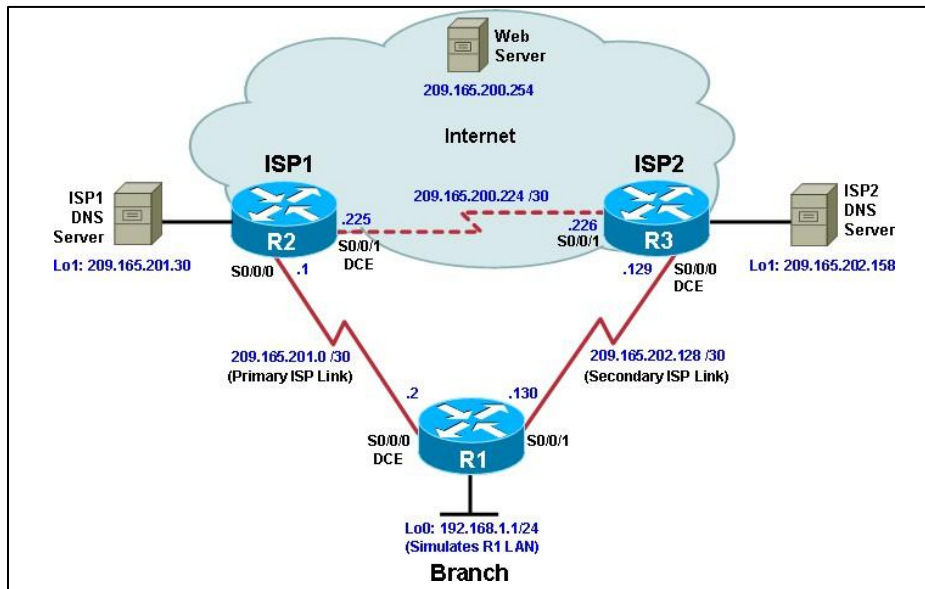_____   _____

MSc (IT) Co-ordinator, IDOL   External Examiner

INDEX

PRACTICAL 1

AIM: Configure IP SLA Tracking and Path Control Topology

Objectives

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

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

Step 1: Configure loopbacks and assign addresses.

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

Router R1

hostname R1

interface Loopback 0

description R1 LAN

ip address 192.168.1.1 255.255.255.0

interface Serial0/0/0

description R1 --> ISP1

ip address 209.165.201.2 255.255.255.252

```
clock rate 128000

bandwidth 128

no shutdown



interface Serial0/0/1

description R1 --> ISP2

ip address 209.165.202.130 255.255.255.252

bandwidth 128

no shutdown
```

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0

description Simulated Internet Web Server

ip address 209.165.200.254 255.255.255.255



interface Loopback1

description ISP1 DNS Server

ip address 209.165.201.30 255.255.255.255



interface Serial0/0/0

description ISP1 --> R1

ip address 209.165.201.1 255.255.255.252

bandwidth 128

no shutdown



interface Serial0/0/1

description ISP1 --> ISP2

ip address 209.165.200.225 255.255.255.252

clock rate 128000

bandwidth 128

no shutdown
```

Router ISP2 (R3)

hostname ISP2

interface Loopback0

 description Simulated Internet Web Server

 ip address 209.165.200.254 255.255.255.255

 interface Loopback1

 description ISP2 DNS Server

 ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0

 description ISP2 --> R1

 ip address 209.165.202.129 255.255.255.252

 clock rate 128000

 bandwidth 128

 no shutdown

 interface Serial0/0/1

 description ISP2 --> ISP1

 ip address 209.165.200.226 255.255.255.252

 bandwidth 128

 no shutdown

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

```
*Mar  1 00:28:51.855: %SYS-5-CONFIG_I: Configured from console by console
R2#show interfaces description
Interface              Status        Protocol Description
Fa0/0                  admin down    down
Se0/0                  up            up       R2 --> ISP1
Fa0/1                  admin down    down
Se1/0                  up            up       R2 --> ISP2
Se1/1                  admin down    down
Se1/2                  admin down    down
Se1/3                  admin down    down
Lo0                    up            up       R2 LAN
R2#conf t
```

Step 2: Configure static routing.

a. Implement the routing policies on the respective routers.

Router R1

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

R1(config)#

Router ISP1 (R2)

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

ISP1(config-router)# **network 209.165.200.224 0.0.0.3**

ISP1(config-router)# **network 209.165.201.0 0.0.0.31**

ISP1(config-router)# **no auto-summary**

ISP1(config-router)# **exit**

ISP1(config)#

ISP1(config-router)# **ip route 192.168.1.0 255.255.255.0 209.165.201.2**

ISP1(config)#

Router ISP2 (R3)

ISP2(config)# **router eigrp 1**

ISP2(config-router)# **network 209.165.200.224 0.0.0.3**

ISP2(config-router)# **network 209.165.202.128 0.0.0.31**

ISP2(config-router)# **no auto-summary**

ISP2(config-router)# **exit**

ISP2(config)#

ISP2(config)# **ip route 192.168.1.0 255.255.255.0 209.165.202.130**

ISP2(config)#

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

```
R2#tclsh
R2(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 = 36/67/120 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 = 24/65/112 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 = 84/131/200 ms
```

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

```
R2(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 56 msec 24 msec 48 msec
Type escape sequence to abort.
Tracing the route to 209.165.201.30

  1 209.165.201.1 52 msec 76 msec 136 msec
Type escape sequence to abort.
Tracing the route to 209.165.202.158

  1 209.165.201.1 16 msec 12 msec 60 msec
  2 209.165.200.226 60 msec 112 msec 112 msec
```

Step 3: Configure IP SLA probes.

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

In this scenario, you will configure ICMP echo probes.

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

R1(config)# **ip sla 11**

R1(config-ip-sla)# **icmp-echo 209.165.201.30**

R1(config-ip-sla-echo)# **frequency 10**

R1(config-ip-sla-echo)# **exit**

R1(config)#

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

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

```
R2(config)#do 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:
```

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

```
R2(config)#do show ip sla monitor statistic
Round trip time (RTT)   Index 11
        Latest RTT: 36 ms
Latest operation start time: *00:51:20.719 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 20
Number of failures: 0
Operation time to live: Forever
```

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

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

R1(config)# **ip sla 22**

R1(config-ip-sla)# **icmp-echo 209.165.202.158**

R1(config-ip-sla-echo)# **frequency 10**

R1(config-ip-sla-echo)# **exit**

R1(config)#

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

R1(config)# **end**

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

```
R2(config)#do 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:


R2(config)#do show ip sla monitor statistics
Round trip time (RTT)   Index 11
        Latest RTT: 48 ms
Latest operation start time: *00:55:00.719 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 42
Number of failures: 0
Operation time to live: Forever

Round trip time (RTT)   Index 22
        Latest RTT: 220 ms
Latest operation start time: *00:54:57.327 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 9
Number of failures: 0
Operation time to live: Forever
```

Step 4: Configure tracking options.

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

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

R1(config)# **no ip route 0.0.0.0 0.0.0.0 209.165.201.1**

R1(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.1 5**

R1(config)# **exit**

b. Verify the routing table.

```
*Mar  1 00:57:49.723: %SYS-5-CONFIG_I: Configured from console by console
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial1/0
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [5/0] via 209.165.201.1
```

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

R1(config)# **track 1 ip sla 11 reachability**

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

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

R1(config-track)# **exit**

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

R1# **debug ip routing**

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

```
*Mar  1 01:02:18.135: %SYS-5-CONFIG_I: Configured from console by console
R2#debug ip routing
IP routing debugging is on
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#
*Mar  1 01:03:12.727: RT: NET-RED 0.0.0.0/0
R2(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
R2(config)#
*Mar  1 01:03:35.363: RT: closer admin distance for 0.0.0.0, flushing 1 routes
*Mar  1 01:03:35.363: RT: NET-RED 0.0.0.0/0
*Mar  1 01:03:35.363: RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 209.165.201.1

*Mar  1 01:03:35.363: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]
*Mar  1 01:03:35.363: RT: NET-RED 0.0.0.0/0
*Mar  1 01:03:35.363: RT: default path is now 0.0.0.0 via 209.165.201.1
*Mar  1 01:03:35.363: RT: new default network 0.0.0.0
*Mar  1 01:03:35.363: RT: NET-RED 0.0.0.0/0
R2(config)#
*Mar  1 01:03:40.363: RT: NET-RED 0.0.0.0/0
R2(config)#
*Mar  1 01:04:12.727: RT: NET-RED 0.0.0.0/0
R2(config)#
*Mar  1 01:05:12.727: RT: NET-RED 0.0.0.0/0
```

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

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial1/0
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

h.   Verify the routing table again.

R1#show ip route | begin Gateway

<mark>Gateway of last resort is 209.165.201.1 to network 0.0.0.0</mark>

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

Step 5: Verify IP SLA operation.

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

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

a.   On ISP1, disable the loopback interface 1.

ISP1(config-if)# **int lo1**

ISP1(config-if)# **shutdown**

Jan 10 10:53:25.091: %LINK-5-CHANGED: Interface Loopback1, changed state to administratively down

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

```
R2#
*Mar  1 01:09:04.539: RT: del 0.0.0.0 via 209.165.201.1, static metric [2/0]
*Mar  1 01:09:04.539: RT: delete network route to 0.0.0.0
*Mar  1 01:09:04.539: RT: NET-RED 0.0.0.0/0
*Mar  1 01:09:04.543: RT: NET-RED 0.0.0.0/0
*Mar  1 01:09:04.543: RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 209.165.202.129

*Mar  1 01:09:04.547: RT: add 0.0.0.0/0 via 209.165.202.129, static metric [3/0]
*Mar  1 01:09:04.547: RT: NET-RED 0.0.0.0/0
*Mar  1 01:09:04.547: RT: default path is now 0.0.0.0 via 209.165.202.129
*Mar  1 01:09:04.547: RT: new default network 0.0.0.0
*Mar  1 01:09:04.551: RT: NET-RED 0.0.0.0/0
R2#
*Mar  1 01:09:12.727: RT: NET-RED 0.0.0.0/0
```

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

c.   On R1, verify the routing table.

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.202.129 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial1/0
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [3/0] via 209.165.202.129
```

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

d. Verify the IP SLA statistics.

```
R2#show ip sla monitor statistics
Round trip time (RTT)    Index 11
        Latest RTT: 88 ms
Latest operation start time: *01:15:40.719 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 140
Number of failures: 26
Operation time to live: Forever

Round trip time (RTT)    Index 22
        Latest RTT: 432 ms
Latest operation start time: *01:15:37.327 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 131
Number of failures: 2
Operation time to live: Forever
```

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

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

   R1# **trace 209.165.200.254 source 192.168.1.1**

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

```
ISP1(config-if)#no shutdown
*Mar  1 00:11:07.027: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
*Mar  1 00:11:07.331: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
```

```
R2#
*Mar  1 01:13:05.571: RT: closer admin distance for 0.0.0.0, flushing 1 routes
*Mar  1 01:13:05.571: RT: NET-RED 0.0.0.0/0
*Mar  1 01:13:05.575: RT: SET_LAST_RDB for 0.0.0.0/0
  NEW rdb: via 209.165.201.1

*Mar  1 01:13:05.575: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]
*Mar  1 01:13:05.575: RT: NET-RED 0.0.0.0/0
*Mar  1 01:13:05.579: RT: default path is now 0.0.0.0 via 209.165.201.1
*Mar  1 01:13:05.579: RT: new default network 0.0.0.0
*Mar  1 01:13:05.579: RT: NET-RED 0.0.0.0/0
```

g. Again examine the IP SLA statistics.

```
R2#
*Mar  1 01:15:12.727: RT: NET-RED 0.0.0.0/0
R2#show ip sla monitor statistics
Round trip time (RTT)    Index 11
        Latest RTT: 88 ms
Latest operation start time: *01:15:40.719 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 140
Number of failures: 26
Operation time to live: Forever

Round trip time (RTT)    Index 22
        Latest RTT: 432 ms
Latest operation start time: *01:15:37.327 UTC Fri Mar 1 2002
Latest operation return code: OK
Number of successes: 131
Number of failures: 2
Operation time to live: Forever
```

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

h. Verify the routing table.

```
R2#
*Mar  1 01:16:12.727: RT: NET-RED 0.0.0.0/0
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial1/0
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```
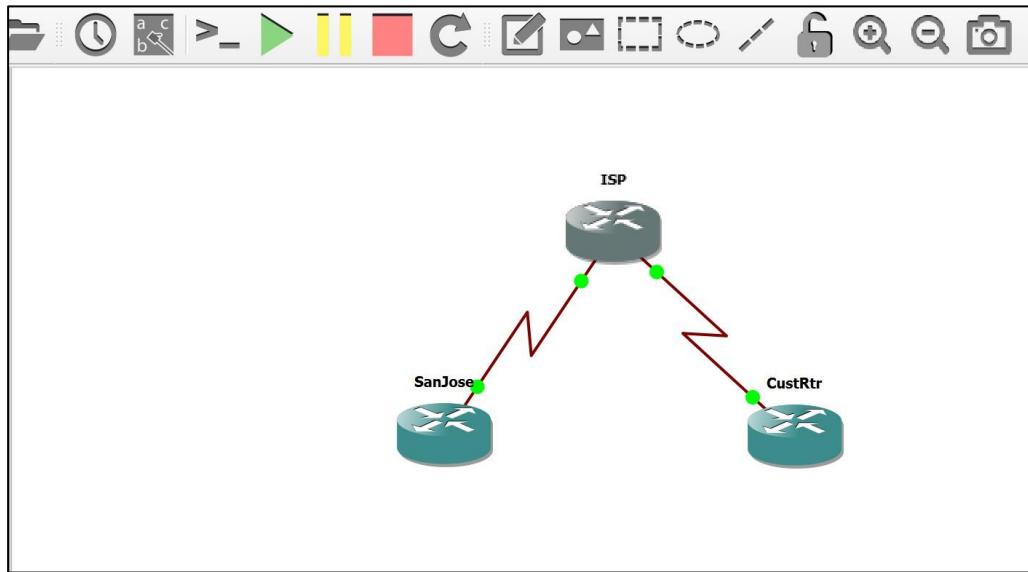
The default static through ISP1 with an administrative distance of 2 is re-established.

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.

Required Resources

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

Step 0: Suggested starting configurations.

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

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

Step 1: Configure interface addresses.

b. Using the addressing scheme in the diagram, create the loopback interfaces and apply IPv4 addresses to these and the serial interfaces on SanJose (R1), ISP (R2), and CustRtr (R3). The ISP loopbacks simulate real networks. Set a clock rate on the DCE serial interfaces.

```
SanJose(config)# interface Loopback0
SanJose(config-if)# ip address 10.1.1.1 255.255.255.0
SanJose(config-if)# exit
SanJose(config)# interface Serial0/0/0
SanJose(config-if)# ip address 192.168.1.5 255.255.255.252
SanJose(config-if)# clock rate 128000
SanJose(config-if)# no shutdown
SanJose(config-if)# end
SanJose#

ISP(config)# interface Loopback0
ISP(config-if)# ip address 10.2.2.1 255.255.255.0
ISP(config-if)# interface Serial0/0/0
ISP(config-if)# ip address 192.168.1.6 255.255.255.252
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface Serial0/0/1
ISP(config-if)# ip address 172.24.1.17 255.255.255.252
ISP(config-if)# clock rate 128000
ISP(config-if)# no shutdown
ISP(config-if)# end
ISP#

CustRtr(config)# interface Loopback0
CustRtr(config-if)# ip address 10.3.3.1 255.255.255.0
CustRtr(config-if)# exit
CustRtr(config)# interface Serial0/0/1
CustRtr(config-if)# ip address 172.24.1.18 255.255.255.252
CustRtr(config-if)# no shutdown
CustRtr(config-if)# end
CustRtr#
```

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

Step 2: Configure BGP.

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

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

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

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

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

```
*Mar  1 06:17:22.290: %SYS-5-CONFIG_I: Configured from console by console
SanJose#show ip bgp neighbors
BGP neighbor is 192.168.1.6,  remote AS 300, external link
  BGP version 4, remote router ID 10.2.2.1
  BGP state = Established, up for 00:08:18
  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:                1          3
    Keepalives:            10         10
    Route Refresh:          0          0
    Total:                 12         14
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 4/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                             Sent       Rcvd
  Prefix activity:           ----       ----
    Prefixes Current:           1          2 (Consumes 104 bytes)
    Prefixes Total:             1          2
    Implicit Withdraw:          0          0
    Explicit Withdraw:          0          0
    Used as bestpath:         n/a          2
    Used as multipath:        n/a          0

                           Outbound    Inbound
  Local Policy Denied Prefixes:    ------    ------
    AS_PATH loop:                   n/a          1
    Bestpath from this peer:          2        n/a
    Total:                            2          1
  Number of NLRIs in the update sent: max 1, min 1

  Connections established 1; dropped 0
  Last reset never
  Connection state is ESTAB, I/O status: 1, unread input bytes: 0
  Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 1
  Local host: 192.168.1.5, Local port: 12191
  Foreign host: 192.168.1.6, Foreign port: 179

  Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

  Event Timers (current time is 0x159F954):
  Timer          Starts    Wakeups         Next
  Retrans            13          0          0x0
  TimeWait            0          0          0x0
  AckHold            12         10          0x0
 --More--
```

Step 3: Remove the private AS.

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

b. Ping the 10.3.3.1 address from SanJose.

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

```
SanJose#PING
Protocol [ip]:
Target IP address: 10.3.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:  10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/70/156 ms
SanJose#
```

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

```
SanJose#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
SanJose#
```

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

```
ISP(config)# router bgp 300                                                    </>
ISP(config-router)# neighbor 192.168.1.5 remove-private-as
```

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

```
ISP# clear ip bgp *                                              </>
ISP#
*Sep  8 18:40:03.551: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Down User reset
*Sep  8 18:40:03.551: %BGP_SESSION-5-ADJCHANGE: neighbor 172.24.1.18 IPv4 Uni
*Sep  8 18:40:03.551: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down User reset
*Sep  8 18:40:03.551: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.5 IPv4 Uni
*Sep  8 18:40:04.515: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Up
*Sep  8 18:40:04.519: %BGP-
ISP#5-ADJCHANGE: neighbor 192.168.1.5 Up
ISP#

SanJose# show ip route

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.1.1.0/24 is directly connected, Loopback0
L        10.1.1.1/32 is directly connected, Loopback0
B        10.2.2.0/24 [20/0] via 192.168.1.6, 00:00:20
B        10.3.3.0/24 [20/0] via 192.168.1.6, 00:01:02
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.4/30 is directly connected, Serial0/0/0
L        192.168.1.5/32 is directly connected, Serial0/0/0
SanJose#
```

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

```
SanJose#show ip bgp
BGP table version is 9, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
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
SanJose#
```

Step 4: Use the AS_PATH attribute to filter routes.

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

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

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

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

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

c. Use the **clear ip bgp *** command to reset the routing information. Wait several seconds and then check the routing table for ISP. The route to 10.1.1.0 should be in the routing table.
Note: To force the local router to resend its BGP table, a less disruptive option is to use the **clear ip bgp * out** or **clear ip bgp * soft** command (the second command performs both outgoing and incoming route resync

```
ISP# clear ip bgp *
ISP#
*Sep  8 18:48:04.915: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Down User reset
*Sep  8 18:48:04.915: %BGP_SESSION-5-ADJCHANGE: neighbor 172.24.1.18 IPv4 Uni
*Sep  8 18:48:04.915: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down User reset
*Sep  8 18:48:04.915: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.1.5 IPv4 Uni
*Sep  8 18:48:04.951: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Up
*Sep  8 18:48:04.955: %BGP-
ISP#5-ADJCHANGE: neighbor 192.168.1.5 Up
ISP#

ISP# show ip route

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 192.168.1.5, 00:00:29
C       10.2.2.0/24 is directly connected, Loopback0
L       10.2.2.1/32 is directly connected, Loopback0
B       10.3.3.0/24 [20/0] via 172.24.1.18, 00:00:29
     172.24.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.24.1.16/30 is directly connected, Serial0/0/1
L       172.24.1.17/32 is directly connected, Serial0/0/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.4/30 is directly connected, Serial0/0/0
L       192.168.1.6/32 is directly connected, Serial0/0/0
ISP#
```

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

```
CustRtr#show ip route
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, Serial0/0
     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, 00:01:45
CustRtr#
```

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

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

f. Run the following Tcl script on all routers to verify whether there is connectivity. All pings from ISP should be successful. SanJose should not be able to ping the CustRtr loopback 10.3.3.1 or the WAN link 172.24.1.16/30. CustRtr should not be able to ping the SanJose loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

```
ISP#tclsh
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 = 8/44/136 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 = 24/32/52 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 = 8/17/32 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 = 28/43/76 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 = 60/92/128 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 = 20/27/32 ms
ISP(tcl)#
```

PRACTICAL 3

AIM: Configuring IBGP and EBGP Sessions, Local Preference, and MED
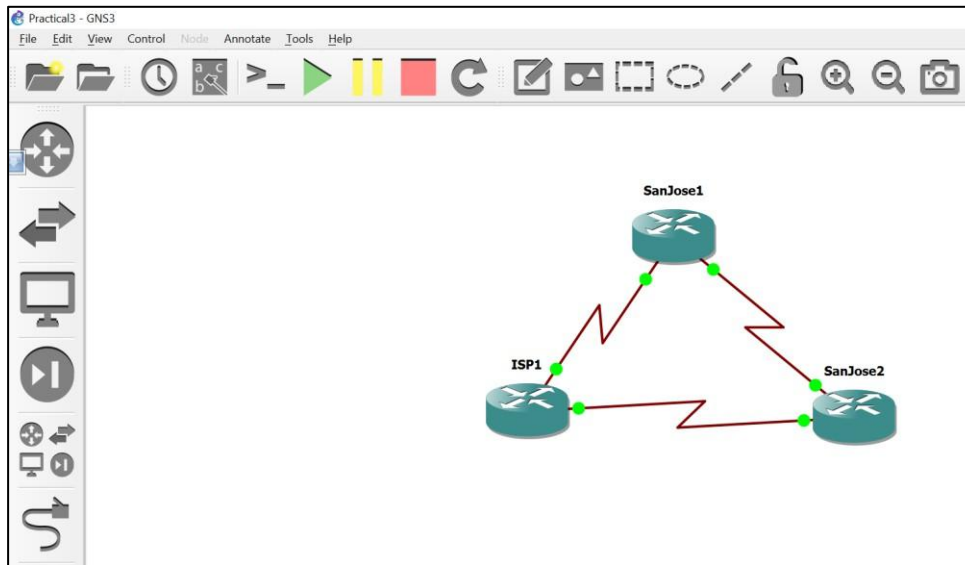
Objectives

- 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.

Required Resources

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

Step 1: Configure interface addresses.

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

Router R1 (hostname ISP)

ISP(config)# **interface Loopback0**
ISP(config-if)# **ip address 192.168.100.1 255.255.255.0**
ISP(config-if)# **exit**
 interface Serial0/0
 ip address 192.168.1.5 255.255.255.252
ISP(config-if)# **clock rate 128000**
ISP(config-if)# **no shutdown**
ISP(config-if)# **exit**
ISP(config)# **interface Serial1/0**
ISP(config-if)# **ip address 192.168.1.1 255.255.255.252**
ISP(config-if)# **no shutdown**
ISP(config-if)# **end**
ISP#

Router R2 (hostname SanJose1)

SanJose1(config)# **interface Loopback0**
SanJose1(config-if)# **ip address 172.16.64.1 255.255.255.0**
SanJose1(config-if)# **exit**
SanJose1(config)# **interface Serial0/0**
SanJose1(config-if)# **ip address 192.168.1.6 255.255.255.252**
SanJose1(config-if)# **no shutdown**
SanJose1(config-if)# **exit**
SanJose1(config)# **interface Serial1/0**
SanJose1(config-if)# **ip address 172.16.1.1 255.255.255.0**
SanJose1(config-if)# **clock rate 128000**
SanJose1(config-if)# **no shutdown**
SanJose1(config-if)# **end**
SanJose1#

Router R3 (hostname SanJose2)

SanJose2(config)# **interface Loopback0**
SanJose2(config-if)# **ip address 172.16.32.1 255.255.255.0**
SanJose2(config-if)# **exit**
SanJose2(config)# **interface Serial0/0**
SanJose2(config-if)# **ip address 192.168.1.2 255.255.255.252**
SanJose2(config-if)# **clock rate 128000**
SanJose2(config-if)# **no shutdown**
SanJose2(config-if)# **exit**
SanJose2(config)# **interface Serial1/0**
SanJose2(config-if)# **ip address 172.16.1.2 255.255.255.0**
SanJose2(config-if)# **no shutdown**
SanJose2(config-if)# **end**
SanJose2#

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

Step 2: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 routers.

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

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

Step 3: Configure IBGP and verify BGP neighbors.

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

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

If multiple pathways to the BGP neighbor exist, the router can use multiple IP interfaces to communicate with the neighbor. The source IP address therefore depends on the outgoing interface.

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

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

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

```
SanJose2#show ip bgp neighbors
BGP neighbor is 172.16.64.1, remote AS 64512, internal link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active
  Last read 00:00:52, last write 00:00:52, hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                      Sent       Rcvd
    Opens:              0          0
    Notifications:      0          0
    Updates:            0          0
    Keepalives:         0          0
    Route Refresh:      0          0
    Total:              0          0
  Default minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0/0
  Output queue size : 0
  Index 1, Offset 0, Mask 0x2
  1 update-group member
                      Sent       Rcvd
  Prefix activity:    ----       ----
    Prefixes Current:   0          0
    Prefixes Total:     0          0
    Implicit Withdraw:  0          0
    Explicit Withdraw:  0          0
    Used as bestpath:  n/a         0
    Used as multipath: n/a         0

                    Outbound   Inbound
  Local Policy Denied Prefixes: --------   -------
    Total:              0          0
  Number of NLRIs in the update sent: max 0, min 0

  Connections established 0; dropped 0
  Last reset never
  No active TCP connection
SanJose2#
SanJose2#
```

Step 4: Configure EBGP and verify BGP neighbors.

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

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

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

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

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

c. Configure SanJose1 as an EBGP peer to ISP.

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

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

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

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

Step 5: View BGP summary output.

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

Step 6: 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 reestablish with each SanJose router.

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

ISP# **ping 172.16.32.1**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
ISP# **ping 172.16.1.2**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/13/16 ms
ISP#

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

```
ISP1#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       192.168.1.6            0             0 64512 i
*> 192.168.100.0    0.0.0.0                0         32768 i
ISP1#
```

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

```
ISP1#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 = 16/29/36 ms
ISP1#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
UUUUU
Success rate is 0 percent (0/5)
ISP1#ping 172.16.1.2 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
.....
Success rate is 0 percent (0/5)
ISP1#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 = 16/22/32 ms
ISP1#ping
Protocol [ip]:
Target IP address:
% Bad IP address
ISP1#
```

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

ISP# **ping**
Protocol [ip]:
Target IP address: **172.16.64.1**
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: **y**
Source address or interface: **192.168.100.1**
Type of service [0]:

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

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

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

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

a. Issue the following commands on the ISP router.

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

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

```
*Mar  1 06:56:44.758: %SYS-5-CONFIG_I: Configured from console by console
ISP1#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       192.168.1.6            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
ISP1#
```

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

```
SanJose2#show ip route
Codes: 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, 3 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, Serial1/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
SanJose2#
```

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

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

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

```
SanJose2#show ip bgp
BGP table version is 2, 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
SanJose2#
*Mar  1 07:05:34.506: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to down
SanJose2#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, 2 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
SanJose2#
*Mar  1 07:06:28.522: %BGP-3-NOTIFICATION: sent to neighbor 192.168.1.1 4/0 (hold time expired) 0 bytes
SanJose2#
```

```
SanJose2#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, 2 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
SanJose2#
```

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

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

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

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

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

```
SanJose1#clear ip bgp * soft
SanJose1#show ip bgp
BGP table version is 4, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
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
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
SanJose1#
```

```
SanJose2#clear ip bgp * soft
SanJose2#show ip bgp
BGP table version is 2, 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
SanJose2#
```

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

```
ISP1#show ip bgp
BGP table version is 9, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       192.168.1.6              0            0 64512 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
ISP1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    172.16.0.0/16 [20/0] via 192.168.1.6, 00:13:28
     192.168.1.0/30 is subnetted, 2 subnets
C       192.168.1.0 is directly connected, Serial1/0
C       192.168.1.4 is directly connected, Serial0/0
C    192.168.100.0/24 is directly connected, Loopback0
ISP1#
```

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

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

Gateway of last resort is not set

       172.16.0.0/16 is variably subnetted, 6 subnets, 3 masks
S   172.16.0.0/16 is directly connected, Null0
C   172.16.1.0/24 is directly connected, Serial0/0/1

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

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

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

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

Gateway of last resort is not set

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

Step 8: Set BGP local preference.

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

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

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

SanJose2(config)# **route-map SECONDARY_T1_IN permit 10**
SanJose2(config-route-map)# **set local-preference 125**
SanJose1(config-route-map)# **exit**
SanJose2(config)# **router bgp 64512**
SanJose2(config-router)# **neighbor 192.168.1.1 route-map SECONDARY_T1_IN in**

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

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

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

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

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

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


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

Step 9: Set BGP MED.

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

```
ISP1#show ip bgp
BGP table version is 9, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
             r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       192.168.1.6              0             0 64512 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
ISP1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B    172.16.0.0/16 [20/0] via 192.168.1.6, 00:13:28
     192.168.1.0/30 is subnetted, 2 subnets
C       192.168.1.0 is directly connected, Serial1/0
C       192.168.1.4 is directly connected, Serial0/0
C    192.168.100.0/24 is directly connected, Loopback0
ISP1#
```

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

```
SanJose2#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
```

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

1. A ping that is sourced from 172.16.32.1 exits SanJose2 through s0/0/1, 172.16.1.2. It then arrives at the s0/0/1 interface for SanJose1.

2. SanJose1 S0/0/0, 192.168.1.6, routes the packet out to arrive at the S0/0/0 interface of ISP.

3. The target of 192.168.100.1 is reached: 192.168.100.1.

4. The packet is next forwarded out the S0/0/1, 192.168.1.1 interface for ISP and arrives at the S0/0/0 interface for SanJose2.

5. SanJose2 then forwards the packet out the last interface, loopback 0, 172.16.32.1.

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

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

SanJose1(config)#**route-map PRIMARY_T1_MED_OUT permit 10**
SanJose1(config-route-map)#**set Metric 50**
SanJose1(config-route-map)#**exit**
SanJose1(config)#**router bgp 64512**

SanJose1(config-router)#**neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out**

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

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

```
*Mar  1 07:32:34.890: %SYS-5-CONFIG_I: Configured from console by console
SanJose1#clear ip bgp * soft
SanJose1#show ip bgp
BGP table version is 4, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
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
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
SanJose1#
```

```
SanJose2#clear ip bgp * soft
SanJose2#show ip bgp
BGP table version is 2, 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
SanJose2#
```

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

```
SanJose2#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
```

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

```
ISP1#show ip bgp
BGP table version is 10, 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       192.168.1.6             50            0 64512 i
*> 192.168.100.0    0.0.0.0                  0        32768 i
ISP1#
```

Step 10: Establish a default route.

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

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

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

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

```
SanJose1#show ip route
Codes: 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 0.0.0.0

     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/0
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0
B    192.168.100.0/24 [20/0] via 192.168.1.5, 00:30:50
B*   0.0.0.0/0 [20/0] via 192.168.1.5, 00:01:15
SanJose1#
```

```
SanJose2#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, 3 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, Serial1/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
SanJose2#
```

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

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

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

SanJose2#

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

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

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

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

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

```
SanJose1#show ip route
Codes: 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 0.0.0.0

     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial1/0
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0
B    192.168.100.0/24 [20/0] via 192.168.1.5, 00:35:56
B*   0.0.0.0/0 [20/0] via 192.168.1.5, 00:06:20
SanJose1#
*Mar  1 07:46:33.830: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to down
*Mar  1 07:46:33.842: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down Interface flap
SanJose1#
```

```
SanJose2#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, 3 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, Serial1/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
SanJose2#
```

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

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

PRACTICAL 4

AIM: Secure the Management Plane
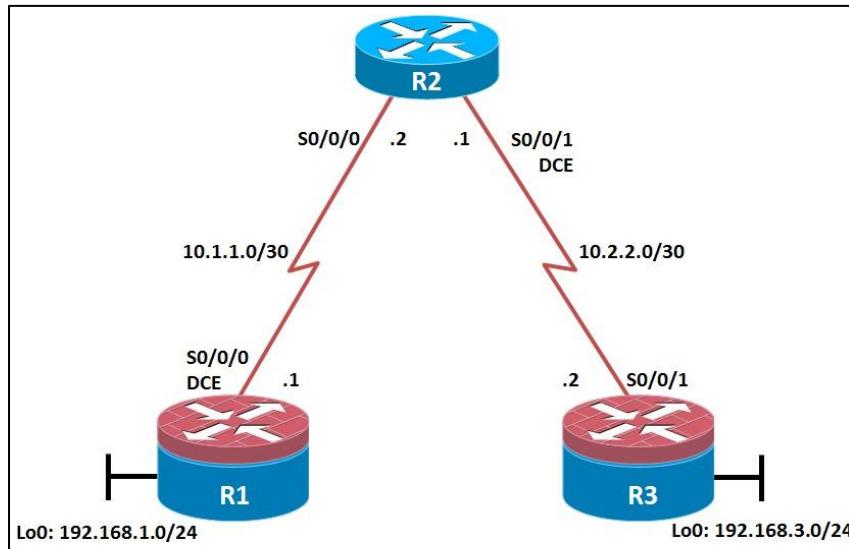
Objectives

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

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

R1

hostname R1

interface Loopback 0

description R1 LAN

ip address 192.168.1.1 255.255.255.0

exit

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

R2
```
hostname R2
!
interface Serial0/0/0
 description R2 --> R1
 ip address 10.1.1.2 255.255.255.252
 no shutdown
exit
interface Serial0/0/1
 description R2 --> R3
 ip address 10.2.2.1 255.255.255.252
 clock rate 128000
 no shutdown
exit
!
end
```

R3
```
hostname R3
!
```

interface Loopback0

interface Loopback0

description R3 LAN

ip address 192.168.3.1 255.255.255.0

exit

interface Serial0/0/1

 description R3 --> R2

 ip address 10.2.2.2 255.255.255.252

 no shutdown

exit

!

end

Step 2: Configure static routes.

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

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

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

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

c. On R2, configure two static routes.

R2(config)# **ip route 192.168.1.0 255.255.255.0 10.1.1.1**

R2(config)# **ip route 192.168.3.0 255.255.255.0 10.2.2.2**

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

```
1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
1(config)#interface Loopback 0
1(config-if)# description R1 LAN
1(config-if)# ip address 192.168.1.1 255.255.255.0
1(config-if)#exit
1(config)#!
1(config)#interface Serial0/0
1(config-if)# description R1 --> R2
1(config-if)# ip address 10.1.1.1 255.255.255.252
1(config-if)# clock rate 128000
1(config-if)# no shutdown
1(config-if)#exit
1(config)#!
1(config)#end
Mar  1 00:00:45.967: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up
1(config)#end
Mar  1 00:00:48.163: %LINK-3-UPDOWN: Interface Serial0/0, changed state to up
1(config)#end
Mar  1 00:00:49.167: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
1(config)#end
1#
Mar  1 00:01:02.523: %SYS-5-CONFIG_I: Configured from console by console
1#
Mar  1 00:01:11.911: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to down
1#
Mar  1 00:02:01.907: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
1(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2
1(config)#tclsh
                 ^
% Invalid input detected at '^' marker.

1(config)#
1#t
Mar  1 00:04:46.339: %SYS-5-CONFIG_I: Configured from console by console
1#tclsh
1(tcl)#foreach address {
>(tcl)#192.168.1.1
>(tcl)#10.1.1.1
>(tcl)#10.1.1.2
>(tcl)#10.2.2.1
>(tcl)#10.2.2.2
>(tcl)#192.168.3.1
>(tcl)#} { ping $address }

Type escape sequence to abort.
```

Step 3: Secure management access.

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

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

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

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

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

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

    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.

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

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

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

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

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

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

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

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

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

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

Step 4: Configure enhanced username password security.

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

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

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

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

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

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

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

R1# **telnet 10.2.2.2**
Trying 10.2.2.2 ... Open
Unauthorized access strictly prohibited!
User Access Verification
Username: **ADMIN**
Password:
R3>

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

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

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

- **Locally**: Users are authenticated against the local device database which is created using the username secret command. Sometimes referred to self-contained AAA.

- **Centrally**: A client-server model where users are authenticated against AAA servers. This provides improved scalability, manageability and control. Communication between the device and AAA servers is secured using either the RADIUS or TACACS+ protocols.

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

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

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

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

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

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

d. Assign both RADIUS servers to a server group.

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

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

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

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

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

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

R1(config)# **line vty 0 4**

R1(config-line)# **login authentication TELNET-LOGIN**

R1(config-line)# **exit**

R1(config)#

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

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

R1# **telnet 10.2.2.2**

Trying 10.2.2.2 ... Open

Unauthorized access strictly prohibited!

User Access Verification

Username: **admin**

Password:

% Authentication failed

Username: **ADMIN**

Password:

```
R1(config)#radius local
R1(config-radsrv)#address ?
% Unrecognized command
R1(config-radsrv)#?
Local RADIUS server configuration commands:
  exit   Exit from local radius server sub mode
  group  Configure client groups
  nas    Configure allowed Network Access Servers
  no     Negate a command or set its defaults
  user   Configure client usernames and passwords

R1(config-radsrv)#
R1#
*Mar  1 01:26:54.443: %SYS-5-CONFIG_I: Configured from console by console
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip domain-name ccnasecurity.com
R1(config)#crypto key zeroize rsa
% No Signature RSA Keys found in configuration.

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

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

R1(config)#
*Mar  1 01:27:58.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#ip ssh version 2
R1(config)#line vty 0 4
R1(config-line)#transport input ssh
R1(config-line)#end
R1#
*Mar  1 01:28:50.811: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
R1#ssh -l ADMIN 10.2.2.2

Password:

[Connection to 10.2.2.2 idle too long; timed out]
```

Step 6: Enabling secure remote management using SSH.

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

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

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

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

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

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

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

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

   R1(config)# **crypto key generate rsa general-keys modulus 1024**
   The name for the keys will be: R1.ccnasecurity.com
   % The key modulus size is 1024 bits
   % Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
   R1(config)#
   Jan 10 13:44:44.711: %SSH-5-ENABLED: SSH 1.99 has been enabled
   R1(config)#

d. Cisco routers support two versions of SSH:

   - **SSH version 1 (SSHv1)**: Original version but has known vulnerabilities.

   - **SSH version 2 (SSHv2)**: Provides better security using the Diffie-Hellman key exchange and the strong integrity-checking message authentication code (MAC).

 e. Configure SSH version 2 on R1.

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

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

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

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

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

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

    R1# **show ip ssh**

    SSH Enabled - version 2.0

    Authentication timeout: 120 secs; Authentication retries: 3

    Minimum expected Diffie Hellman key size : 1024 bits

    IOS Keys in SECSH format(ssh-rsa, base64 encoded):

    ssh-rsa

    AAAAB3NzaC1yc2EAAAADAQABAAAAgQC3Lehh7ReYlgyDzls6wq+mFzxqzoaZFr9XGx+Q/y
io

    dFYw00hQo80tZy1W1Ff3Pz6q7Qi0y00urwddHZ0kBZceZK9EzJ6wZ+9a87KKDETCWrGSLi6c8l
E/y4K+

    Z/oVrMMZk7bpTM1MFdP41YgkTf35utYv+TcqbsYo++KJiYk+xw==

    R1#

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

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

    R1# **ssh -l ADMIN 10.2.2.2**

    Password:

    Unauthorized access strictly prohibited!

    R3>
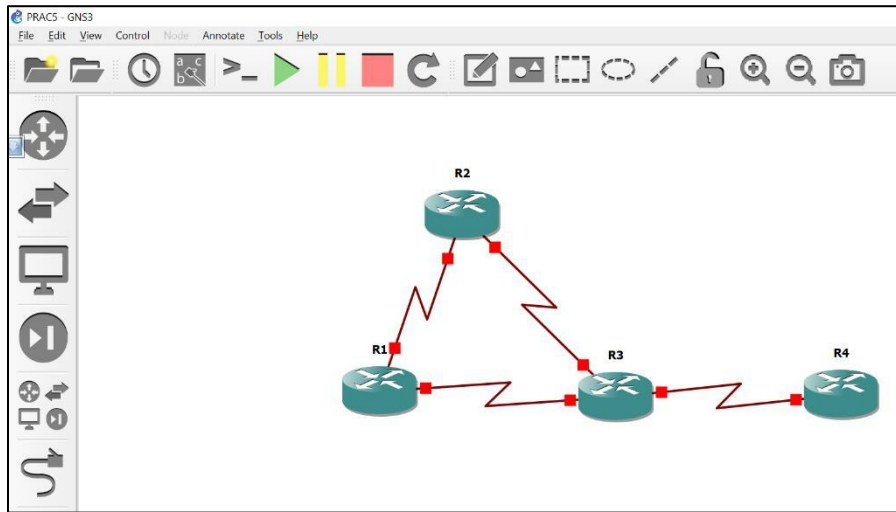
    R3> **en**

    Password:

    R3#

PRACTICAL NO 5

AIM: Configure and Verify 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.

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

b. 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.

Router R1

hostname R1

!

interface Lo1

 description R1 LAN

 ip address 192.168.1.1 255.255.255.0

```
!

interface Serial0/0

 description R1 --> R2

 ip address 172.16.12.1 255.255.255.248

 clock rate 128000

 bandwidth 128

 no shutdown

!

interface Serial1/0

 description R1 --> R3

 ip address 172.16.13.1 255.255.255.248

 bandwidth 64

 no shutdown

!

end
```

Router R2
```
hostname R2

!

interface Lo2

 description R2 LAN

 ip address 192.168.2.1 255.255.255.0

!

interface Serial0/0

 description R2 --> R1

 ip address 172.16.12.2 255.255.255.248

 bandwidth 128

no shutdown

interface Serial1/0

 description R2 --> R3
```

ip address 172.16.23.2 255.255.255.248

clock rate 128000

bandwidth 128

no shutdown

!

end

Router R3

hostname R3

!

interface Lo3

 description R3 LAN

 ip address 192.168.3.1 255.255.255.0

!

interface Serial0/0

 description R3 --> R1

 ip address 172.16.13.3 255.255.255.248

 clock rate 64000

 bandwidth 64

 no shutdown

!

interface Serial1/0

 description R3 --> R2

 ip address 172.16.23.3 255.255.255.248

 bandwidth 128

 no shutdown

!

interface Serial1/1

 description R3 --> R4

 ip address 172.16.34.3 255.255.255.248

clock rate 64000

bandwidth 64

no shutdown

!

end

Router R4

hostname R4

!

interface Lo4

 description R4 LAN A

 ip address 192.168.4.1 255.255.255.128

!

interface Lo5

 description R4 LAN B

 ip address 192.168.4.129 255.255.255.128

!

interface Serial0/0

 description R4 --> R3

 ip address 172.16.34.4 255.255.255.248

 bandwidth 64

 no shutdown

!

end

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

```
R1#
*Mar  1 00:08:25.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R1#show ip interface brief | include up
Serial0/0                  172.16.12.1     YES manual up                    up
Serial1/0                  172.16.13.1     YES manual up                    up
Loopback1                  192.168.1.1     YES manual up                    up
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 up
  Internet address is 172.16.12.1/29
FastEthernet0/1 is administratively down, line protocol is down
Serial1/0 is up, line protocol is up
  Internet address is 172.16.13.1/29
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
Loopback1 is up, line protocol is up
  Internet address is 192.168.1.1/24
R1#
R1#show interfaces description | include up
Se0/0               up             up       R1 --> R2
Se1/0               up             up       R1 --> R3
Lo1                 up             up       R1 LAN
R1#
```

```
*Mar  1 00:08:25.491: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
R2#show ip interface brief | include up
Serial0/0                  172.16.12.2     YES manual up                    up
Serial1/0                  172.16.23.2     YES manual up                    up
Loopback2                  192.168.2.1     YES manual up                    up
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
Serial1/0 is up, line protocol is up
  Internet address is 172.16.23.2/29
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
Loopback2 is up, line protocol is up
  Internet address is 192.168.2.1/24
R2#
R2#show interfaces description | include up
Se0/0               up             up       R2 --> R1
Se1/0               up             up       R2 --> R3
Lo2                 up             up       R2 LAN
R2#
```

```
*Mar  1 00:08:55.535: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up
R3#show ip interface brief | include up
Serial0/0                  172.16.13.3     YES manual up                    up
Serial1/0                  172.16.23.3     YES manual up                    up
Serial1/1                  172.16.34.3     YES manual up                    up
Loopback3                  192.168.3.1     YES manual up                    up
R3#show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
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
Serial1/0 is up, line protocol is up
  Internet address is 172.16.23.3/29
Serial1/1 is up, line protocol is up
  Internet address is 172.16.34.3/29
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 interfaces description | include up
Se0/0               up             up       R3 --> R1
Se1/0               up             up       R3 --> R2
Se1/1               up             up       R3 --> R4
Lo3                 up             up       R3 LAN
R3#
```

```
*Mar  1 00:08:43.911: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to up
R4#show ip interface brief | include up
Serial0/0                  172.16.34.4     YES manual up                    up
Loopback4                  192.168.4.1     YES manual up                    up
Loopback5                  192.168.4.129   YES manual up                    up
R4#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 172.16.34.4/29
FastEthernet0/1 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 | include up
Se0/0               up             up       R4 --> R3
Lo4                 up             up       R4 LAN A
Lo5                 up             up       R4 LAN B
R4#
```

Step 2: 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.

Router R1

router eigrp 1

 network 192.168.1.0

 network 172.16.12.0 0.0.0.7

 network 172.16.13.0 0.0.0.7

 no auto-summary

Router R2

router eigrp 1

 network 192.168.2.0

 network 172.16.12.0 0.0.0.7

 network 172.16.23.0 0.0.0.7

 no auto-summary

Router R3

router eigrp 1

 network 192.168.3.0

 network 172.16.13.0 0.0.0.7

 network 172.16.23.0 0.0.0.7

 network 172.16.34.0 0.0.0.7

 no auto-summary

Router R4

router eigrp 1

 network 192.168.4.0

 network 172.16.34.0 0.0.0.7

 no auto-summary

Step 3: Verify EIGRP connectivity.

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

```
*Mar  1 00:21:22.099: %SYS-5-CONFIG_I: Configured from console by console
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             Se1/0             12 00:02:16  109  2280  0  14
0   172.16.12.2             Se0/0             11 00:02:43   69  1140  0  13
R1#
```

```
*Mar  1 00:22:46.915: %SYS-5-CONFIG_I: Configured from console by console
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             Se1/0            11 00:06:08  105   1140  0  15
0   172.16.12.1             Se0/0            12 00:06:35   55   1140  0  12
R2#
```

```
*Mar  1 00:23:01.851: %SYS-5-CONFIG_I: Configured from console by console
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             Se1/1            12 00:06:03   86   3420  0  3
1   172.16.23.2             Se1/0            12 00:06:27   79   1140  0  12
0   172.16.13.1             Se0/0            13 00:06:28   88   2280  0  13
R3#
```

```
*Mar  1 00:22:59.739: %SYS-5-CONFIG_I: Configured from console by console
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            12 00:06:12   74   2280  0  13
R4#
```

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

R1# **tclsh**

foreach address {

172.16.12.1

172.16.12.2

172.16.13.1

172.16.13.3

172.16.23.2

172.16.23.3

172.16.34.3

172.16.34.4

192.168.1.1

192.168.2.1

192.168.3.1

192.168.4.1

192.168.4.129

} { ping $address }

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

Step 5: Verify the current path.

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

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

```
R1#show ip route | begin Gateway
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, 00:13:05, Serial1/0
D       172.16.23.0 [90/21024000] via 172.16.12.2, 00:13:06, Serial0/0
C       172.16.12.0 is directly connected, Serial0/0
C       172.16.13.0 is directly connected, Serial1/0
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:12:41, Serial1/0
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:12:41, Serial1/0
C    192.168.1.0/24 is directly connected, Loopback1
D    192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:13:05, Serial0/0
D    192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:13:05, 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.

```
*Mar  1 00:33:44.259: %SYS-5-CONFIG_I: Configured from console by console
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 4 msec 40 msec 76 msec
  2 172.16.23.2 144 msec 92 msec 120 msec
  3 172.16.12.1 112 msec 156 msec 216 msec
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 48 msec 88 msec 52 msec
  2 172.16.23.2 132 msec 132 msec 156 msec
  3 172.16.12.1 172 msec 164 msec 184 msec
R4#
```

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

```
R3#show ip route | begin Gateway
Gateway of last resort is not set

     172.16.0.0/29 is subnetted, 4 subnets
C       172.16.34.0 is directly connected, Serial1/1
C       172.16.23.0 is directly connected, Serial1/0
D       172.16.12.0 [90/21024000] via 172.16.23.2, 00:17:00, Serial1/0
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, 00:16:35, Serial1/1
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:16:35, Serial1/1
D    192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:17:00, Serial1/0
D    192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:17:00, Serial1/0
C    192.168.3.0/24 is directly connected, Loopback3
R3#
```

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

```
R3#show interfaces serial0/0
Serial0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Description: 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
     546 packets input, 38983 bytes, 0 no buffer
     Received 216 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     508 packets output, 35966 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#show interfaces serial0/0 | include BW
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
R3#show interfaces serial1/0 | include BW
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
R3#
```

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 (Serial1/0), 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#
```

Step 6: Configure PBR to provide path control.

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

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

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

The steps required to implement path control include the following:

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

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

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

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

   R3(config)# **ip access-list standard PBR-ACL**

   R3(config-std-nacl)# **remark ACL matches R4 LAN B traffic**

   R3(config-std-nacl)# **permit 192.168.4.128 0.0.0.127**

   R3(config-std-nacl)# **exit**

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

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

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

   R3(config-route-map)# **match ip address PBR-ACL**

   R3(config-route-map)# **set ip next-hop 172.16.13.1**

   R3(config-route-map)# **exit**

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

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

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

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

   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

Step 7: Test the policy.

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

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

R3# **conf t**

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

R3(config)# **access-list 1 permit 192.168.4.0 0.0.0.255**

R3(config)# **exit**

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

R3# **debug ip policy ?**

 <1-199> Access list

 dynamic dynamic PBR

 <cr>

R3# **debug ip policy 1**

Policy routing debugging is on for access list 1

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

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 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 36 msec 60 msec 60 msec
  2 172.16.23.2 168 msec 148 msec 84 msec
  3 172.16.12.1 196 msec 144 msec 124 msec
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 60 msec 44 msec 84 msec
  2 172.16.23.2 128 msec 188 msec 124 msec
  3 172.16.12.1 160 msec 168 msec 136 msec
R4#
```

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

```
R3#
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#
```

PRACTICAL 6

AIM: IP Service Level Agreements and Remote SPAN in a Campus Environment

Objectives

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

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

Router R1

hostname R1

interface Loopback 0

description R1 LAN

 ip address 192.168.1.1 255.255.255.0

interface Serial0/0/0

 description R1 --> ISP1

```
ip address 209.165.201.2 255.255.255.252

clock rate 128000

bandwidth 128

no shutdown

interface Serial0/0/1

 description R1 --> ISP2

 ip address 209.165.202.130 255.255.255.252

 bandwidth 128

 no shutdown
```

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0

 description Simulated Internet Web Server

 ip address 209.165.200.254 255.255.255.255

interface Loopback1

 description ISP1 DNS Server

 ip address 209.165.201.30 255.255.255.255

interface Serial0/0/0

 description ISP1 --> R1

 ip address 209.165.201.1 255.255.255.252

 bandwidth 128

 no shutdown

interface Serial0/0/1

 description ISP1 --> ISP2

 ip address 209.165.200.225 255.255.255.252

 clock rate 128000

 bandwidth 128

 no shutdown
```

Router ISP2 (R3)

hostname ISP2

interface Loopback0

 description Simulated Internet Web Server

 ip address 209.165.200.254 255.255.255.255

interface Loopback1

 description ISP2 DNS Server

 ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0

 description ISP2 --> R1

 ip address 209.165.202.129 255.255.255.252

 clock rate 128000

 bandwidth 128

 no shutdown

interface Serial0/0/1

 description ISP2 --> ISP1

 ip address 209.165.200.226 255.255.255.252

 bandwidth 128

 no shutdown

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

 R1# **show interfaces description | include up**

Se0/0/0   up   up   R1 --> ISP1

Se0/0/1   up   up   R1 --> ISP2

Lo0   up   up   R1 LAN

Step 2: Configure static routing.

The current routing policy in the topology is as follows:

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

a. Implement the routing policies on the respective routers.

Router R1

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

R1(config)#

Router ISP1 (R2)

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

ISP1(config-router)# **network 209.165.200.224 0.0.0.3**

ISP1(config-router)# **network 209.165.201.0 0.0.0.31**

ISP1(config-router)# **no auto-summary**

ISP1(config-router)# **exit**

ISP1(config)#

ISP1(config-router)# **ip route 192.168.1.0 255.255.255.0 209.165.201.2**

ISP1(config)#

Router ISP2 (R3)

ISP2(config)# **router eigrp 1**

ISP2(config-router)# **network 209.165.200.224 0.0.0.3**

ISP2(config-router)# **network 209.165.202.128 0.0.0.31**

ISP2(config-router)# **no auto-summary**

ISP2(config-router)# **exit**

ISP2(config)#

ISP2(config)# **ip route 192.168.1.0 255.255.255.0 209.165.202.130**

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

foreach address {

209.165.200.254

209.165.201.30

209.165.202.158

} {

ping $address source 192.168.1.1}

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

foreach address {

209.165.200.254

209.165.201.30

209.165.202.158

} {

trace $address source 192.168.1.1

}

Step 3: Configure IP SLA probes.

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

In this scenario, you will configure ICMP echo probes.

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

R1(config)# **ip sla 11**

R1(config-ip-sla)# **icmp-echo 209.165.201.30**

R1(config-ip-sla-echo)# **frequency 10**

R1(config-ip-sla-echo)# **exit**

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

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

R1# **show ip sla configuration 11**

IP SLAs Infrastructure Engine-III

Entry number: 11

Owner:

Tag:

Operation timeout (milliseconds): 5000

Type of operation to perform: icmp-echo

Target address/Source address: 209.165.201.30/0.0.0.0

Type Of Service parameter: 0x0

Request size (ARR data portion): 28

Verify data: No

Vrf Name:

Schedule:

==Operation frequency (seconds): 10== (not considered if randomly scheduled)

==Next Scheduled Start Time: Start Time already passed==

Group Scheduled : FALSE

Randomly Scheduled : FALSE

==Life (seconds): Forever==

Entry Ageout (seconds): never

Recurring (Starting Everyday): FALSE

Status of entry (SNMP RowStatus): Active

Threshold (milliseconds): 5000

Distribution Statistics:

Number of statistic hours kept: 2

Number of statistic distribution buckets kept: 1

Statistic distribution interval (milliseconds): 20

Enhanced History:

History Statistics:

Number of history Lives kept: 0

Number of history Buckets kept: 15

History Filter Type: None

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

R1# **show ip sla statistics**

IPSLAs Latest Operation Statistics

==IPSLA operation id: 11==

Latest RTT: 8 milliseconds

Latest operation start time: 10:33:18 UTC Sat Jan 10 2015

==Latest operation return code: OK==

==Number of successes: 51==

Number of failures: 0

<mark>Operation time to live: Forever</mark>

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

R1(config)# **ip sla 22**

R1(config-ip-sla)# **icmp-echo 209.165.202.158**

R1(config-ip-sla-echo)# **frequency 10**

R1(config-ip-sla-echo)# **exit**

R1(config)#

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

R1(config)# **end**

R1#

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

R1# **show ip sla configuration 22**

IP SLAs Infrastructure Engine-III

<mark>Entry number: 22</mark>

Owner:

Tag:

Operation timeout (milliseconds): 5000

Type of operation to perform: <mark>icmp-echo</mark>

<mark>Target address</mark>/Source address: <mark>209.165.202.158</mark>/0.0.0.0

Type Of Service parameter: 0x0

Request size (ARR data portion): 28

Verify data: No

Vrf Name:

Schedule:

<mark>Operation frequency (seconds): 10</mark> (not considered if randomly scheduled)

Next Scheduled Start Time: <mark>Start Time already passed</mark>

Group Scheduled : FALSE

Randomly Scheduled : FALSE

==Life (seconds): Forever==

Entry Ageout (seconds): never

Recurring (Starting Everyday): FALSE

Status of entry (SNMP RowStatus): Active

Threshold (milliseconds): 5000

Distribution Statistics:

Number of statistic hours kept: 2

Number of statistic distribution buckets kept: 1

Statistic distribution interval (milliseconds): 20

Enhanced History:

History Statistics:

Number of history Lives kept: 0

Number of history Buckets kept: 15

History Filter Type: None

R1# **show ip sla configuration 22**

IP SLAs, Infrastructure Engine-II.

Entry number: ==22==

Owner:

Tag:

Type of operation to perform: ==icmp-echo==

Target address/Source address: ==209.165.201.158==/0.0.0.0

Type Of Service parameter: 0x0

Request size (ARR data portion): 28

Operation timeout (milliseconds): 5000

Verify data: No

Vrf Name:

Schedule:

Operation frequency (seconds): ==10== (not considered if randomly scheduled)

Next Scheduled Start Time: <mark>Start Time already passed</mark>

Group Scheduled : FALSE

Randomly Scheduled : FALSE

Life (seconds): <mark>Forever</mark>

Entry Ageout (seconds): never

Recurring (Starting Everyday): FALSE

Status of entry (SNMP RowStatus): Active

Threshold (milliseconds): 5000 (not considered if react RTT is configured)

Distribution Statistics:

Number of statistic hours kept: 2

Number of statistic distribution buckets kept: 1

Statistic distribution interval (milliseconds): 20

History Statistics:

Number of history Lives kept: 0

Number of history Buckets kept: 15

History Filter Type: None

Enhanced History:

R1# **show ip sla statistics 22**

IPSLAs Latest Operation Statistics

<mark>IPSLA operation id: 22</mark>

Latest RTT: 16 milliseconds

Latest operation start time: 10:38:29 UTC Sat Jan 10 2015

<mark>Latest operation return code: OK</mark>

<mark>Number of successes: 82</mark>

Number of failures: 0

<mark>Operation time to live: Forever</mark>

Step 4: Configure tracking options.

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

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

R1(config)# **no ip route 0.0.0.0 0.0.0.0 209.165.201.1**

R1(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.1 5**

R1(config)# **exit**

B. Verify the routing table.

R1# **show ip route | begin Gateway**

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

S*   0.0.0.0/0 [5/0] via 209.165.201.1

       192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C   192.168.1.0/24 is directly connected, Loopback0

L   192.168.1.1/32 is directly connected, Loopback0

       209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C   209.165.201.0/30 is directly connected, Serial0/0/0

L   209.165.201.2/32 is directly connected, Serial0/0/0

       209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C   209.165.202.128/30 is directly connected, Serial0/0/1

L   209.165.202.130/32 is directly connected, Serial0/0/1

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

R1(config)# **track 1 ip sla 11 reachability**

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

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

R1(config-track)# **exit**

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

R1# **debug ip routing**

IP routing debugging is on

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

references IP SLA operation number 11.

references IP SLA operation number 11.

R1(config)# **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1**

R1(config)#

Jan 10 10:45:39.119: RT: <mark>updating static 0.0.0.0/0</mark> (0x0) :

 via 209.165.201.1 0 1048578

Jan 10 10:45:39.119: RT: <mark>closer admin distance for 0.0.0.0</mark>, <mark>flushing 1 routes</mark>

Jan 10 10:45:39.119: RT: <mark>add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]</mark>

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :

 via 209.165.201.1   0 1048578

Jan 10 10:45:39.119: RT: rib update return code: 17

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0) :

 via 209.165.201.1 0 1048578

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

R1(config)# **track 2 ip sla 22 reachability**

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

R1(config-track)# **exit**

R1(config)# **ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2**

h. Verify the routing table again.

R1#show ip route | begin Gateway

<mark>Gateway of last resort is 209.165.201.1 to network 0.0.0.0</mark>

<mark>S*   0.0.0.0/0 [2/0] via 209.165.201.1</mark>

 192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.1.0/24 is directly connected, Loopback0

L 192.168.1.1/32 is directly connected, Loopback0

 209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C 209.165.201.0/30 is directly connected, Serial0/0/0

L 209.165.201.2/32 is directly connected, Serial0/0/0

 209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C 209.165.202.128/30 is directly connected, Serial0/0/1

L   209.165.202.130/32 is directly connected, Serial0/0/1

Step 5: Verify IP SLA operation.

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

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

a. On ISP1, disable the loopback interface 1.

ISP1(config-if)# **int lo1**

ISP1(config-if)# **shutdown**

Jan 10 10:53:26.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to down

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

Jan 10 10:53:59.551: %TRACK-6-STATE: 1 ip sla 11 reachability Up -> Down

Jan 10 10:53:59.551: RT: del 0.0.0.0 via 209.165.201.1, static metric [2/0]

Jan 10 10:53:59.551: RT: delete network route to 0.0.0.0/0

Jan 10 10:53:59.551: RT: default path has been cleared

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

   via 209.165.202.129   0 1048578

Jan 10 10:53:59.551: RT: add 0.0.0.0/0 via 209.165.202.129, static metric [3/0]

Jan 10 10:53:59.551: RT: default path is now 0.0.0.0 via 209.165.202.129

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

   via 209.165.201.1   0 1048578

Jan 10 10:53:59.551: RT: rib update return code: 17

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

   via 209.165.202.129   0 1048578

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

   via 209.165.201.1   0 1048578

Jan 10 10:53:59.551: RT: rib update return code: 17

c. On R1, verify the routing table.

R1# **show ip route | begin Gateway**

<mark>Gateway of last resort is 209.165.202.129 to network 0.0.0.0</mark>

<mark>S*   0.0.0.0/0 [3/0] via 209.165.202.129</mark>

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C   192.168.1.0/24 is directly connected, Loopback0

L   192.168.1.1/32 is directly connected, Loopback0

209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C   209.165.201.0/30 is directly connected, Serial0/0/0

L   209.165.201.2/32 is directly connected, Serial0/0/0

209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C   209.165.202.128/30 is directly connected, Serial0/0/1

L   209.165.202.130/32 is directly connected, Serial0/0/1

d. Verify the IP SLA statistics.

R1# **show ip sla statistics**

IPSLAs Latest Operation Statistics

IPSLA operation id: 11

       Latest RTT: NoConnection/Busy/Timeout

Latest operation start time: 11:01:08 UTC Sat Jan 10 2015

<mark>Latest operation return code: Timeout</mark>

Number of successes: 173

<mark>Number of failures: 45</mark>

Operation time to live: Forever

IPSLA operation id: 22

       Latest RTT: 8 milliseconds

Latest operation start time: 11:01:09 UTC Sat Jan 10 2015

Latest operation return code: OK

Number of successes: 218

Number of failures: 0

Operation time to live: Forever

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

R1# **trace 209.165.200.254 source 192.168.1.1**

Type escape sequence to abort.

Tracing the route to 209.165.200.254

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

  1 209.165.202.129 4 msec * *

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

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

ISP1(config-if)# **no shutdown**

Jan 10 11:05:46.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up

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

R1#

Jan 10 11:06:20.551: %TRACK-6-STATE: 1 ip sla 11 reachability Down -> Up

Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0) :

   via 209.165.201.1   0 1048578

Jan 10 11:06:20.551: RT: closer admin distance for 0.0.0.0, flushing 1 routes

Jan 10 11:06:20.551: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]

Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0) :

   via 209.165.202.129   0 1048578

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

Jan 10 11:06:20.551: RT: u

R1#pdating static 0.0.0.0/0 (0x0) :

   via 209.165.202.129   0 1048578

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

Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0) :

   via 209.165.201.1   0 1048578

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

g. Again examine the IP SLA statistics.

R1# **show ip sla statistics**

IPSLAs Latest Operation Statistics

IPSLA operation id: 11

Latest RTT: 8 milliseconds

Latest operation start time: 11:07:38 UTC Sat Jan 10 2015

Latest operation return code: OK

Number of successes: 182

Number of failures: 75

Operation time to live: Forever

IPSLA operation id: 22

Latest RTT: 16 milliseconds

Latest operation start time: 11:07:39 UTC Sat Jan 10 2015

Latest operation return code: OK

Number of successes: 257

Number of failures: 0

Operation time to live: Forever

h. Verify the routing table.

R1# **show ip route | begin Gateway**

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

S*   0.0.0.0/0 [2/0] via 209.165.201.1

    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C   192.168.1.0/24 is directly connected, Loopback0

L   192.168.1.1/32 is directly connected, Loopback0

    209.165.201.0/24 is variably subnetted, 2 subnets, 2 masks

C   209.165.201.0/30 is directly connected, Serial0/0/0

L   209.165.201.2/32 is directly connected, Serial0/0/0

    209.165.202.0/24 is variably subnetted, 2 subnets, 2 masks

C   209.165.202.128/30 is directly connected, Serial0/0/1 L

209.165.202.130/32 is directly connected, Serial0/0/1 R1#

The default static through ISP1 with an administrative distance of 2 is reestablished.

There are many possibilities available with object tracking and Cisco IOS IP SLAs. As shown in this lab, a probe can be based on reachability, changing routing operations, and path control based on the ability to reach an object. However, Cisco IOS IP SLAs also allow paths to be changed based on network conditions such as delay, load, and other factors.

Before deploying a Cisco IOS IP SLA solution, the impact of the additional probe traffic being generated should be considered, including how that traffic affects bandwidth utilization, and congestion levels. Tuning the configuration (for example, with the **delay** and **frequency** commands) is critical to mitigate possible issues related to excessive transitions and route changes in the presence of flapping tracked objects.

The benefits of running IP SLAs should be carefully evaluated. The IP SLA is an additional task that must be performed by the router's CPU. A large number of intensive SLAs could be a significant burden on the CPU, possibly interfering with other router functions and having detrimental impact on the overall router performance. The CPU load should be monitored after the SLAs are deployed to verify that they do not cause excessive utilization of the router CPU.

PRACTICAL 7

AIM: Inter-VLAN Routing
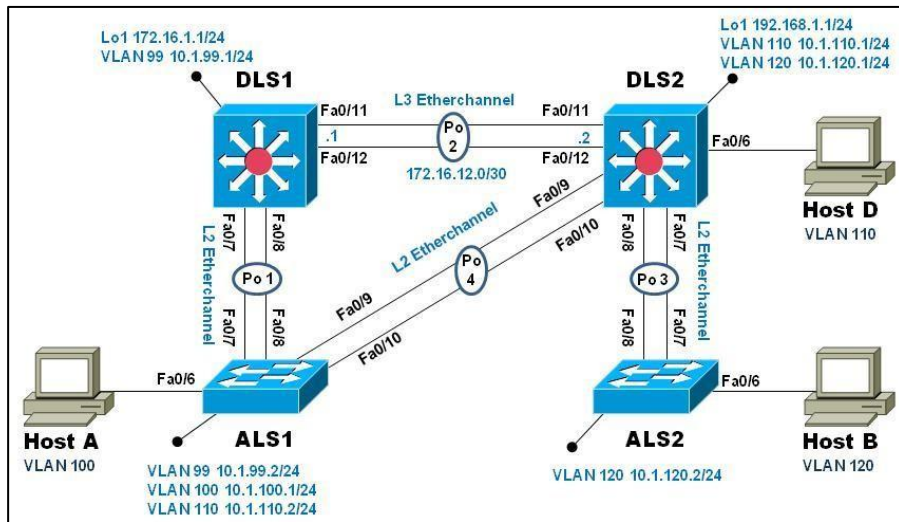
Objectives

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

Required Resources

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

Part 1:   Configure Multilayer Switching using Distribution Layer Switches

Step 1:   Load base config

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

DLS1# **tclsh reset.tcl**

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

[OK]

Erase of nvram: complete

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

Proceed with reload? [confirm]

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

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

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

Switch> **en**

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

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

Destination filename [running-config]?

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

DLS1#

Step 2:   Verify switch management database configuration

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

ALS1# **sho sdm pref**

 The current template is "default" template.

ALS1# **conf t**

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

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

Changes to the running SDM preferences have been stored, but cannot take effect

until the next reload.

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

ALS1(config)# **end**

ALS1# **reload**

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

*Mar 1 02:12:00.699: %SYS-5-CONFIG_I: Configured from console by console

Building configuration...

[OK]

Proceed with reload? [confirm]

Step 3:   Configure layer 3 interfaces on the DLS switches

Enable IP Routing, create broadcast domains (VLANs), and configure the DLS switches with the layer 3 interfaces and addresses shown:
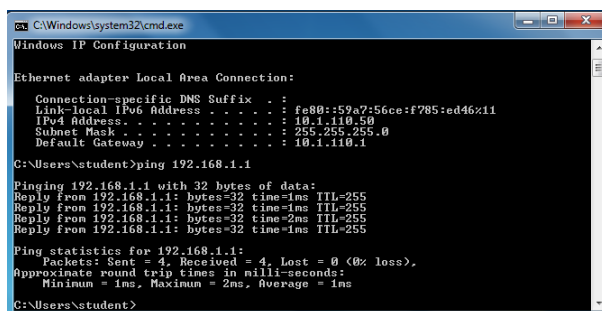
Switch	Interface	Address/Mask
DLS1	VLAN 99	10.1.99.1/24
DLS1	Loopback 1	172.16.1.1/24
DLS2	VLAN 110	10.1.110.1/24

An example from DLS2:

DLS2(config)# **ip routing**

DLS2(config)# **vlan 110**

DLS2(config-vlan)# **name Management**

DLS2(config-vlan)# **exit**

DLS2(config)# **vlan 120**

DLS2(config-vlan)# **name Local**

DLS2(config-vlan)# **exit**

DLS2(config)# **int vlan 110**

DLS2(config-if)# **ip address 10.1.110.1 255.255.255.0**

DLS2(config-if)# **no shut**

DLS2(config-if)# **exit**

DLS2(config)# **int vlan 120**

DLS2(config-if)# **ip address 10.1.120.1 255.255.255.0**

DLS2(config-if)# **no shut**

DLS2(config-if)# **exit**

DLS2(config)# **int loopback 1**

DLS2(config-if)# **ip address 192.168.1.1 255.255.255.0**

DLS2(config-if)# **no shut**

DLS2(config-if)# **exit**

DLS2(config)#

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

DLS2(config)# **int f0/6**

DLS2(config-if)# **switchport host**

switchport mode will be set to access

spanning-tree portfast will be enabled

channel group will be disabled

DLS2(config-if)# **switchport access vlan 110**

DLS2(config-if)# **no shut**

DLS2(config-if)# **exit**

DLS2(config)#

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

Now you will interconnect the multilayer switches in preparation to demonstrate other routing capabilities. Configure a layer 3 EtherChannel between the DLS switches. This will provide the benefit

of increased available bandwidth between the two multilayer switches. To convert the links from layer 2 to layer 3, issue the **no switchport** command.

DLS1	172.16.12.1/30	DLS2	172.16.12.2/30

Example from DLS1:

DLS1(config)# **interface range f0/11-12**

DLS1(config-if-range)# **no switchport**

DLS1(config-if-range)# **channel-group 2 mode desirable**

Creating a port-channel interface Port-channel 2

DLS1(config-if-range)# **no shut**

DLS1(config-if-range)# **exit**

DLS1(config)# **interface port-channel 2**

DLS1(config-if)# **ip address 172.16.12.1 255.255.255.252**

DLS1(config-if)# **no shut**

DLS1(config-if)# **exit**

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

DLS2# **show etherchannel summary**

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

     I - stand-alone s - suspended

     H - Hot-standby (LACP only)

     R - Layer3   S - Layer2

     U - in use   f - failed to allocate aggregator

     M - not in use, minimum links not met

     u - unsuitable for bundling

     w - waiting to be aggregated

     d - default port

Number of channel-groups in use: 1

Number of aggregators: 1

Group Port-channel Protocol   Ports

--------+---------------+------------+--

2   Po2(RU)   PAgP   Fa0/11(P)   Fa0/12(P)

DLS2# **ping 172.16.12.1**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 1/3/9 ms

DLS2#

Step 5:   Configure default routing between DLS switches

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

DLS1(config)# **ip route 0.0.0.0 0.0.0.0 port-channel 2**

%Default route without gateway, if not a point-to-point interface, may impact performance

DLS1(config)# **ip route 0.0.0.0 0.0.0.0 port-channel 2 172.16.12.2**

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

DLS2# **show ip route**

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

Gateway of last resort is 172.16.12.1 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 172.16.12.1, Port-channel2

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

C   10.1.110.0/24 is directly connected, Vlan110

L   10.1.110.1/32 is directly connected, Vlan110

      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

C   172.16.12.0/30 is directly connected, Port-channel2

L   172.16.12.2/32 is directly connected, Port-channel2

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C   192.168.1.0/24 is directly connected, Loopback1

L   192.168.1.1/32 is directly connected, Loopback1

DLS2# **ping 172.16.1.1**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/9 ms

DLS2#

Step 6:   Configure the remaining EtherChannels for the topology

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

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

Example from ALS1:

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

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

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

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

Creating a port-channel interface Port-channel 1

ALS1(config-if-range)# **no shut**

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

ALS1(config)# **interface range f0/9-10**

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

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

ALS1(config-if-range)# **channel-group 4 mode desirable**

Creating a port-channel interface Port-channel 4

ALS1(config-if-range)# **no shut**

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

ALS1(config)#end

ALS1# **show etherchannel summary**

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

     I - stand-alone s - suspended

     H - Hot-standby (LACP only)

     R - Layer3   S - Layer2

     U - in use   f - failed to allocate aggregator

     M - not in use, minimum links not met

     u - unsuitable for bundling

     w - waiting to be aggregated

     d - default port

Number of channel-groups in use: 2

Number of aggregators: 2

Group Port-channel Protocol   Ports

-------+---------------+------------+---

1   Po1(SU)   PAgP   Fa0/7(P)   Fa0/8(P)

4   Po4(SU)   PAgP   Fa0/9(P)   Fa0/10(P)

ALS1# **show interface trunk**

Port   Mode   Encapsulation Status   Native vlan

Po1   on   802.1q   trunking   1

Po4   on   802.1q   trunking   1

Port   Vlans allowed on trunk

Po1   1-109,111-4094

Po4   110

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

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

- Create the management VLANs (99 at ALS1, 120 at ALS2)

- Configure interface VLAN 99 at ALS1 and interface VLAN 120 at ALS2

- Assign addresses (refer to the diagram) and default gateways (at DLS1/DLS2 respectively).

ALS2(config)# **vlan 120**

ALS2(config-vlan)# **name Management**

ALS2(config-vlan)# **exit**

ALS2(config)# **int vlan 120**

ALS2(config-if)# **ip address 10.1.120.2 255.255.255.0**

ALS2(config-if)# **no shut**

ALS2(config-if)# **exit**

ALS2(config)# **ip default-gateway 10.1.120.1**

ALS2(config)# **end**

ALS2# **ping 10.1.99.2**

Type escape sequence to abort.

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

..!!!

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

ALS2#

ALS2# **traceroute 10.1.99.2**

Type escape sequence to abort.

Tracing the route to 10.1.99.2

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

  1 10.1.120.1 0 msec 0 msec 8 msec

  2 172.16.12.1 0 msec 0 msec 8 msec

  3 10.1.99.2 0 msec 0 msec *

Part 2:   Configure Multilayer Switching at ALS1

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

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

- Enable intervlan routing between two VLANs locally at ALS1

- Enable IP Routing

- Configure a static route for DLS2's Lo1 network travel via Port-Channel 4.

Step 1:   Configure additional VLANs and VLAN interfaces

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

ALS1(config)# **ip routing**

ALS1(config)# **vlan 100**

ALS1(config-vlan)# **name Local**

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

ALS1(config)# **vlan 110**

ALS1(config-vlan)# **name InterNode**

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

ALS1(config)# **int vlan 100**

ALS1(config-if)# **ip address 10.1.100.1 255.255.255.0**

ALS1(config-if)# **no shut**

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

ALS1(config)# **int vlan 110**

ALS1(config-if)# **ip address 10.1.110.2 255.255.255.0**

ALS1(config-if)# **no shut**

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

Step 2: Configure and test Host Access

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

ALS1(config)# **interface f0/6**

ALS1(config-if)# **switchport host**

switchport mode will be set to access

spanning-tree portfast will be enabled

channel group will be disabled

ALS1(config-if)# **switchport access vlan 100**

ALS1(config-if)# **no shut**

ALS1(config-if)# **exit**
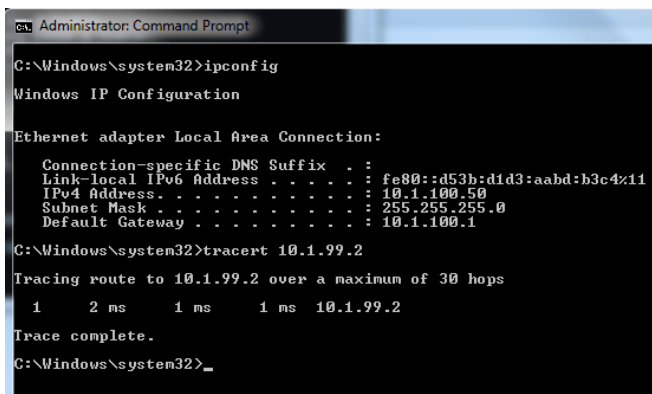
Step 3: Configure and verify static routing across the network

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

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

ALS1(config)# **ip route 192.168.1.0 255.255.255.0 vlan 110**

ALS1(config)# **ip route 0.0.0.0 0.0.0.0 10.1.99.1**

ALS1(config)# **end**

ALS1# **show ip route**

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

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

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

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

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

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

    o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

    + - replicated route, % - next hop override

Gateway of last resort is 10.1.99.1 to network 0.0.0.0

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

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

C   10.1.99.0/24 is directly connected, Vlan99

L   10.1.99.2/32 is directly connected, Vlan99

C   10.1.100.0/24 is directly connected, Vlan100 L

10.1.100.1/32 is directly connected, Vlan100 C

10.1.110.0/24 is directly connected, Vlan110 L

10.1.110.2/32 is directly connected, Vlan110 S

192.168.1.0/24 is directly connected, Vlan110

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

From ALS1, a traceroute to 10.1.120.2 should take three hops

ALS1# **traceroute 10.1.120.2**

Type escape sequence to abort.

Tracing the route to 10.1.120.2

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

  1 10.1.99.1 0 msec 0 msec 0 msec

  2 172.16.12.2 9 msec 0 msec 0 msec

  3 10.1.120.2 0 msec 8 msec *

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

ALS1# **traceroute 192.168.1.1**

Type escape sequence to abort.
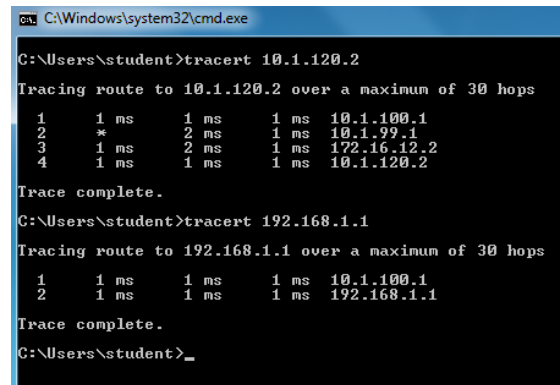
Tracing the route to 192.168.1.1

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

  1 10.1.110.1 0 msec 0 msec *

ALS1#

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


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

C:\Users\student>tracert 10.1.120.2

Tracing route to 10.1.120.2 over a maximum of 30 hops

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

Trace complete.

C:\Users\student>tracert 192.168.1.1

Tracing route to 192.168.1.1 over a maximum of 30 hops

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

Trace complete.

C:\Users\student>_
```

Step 4: End of Lab

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

PRACTICAL 9

AIM: Simulating VRF

```
CUSTOMER-3                    100:3          Fa1/2
#conf t
er configuration commands, one per line.  End with CNTL/Z.
(config)#interface FastEthernet1/3
(config-if)#ip vrf forwarding CUSTOMER-1
(config-if)#
#conf t
r 1 00:16:11.955: %SYS-5-CONFIG_I: Configured from console by console
# show ip vrf
ame                          Default RD     Interfaces
CUSTOMER-1                    100:1          Fa0/0
                                            Fa1/0
                                            Fa1/3
CUSTOMER-2                    100:2          Fa1/1
CUSTOMER-3                    100:3          Fa1/2
#conf t
er configuration commands, one per line.  End with CNTL/Z.
(config)#interface FastEthernet1/4
(config-if)#ip vrf forwarding CUSTOMER-2
(config-if)#
# show ip vrf
r 1 00:16:49.279: %SYS-5-CONFIG_I: Configured from console by console
# show ip vrf
ame                          Default RD     Interfaces
CUSTOMER-1                    100:1          Fa0/0
                                            Fa1/0
                                            Fa1/3
CUSTOMER-2                    100:2          Fa1/1
                                            Fa1/4
CUSTOMER-3                    100:3          Fa1/2
#conf t
er configuration commands, one per line.  End with CNTL/Z.
(config)#interface FastEthernet1/5
(config-if)#ip vrf forwarding CUSTOMER-3
(config-if)#
# show ip vrf
r 1 00:17:24.979: %SYS-5-CONFIG_I: Configured from console by console
# show ip vrf
ame                          Default RD     Interfaces
CUSTOMER-1                    100:1          Fa0/0
                                            Fa1/0
                                            Fa1/3
CUSTOMER-2                    100:2          Fa1/1
                                            Fa1/4
CUSTOMER-3                    100:3          Fa1/2
                                            Fa1/5
#
```



```
CUST-R1#
CUST-R1#
CUST-R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
CUST-R1(config)#router ospf 1
CUST-R1(config-router)#
*Mar  1 00:08:15.851: %OSPF-4-NORTRID: OSPF process 1 cannot pick a router-id.
  Please configure manually or bring up an interface with an ip address.
CUST-R1(config-router)#network 0.0.0.0 255.255.255.255 area 0
CUST-R1(config-router)#
```

PRACTICAL 10A

AIM: Simulating SDN with Open Daylight SDN Controller with the Mininet Network Emulator

We see that interface **enp0s8** has no IP address. This is the second network adapter connected to **vboxnet0**. VirtualBox can assign an IP address on this interface using DHCP if the DCHP client requests it. So, run the following command to set up interface **enp0s8**:

```
brian@odl:~$ sudo dhclient enp0s8
```


Now we see the VirtualBox DHCP server connected to the host-only network assigned the IP address **192.168.56.101** to this interface. This is the IP address we should use when connecting to any application running on the VM.

Install java

```
$ sudo apt-get update

$ sudo apt-get install default-jre-headless

wget https://nexus.opendaylight.org/content/groups/public/org/opendaylight/integration/distribut
ion-karaf/0.4.0-Beryllium/distribution-karaf-0.4.0-Beryllium.tar.gz
```

To run OpenDaylight, run the *karaf* command inside the package distribution folder.

```
brian@odl:~$ cd distribution-karaf-0.4.0-Beryllium

brian@odl:~$ ./bin/karaf
```

Now the OpenDaylight controller is running.