Diplomado de Profundización CISCO CCNP
Paso 3: Actividad colaborativa 2

Presentado por:
MARILIN VELASCO VELEZ

Presentado a:

GERARDO GRANADOS ACUÑA

Grupo colaborativo:

208014-1

Universidad Nacional Abierta y a Distancia
Vicerrectoría Académica y de Investigación

TABLA DE CONTENIDO

INTRODUCCIÓN

Los laboratorios utilizan un modelo de Internet para analizar datos reales sin afectar el desempeño de la red. Actividades en el software Packet Tracer ayudan a analizar la operación de la red y de diferentes protocolos, además de construir pequeños ambientes simulados de redes. Como estudiantes construimos topologías LAN aplicando los conceptos básicos de cableado, realizando configuraciones en dispositivos como switches y routers, así como la implementación de esquemas de direccionamiento.

Este trabajo tiene la aplicación de la teoría de CCNP ROUTE en los diferentes laboratorios en donde se puede evidenciar el funcionamiento de los diferentes protocolos tanto a nivel de IPv6 como IPv4.

Simultaneamente que leemos el módulo de CCNP estamos realizando los laboratorios sugeridos en la plataforma para consolidar el conocimiento en el área de redes de telecomunicaciones.

OBJETIVOS

Objetivo general:

- Identificar problemas de conectividad en entornos corporativos y residenciales

Objetivos específicos:

• Implementar Path Control

• Implementación de una solución Border Gateway Protocolo (BGP) para conectividad ISP

• Implementación de instalaciones de enrutamiento para sucursales y trabajadores móviles

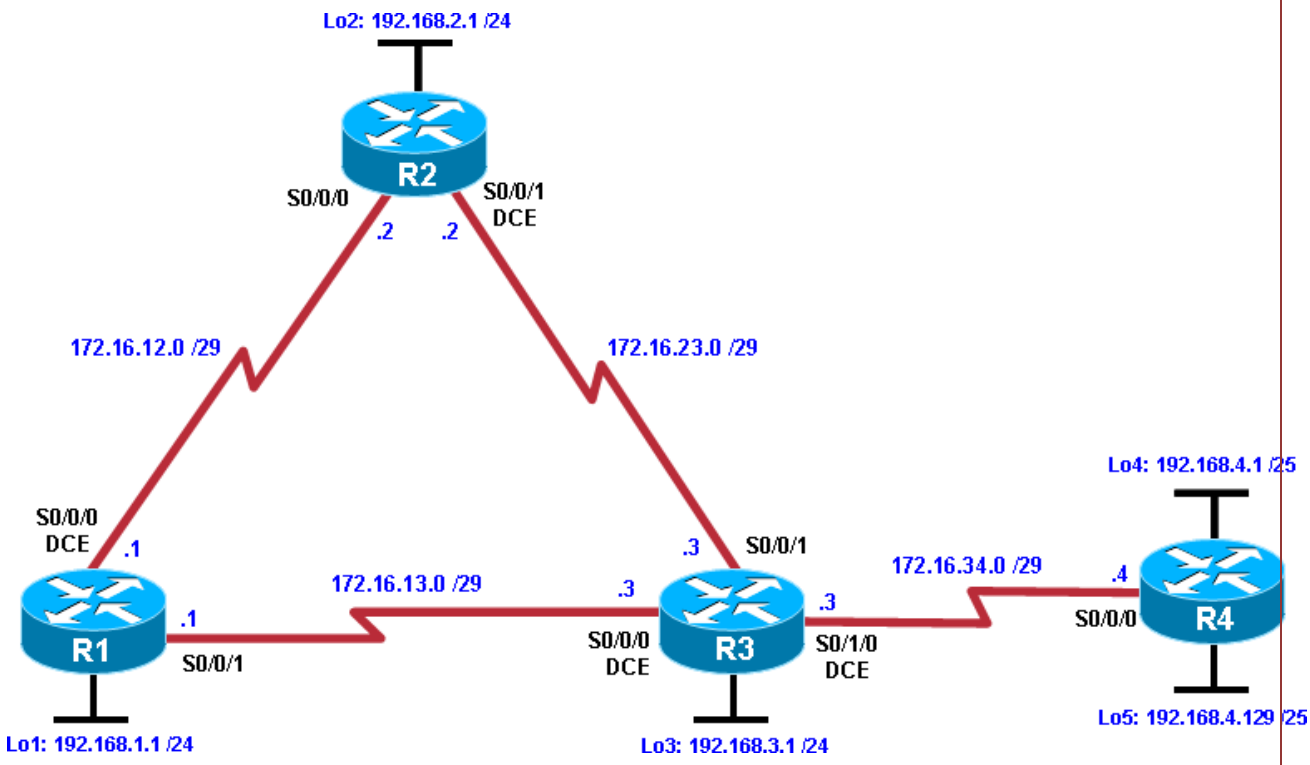
• Implementación de IPv6 en la red empresarial

DESARROLLO DE LA ACTIVIDAD

CCNPv7 ROUTE **FABIAN BERMUDEZ**

Chapter 5 Lab 5-1, Configure and Verify Path Control Using PBR

Topology



Objectives

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

Background

You want to experiment with policy-based routing (PBR) to see how it is implemented and to study how it could be of value to your organization. To this end, you have interconnected and

configured a test network with four routers. All routers are exchanging routing information using EIGRP.

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

Required Resources

- 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.

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

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

Router R1

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

```
interface Serial0/0/1
 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 Serial0/0/0
 description R2 --> R1
 ip address 172.16.12.2 255.255.255.248
 bandwidth 128
 no shutdown

interface Serial0/0/1
 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 Serial0/0/0
 description R3 --> R1
 ip address 172.16.13.3 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 description R3 --> R2
 ip address 172.16.23.3 255.255.255.248
 bandwidth 128
 no shutdown
!
interface Serial0/1/0
 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 Serial0/0/0
 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. The output from router R3 is shown here as an example.

```
R3# show ip interface brief | include up
Serial0/0/0                172.16.13.3     YES manual up
up
Serial0/0/1                172.16.23.3     YES manual up
up
Serial0/1/0                172.16.34.3     YES manual up
up
Loopback3                  192.168.3.1     YES manual up
up
R3#
R3# show protocols
Global values:
  Internet Protocol routing is enabled
```

```
Embedded-Service-Engine0/0 is administratively down, line
protocol is down

GigabitEthernet0/0 is administratively down, line protocol is
down

GigabitEthernet0/1 is administratively down, line protocol is
down

Serial0/0/0 is up, line protocol is up

  Internet address is 172.16.13.3/29

Serial0/0/1 is up, line protocol is up

  Internet address is 172.16.23.3/29

Serial0/1/0 is up, line protocol is up

  Internet address is 172.16.34.3/29

Serial0/1/1 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

Se0/0/0                        up             up       R3 -->
R1

Se0/0/1                        up             up       R3 -->
R2

Se0/1/0                        up             up       R3 -->
R4

Lo3                            up             up       R3 LAN

R3#
```

Step 3: Configure basic EIGRP.

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

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

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

Router R1

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

Router R2

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

Router R3

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

Router R4

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

You should see EIGRP neighbor relationship messages being generated.

Step 4: Verify EIGRP connectivity.

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

```
R1# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
```

```
H   Address                 Interface         Hold Uptime
SRTTRTOQ    Seq
                                              (sec)
(ms)        Cnt Num
1   172.16.13.3             Se0/0/1             10 00:01:55
27  2340  0  9
0   172.16.12.2             Se0/0/0             13 00:02:07
8  1170  0  11
R1#
```

```
R2# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface         Hold Uptime
SRTTRTOQ    Seq
                                              (sec)
(ms)        Cnt Num
1   172.16.23.3             Se0/0/1             12 00:02:15
12  1170  0  10
0   172.16.12.1             Se0/0/0             11 00:02:27
9  1170  0  13
R2#
```

```
R3# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address                 Interface         Hold Uptime
SRTTRTOQ    Seq
                                              (sec)
(ms)        Cnt Num
2   172.16.34.4             Se0/1/0             12 00:02:14
44  2340  0  3
1   172.16.23.2             Se0/0/1             11 00:02:23
10  1170  0  10
0   172.16.13.1             Se0/0/0             10 00:02:23
1031  5000  0  12
```

```
R3#

R4# show ip eigrp neighbors

EIGRP-IPv4 Neighbors for AS(1)

H   Address                 Interface         Hold Uptime
SRTTRTOQ    Seq

                                              (sec)
(ms)       Cnt Num

0   172.16.34.3             Se0/0/0             10 00:02:22
37  2340  0  11

R4#
```

Did you receive the output you expected?

Yes

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


      172.16.0.0/16 is variably subnetted, 6 subnets, 2 masks

C        172.16.12.0/29 is directly connected, Serial0/0/0

L        172.16.12.1/32 is directly connected, Serial0/0/0

C        172.16.13.0/29 is directly connected, Serial0/0/1

L        172.16.13.1/32 is directly connected, Serial0/0/1

D 172.16.23.0/29 [90/21024000] via 172.16.12.2,
00:07:22, Serial0/0/0

D 172.16.34.0/29 [90/41024000] via 172.16.13.3,
00:07:22, Serial0/0/1

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

C        192.168.1.0/24 is directly connected, Loopback1

L        192.168.1.1/32 is directly connected, Loopback1

D     192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:07:22,
Serial0/0/0

D     192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:07:22,
Serial0/0/0

      192.168.4.0/25 is subnetted, 2 subnets

D        192.168.4.0 [90/41152000] via 172.16.13.3, 00:07:14,
Serial0/0/1

D        192.168.4.128 [90/41152000] via 172.16.13.3,
00:07:14, Serial0/0/1

R1#
```

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

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

```
R4# traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.

Tracing the route to 192.168.1.1

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

  1 172.16.34.3 12 msec 12 msec 16 msec

  2 172.16.23.2 20 msec 20 msec 20 msec

  3 172.16.12.1 24 msec *  24 msec

R4#

R4# traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.

Tracing the route to 192.168.1.1

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

  1 172.16.34.3 12 msec 16 msec 12 msec

  2 172.16.23.2 28 msec 20 msec 16 msec

  3 172.16.12.1 24 msec *  24 msec

R4#
```

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

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

Debido a que las interfaces serie entre los routers R1 y R3 se han configurado con un ancho de banda inferior de 64 Kb / s, dándole una métrica más alta. Todas las demás interfaces serie utilizan el ajuste de ancho de banda de 128 Kb / s. R3 elige enviar todos los paquetes a R2 debido a su métrica inferior.

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
```

```
      172.16.0.0/16 is variably subnetted, 7 subnets, 2 masks
D        172.16.12.0/29 [90/21024000] via 172.16.23.2,
00:10:54, Serial0/0/1
C        172.16.13.0/29 is directly connected, Serial0/0/0
L        172.16.13.3/32 is directly connected, Serial0/0/0
C        172.16.23.0/29 is directly connected, Serial0/0/1
L        172.16.23.3/32 is directly connected, Serial0/0/1
C        172.16.34.0/29 is directly connected, Serial0/1/0
L 172.16.34.3/32 is directly connected, Serial0/1/0
D 192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:10:54,
Serial0/0/1
D 192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:10:54,
Serial0/0/1
      192.168.3.0/24 is variably subnetted, 2 subnets, 2
masks
C        192.168.3.0/24 is directly connected, Loopback3
L        192.168.3.1/32 is directly connected, Loopback3
      192.168.4.0/25 is subnetted, 2 subnets
D        192.168.4.0 [90/40640000] via 172.16.34.4, 00:10:47,
Serial0/1/0
D        192.168.4.128 [90/40640000] via 172.16.34.4,
00:10:47, Serial0/1/0
R3#
```

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

```
R3# show interfaces serial0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Description: R3 --> R1
  Internet address is 172.16.13.3/29
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

```
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:01, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total
output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     399 packets input, 29561 bytes, 0 no buffer
     Received 186 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0
abort
     393 packets output, 29567 bytes, 0 underruns
     0 output errors, 0 collisions, 3 interface resets
     0 unknown protocol drops
     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 serial0/0/0 | include BW
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
R3# show interfaces serial0/0/1 | include BW
  MTU 1500 bytes, BW 128 Kbit/sec, DLY 20000 usec,
R3#
```

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

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

```
R3# show ip eigrp topology 192.168.1.0

EIGRP-IPv4 Topology Entry for AS(1)/ID(192.168.3.1) for
192.168.1.0/24

  State is Passive, Query origin flag is 1, 1 Successor(s),
FD is 21152000

  Descriptor Blocks:

  172.16.23.2 (Serial0/0/1), from 172.16.23.2, Send flag is
0x0

      Composite metric is (21152000/20640000), route is
Internal

      Vector                metric:

        Minimum bandwidth is 128 Kbit

        Total delay is 45000 microseconds

        Reliability is 255/255

        Load is 1/255

        Minimum MTU is 1500

        Hop count is 2

        Originating router is 192.168.1.1

  172.16.13.1 (Serial0/0/0), 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

        Originating router is 192.168.1.1

R3#
```

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

Step 6: Configure PBR to provide path control.

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

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

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

The steps required to implement path control include the following:

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

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

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

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

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

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

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

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

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
R3#
```

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

Step 7: Test the policy.

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

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

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

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

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

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

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

```
R4# traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

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

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

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

```
R3#
Jan 10 10:49:48.411: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, policy rejected -- normal forwarding
Jan 10 10:49:48.427: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, policy rejected -- normal forwarding
Jan 10 10:49:48.439: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, policy rejected -- normal forwarding
Jan 10 10:49:48.451: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy rejected(no match) - normal
forwarding
```

```
Jan 10 10:49:48.471: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy rejected(no match) - normal
forwarding

Jan 10 10:49:48.491: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy rejected(no match) - normal
forwarding

Jan 10 10:49:48.511: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy rejected(no match) - normal
forwarding

Jan 10 10:49:48.539: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy rejected(no match) - normal
forwarding

Jan 10 10:49:51.539: IP: s=192.168.4.1 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy rejected(no match) - normal
forwarding

R3#
```

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

No toma la ruta especificada PBR porque LAN A no cumple con los criterios especificados en la lista de acceso PBR-ACL.

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

```
R4# traceroute 192.168.1.1 source 192.168.4.129


Type escape sequence to abort.

Tracing the route to 192.168.1.1


  1 172.16.34.3 12 msec 12 msec 16 msec

  2 172.16.13.1 28 msec 28 msec *
```

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

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

```
R3#

R3#
```

```
Jan 10 10:50:04.283: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, policy match

Jan 10 10:50:04.283: IP: route map R3-to-R1, item 10, permit

Jan 10 10:50:04.283: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1 (Serial0/0/0), len 28, policy routed

Jan 10 10:50:04.283: IP: Serial0/1/0 to Serial0/0/0
172.16.13.1

Jan 10 10:50:04.295: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, policy match

Jan 10 10:50:04.295: IP: route map R3-to-R1, item 10, permit

Jan 10 10:50:04.295: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1 (Serial0/0/0), len 28, policy routed

Jan 10 10:50:04.295: IP: Serial0/1/0 to Serial0/0/0
172.16.13.1

Jan 10 10:50:04.311: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, policy match

Jan 10 10:50:04.311: IP: route map R3-to-R1, item 10, permit

Jan 10 10:50:04.311: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1 (Serial0/0/0), len 28, policy routed

Jan 10 10:50:04.311: IP: Serial0/1/0 to Serial0/0/0
172.16.13.1

Jan 10 10:50:04.323: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy match

Jan 10 10:50:04.323: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, PBR Counted

Jan 10 10:50:04.323: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, g=172.16.13.1, len 28, FIB policy routed

Jan 10 10:50:04.351: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy match

Jan 10 10:50:04.351: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, PBR Counted

Jan 10 10:50:04.351: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, g=172.16.13.1, len 28, FIB policy routed

Jan 10 10:50:07.347: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, FIB policy match
```

```
Jan 10 10:50:07.347: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, len 28, PBR Counted

Jan 10 10:50:07.347: IP: s=192.168.4.129 (Serial0/1/0),
d=192.168.1.1, g=172.16.13.1, len 28, FIB policy routed

R3#
```

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

```
R3# show route-map

route-map R3-to-R1, permit, sequence 10

  Match clauses:

    ip address (access-lists): PBR-ACL

  Set clauses:

    ip next-hop 172.16.13.1

Nexthop tracking current: 0.0.0.0

172.16.13.1, fib_nh:0,oce:0,status:0


  Policy routing matches: 12 packets, 384 bytes

R3#
```

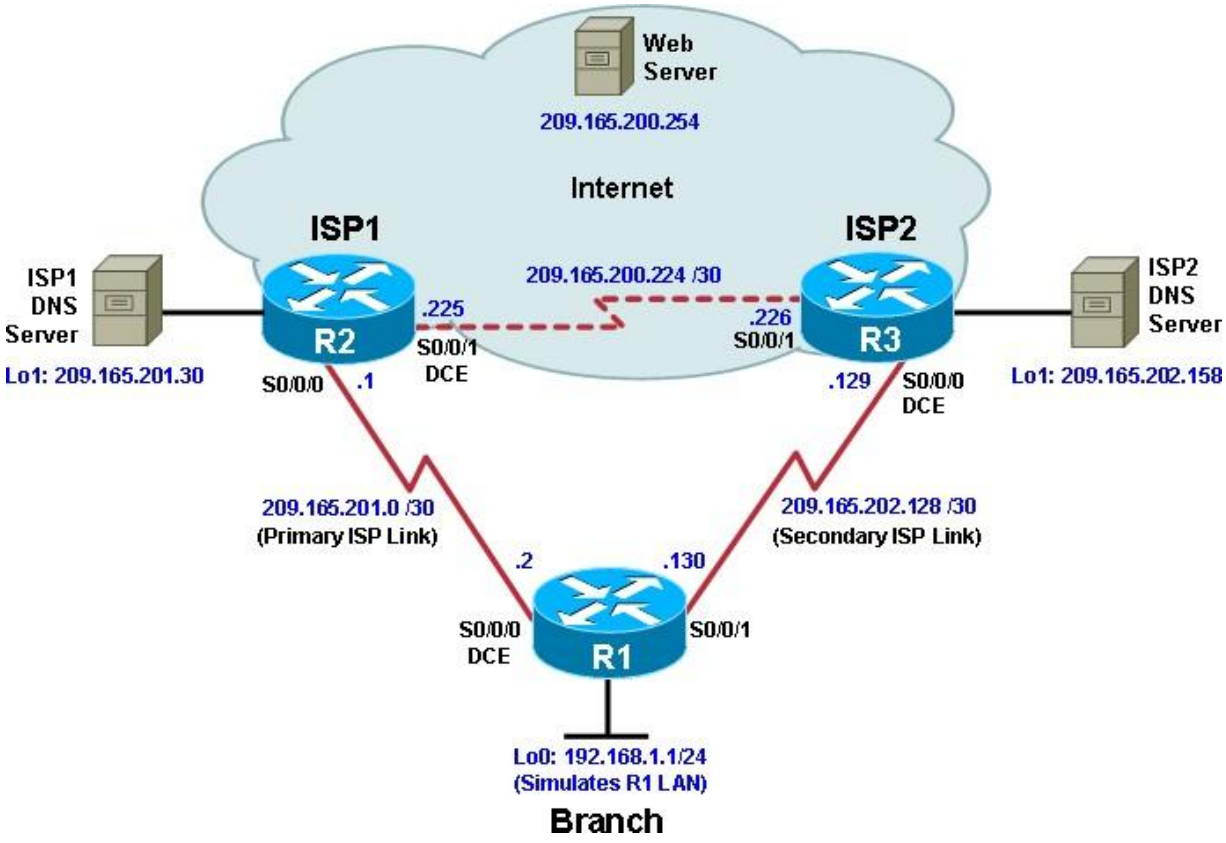
Note: There are now matches to the policy because packets matching the ACL have passed through R3 S0/1/0.

CCNPv7 ROUTE

PABLO FELIPE REYES

Chapter 5 Lab 5-2, Configure IP SLA Tracking and Path Control

Topology



Objectives

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

Background

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

At times, a link to an ISP could be operational, yet users cannot connect to any other outside Internet resources. The problem might be with the ISP or downstream from them. Although policy-based routing (PBR) can be implemented to alter path control, you will implement the

Cisco IOS SLA feature to monitor this behavior and intervene by injecting another default route to a backup ISP.

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

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

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

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

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

Router R1

```
hostname R1


interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0


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

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

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255

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

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

Router ISP2 (R3)

```
hostname ISP2

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255

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

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

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

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

All three interfaces should be active. Troubleshoot if necessary.

Step 2: Configure static routing.

The current routing policy in the topology is as follows:

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

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

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

Router R1

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

R1(config)#
```

Router ISP1 (R2)

```
ISP1(config)# router eigrp 1

ISP1(config-router)# network 209.165.200.224 0.0.0.3

ISP1(config-router)# network 209.165.201.0 0.0.0.31

ISP1(config-router)# no auto-summary

ISP1(config-router)# exit

ISP1(config)#

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

ISP1(config)#
```

Router ISP2 (R3)

```
ISP2(config)# router eigrp 1

ISP2(config-router)# network 209.165.200.224 0.0.0.3

ISP2(config-router)# network 209.165.202.128 0.0.0.31

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

```
ISP2(config-router)# exit

ISP2(config)#

ISP2(config)# ip route 192.168.1.0 255.255.255.0
209.165.202.130

ISP2(config)#
```

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

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

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

```
foreach address {

209.165.200.254

209.165.201.30

209.165.202.158

} {

ping $address source 192.168.1.1

}
```

All pings should be successful. Troubleshoot if necessary.

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

```
foreach address {

209.165.200.254

209.165.201.30

209.165.202.158

} {

trace $address source 192.168.1.1
```

}

Through which ISP is traffic flowing?

_____

_____

Step 3: Configure IP SLA probes.

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

In this scenario, you will configure ICMP echo probes.

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

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

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

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

```
R1# show ip sla configuration 11
IP SLAs Infrastructure Engine-III
Entry number: 11
Owner:
Tag:
Operation    timeout      (milliseconds):      5000
Type  of  operation  to  perform:  icmp-echo
Target address/Source address: 209.165.201.30/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
```

```
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if
randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds):  Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None



R1#
```

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

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

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

```
IPSLAs Latest Operation Statistics

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

R1#
```

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

n. 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#
```

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

```
R1# show ip sla configuration 22
IP SLAs Infrastructure Engine-III
Entry number: 22
Owner:
```

```
Tag:
Operation timeout (milliseconds): 5000
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.202.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if
randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds):  Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
```

```
R1#


R1# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if
randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds):  Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT
is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
```

```
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:

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

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


R1#
```

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

Step 4: Configure tracking options.

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

p. 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
```

q. Verify the routing table.

```
R1# show ip route | begin Gateway

Gateway of last resort is 209.165.201.1 to network 0.0.0.0


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

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

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

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

R1(config-track)#
```

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

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

```
R1(config-track)# exit

R1(config)#
```

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

```
R1# debug ip routing

IP routing debugging is on

R1#
```

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

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

R1(config)#

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0)   :
    via 209.165.201.1    0 1048578


Jan 10 10:45:39.119: RT: closer admin distance for 0.0.0.0,
flushing 1 routes

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

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0)   :
    via 209.165.201.1    0 1048578


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

Jan 10 10:45:39.119: RT: updating static 0.0.0.0/0 (0x0)   :
    via 209.165.201.1    0 1048578


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

R1(config)#
```

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

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

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

w. Verify the routing table again.

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

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

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

Step 5: Verify IP SLA operation.

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

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

x. On ISP1, disable the loopback interface 1.

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

ISP1(config-if)# shutdown

ISP1(config-if)#

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

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

ISP1(config-if)#
```

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

```
R1#

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

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

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

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

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

    via 209.165.202.129    0 1048578


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

```
Jan 10 10:53:59.551: RT: default path is now 0.0.0.0 via
209.165.202.129
Jan 10 10:53:59.551: RT: updating static 0.0.0.0/0 (0x0)   :
    via 209.165.201.1    0 1048578

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

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

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

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

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

z. On R1, verify the routing table.

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

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

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

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

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

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

R1#
```

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

aa. Verify the IP SLA statistics.

```
R1# show ip sla statistics

IPSLAs Latest Operation Statistics


IPSLA operation id: 11

     Latest RTT: NoConnection/Busy/Timeout

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

Latest operation return code: Timeout

Number of successes:  173

Number of failures: 45

Operation time to live: Forever




IPSLA operation id: 22

     Latest RTT: 8 milliseconds

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

Latest operation return code: OK

Number of successes: 218

Number of failures: 0

Operation time to live: Forever
```

```
R1#
```

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

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

```
R1# trace 209.165.200.254 source 192.168.1.1

Type escape sequence to abort.

Tracing the route to 209.165.200.254

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

  1 209.165.202.129 4 msec *  *

R1#
```

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

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

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

Jan 10 11:05:45.847: %LINK-3-UPDOWN: Interface Loopback1,
changed state to up

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

ISP1(config-if)#
```

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

```
R1#

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

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

    via 209.165.201.1    0 1048578


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

```
Jan 10 11:06:20.551: RT: add 0.0.0.0/0 via 209.165.201.1,
static metric [2/0]
Jan 10 11:06:20.551: RT: updating static 0.0.0.0/0 (0x0)   :
    via 209.165.202.129    0 1048578

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

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

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

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

dd. Again examine the IP SLA statistics.

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

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

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


R1#
```

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

ee. 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.

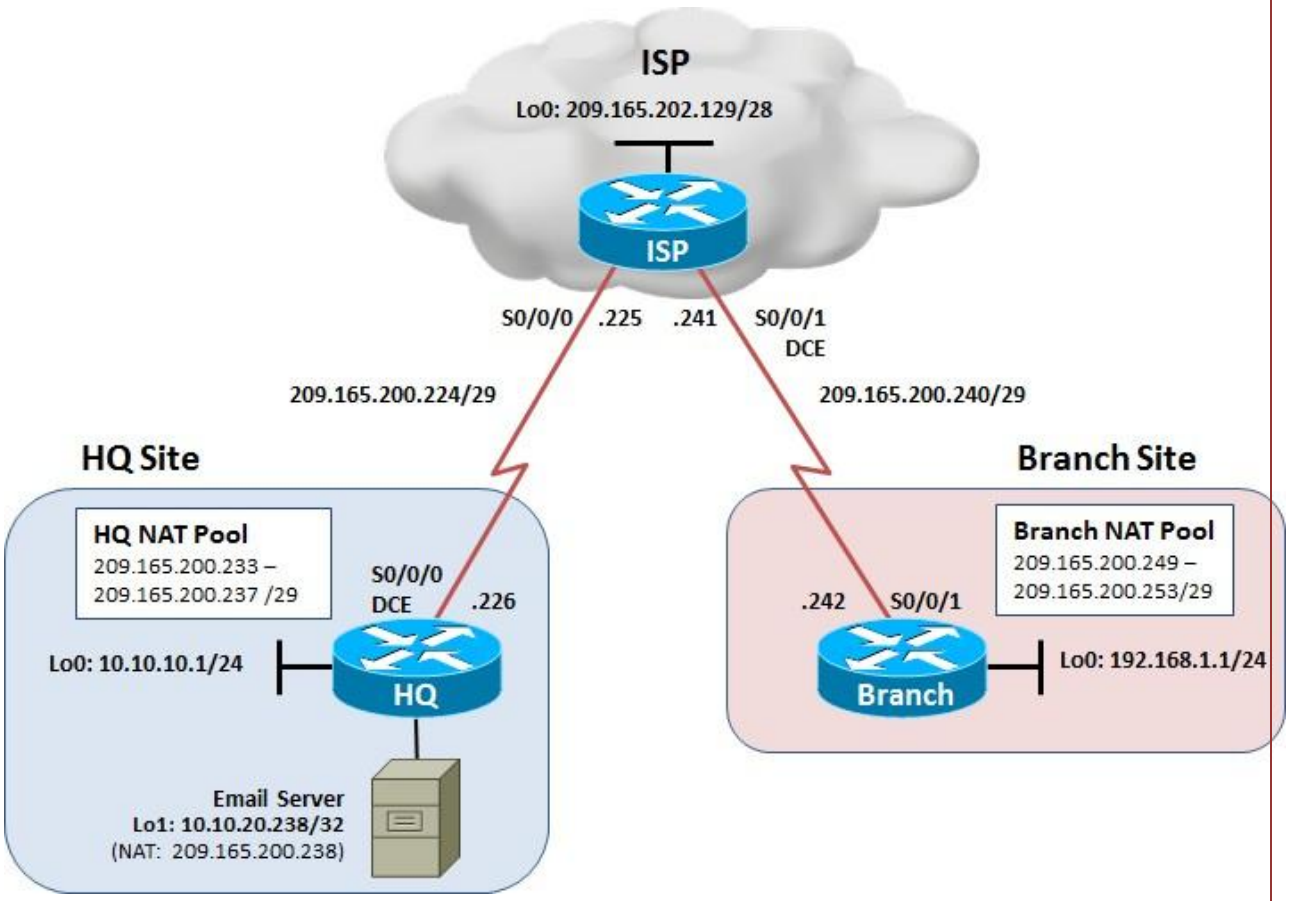
There are many possibilities available with object tracking and Cisco IOS IP SLAs. As shown in this lab, a probe can be based on reachability, changing routing operations, and path control based on the ability to reach an object. However, Cisco IOS IP SLAs also allow paths to be changed based on network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

Chapter 6 Lab 6-1, Configure NAT Services ALVARO ELIU MELO

Topology



Objectives

- Configure dynamic NAT and static NAT on the HQ router.
- Configure dynamic NAT on the Branch router.
- Verify the configuration and operation using **show** commands.

Background

The HQ and Branch sites must be configured to support NAT. Specifically, the HQ and Branch routers will be configured to provide inside LAN users with outside public addresses using NAT. The HQ router will also provide static NAT to access the Email server from the outside network.

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

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

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

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

HQ (R1)

```
hostname HQ
!
interface Loopback0
 description Headquarters LAN
 ip address 10.10.10.1 255.255.255.0
exit
!
interface Loopback1
 description Simulates the Email Server
 ip address 10.10.20.238 255.255.255.255
exit
!
interface Serial0/0/0
 description Connection to ISP
```

```
 ip address 209.165.200.226 255.255.255.248
 clock rate 128000
 no shut
exit
!
end
```

ISP (R2)

```
hostname ISP
!
interface Loopback0
 description Simulating the Internet
 ip address 209.165.202.129 255.255.255.240
exit
!
interface Serial0/0/0
 description Connection to HQ
 ip address 209.165.200.225 255.255.255.248
 no shut
exit
!
interface  Serial0/0/1
 description Connection to Branch
 ip address 209.165.200.241 255.255.255.248
 clock rate 128000
 no shut
exit
!
ip route 209.165.200.232 255.255.255.248 Serial0/0/0
ip route 209.165.200.248 255.255.255.248 Serial0/0/1
!
```

```
end
```

Branch (R3)

```
hostname Branch
!
interface Loopback0
 description Branch LAN
 ip address 192.168.1.1 255.255.255.0
exit
!
interface Serial0/0/1
 description Connection to ISP
 ip address 209.165.200.242 255.255.255.248
 no shut
exit
!
end
```

a. Verify your configuration by using the **show ip interface brief** and the **show interfaces description** command. The output from the Branch router is shown here as an example.

```
Branch# show ip interface brief | include up
Serial0/0/1                209.165.200.242 YES manual up
up
Loopback0                  192.168.1.1     YES manual up
up
Branch#
Branch# show interfaces description | include up
Se0/0/1                          up             up       Connection
to ISP
Lo0                              up             up       Branch LAN
Branch#
```

b. From the Branch router, run the following Tcl script to verify connectivity.

```
foreach address {
209.165.200.241
209.165.202.129
```

```
209.165.200.226
} { ping $address }



Branch# tclsh
Branch(tcl)# foreach address {
+>209.165.200.241
+>209.165.202.129
+>209.165.200.226
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, timeout is
2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/13/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.129, timeout is
2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.226, timeout is
2 seconds:
.....
Success rate is 0 percent (0/5)
Branch(tcl)#
```

Why do the pings to the ISPs loopback and HQ router address fail?

El ping falla porque los enrutadores Branch y HQ requieren una ruta predeterminada al ISP.

Step 2: Configure default static routes on Branch and HQ.

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

```
HQ(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.225
```

b. On the Branch router, configure a default static route to ISP.

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

```
Branch(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.241
```

c. From the Branch router, run the following Tcl script to verify connectivity.

```
foreach address {
209.165.200.241
209.165.202.129
209.165.200.226
+>} { ping $address}


Branch# tclsh
Branch(tcl)# foreach address {
+>209.165.200.241
+>209.165.202.129
+>209.165.200.226
+>} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, timeout is
2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/13/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.129, timeout is
2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/13/16 ms
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 209.165.200.226, timeout is
2 seconds:

!!!!!

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

Branch(tcl)#
```

Are the pings now successful?

Si

Connectivity from the Branch router to external addresses has been established. But could a Branch LAN user successfully reach those external addresses? To verify, initiate pings sourced from the Branch LAN interface to the ISP interface, the ISPs loopback interface, and the HQ Internet interface. Run the following Tcl script on the Branch router to verify connectivity.

```
foreach address {

209.165.200.241

209.165.202.129

209.165.200.226

} { ping $address source 192.168.1.1}
```

```
Branch# tclsh

Branch(tcl)# foreach address {

+>209.165.200.241

+>209.165.202.129

+>209.165.200.226

+>} { ping $address source 192.168.1.1}

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 209.165.200.241, timeout is
2 seconds:

Packet sent with a source address of 192.168.1.1

.....

Success rate is 0 percent (0/5)
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 209.165.202.129, timeout is
2 seconds:

Packet sent with a source address of 192.168.1.1

.....

Success rate is 0 percent (0/5)

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 209.165.200.226, timeout is
2 seconds:

Packet sent with a source address of 192.168.1.1

.....

Success rate is 0 percent (0/5)

Branch(tcl)#
```

Note: You can also specify the router interface designator (for example, S0/0/0, Fa0/0, or Lo1) as the source for the extended ping, as follows:

```
Branch# ping 209.165.200.226 source Lo1
```

Why are the pings unsuccessful?

Los pings fallan porque la dirección IP de origen 192.168.1.1 es una dirección privada interna, y el ISP no es consciente de esta dirección.

El ISP no puede redirigir a la dirección privada interna de la LAN de sucursal.

Step 3: Configure NAT on the HQ router.

The HQ and Branch internal LANs will be translated to global public IP addresses using NAT when exiting the corporate network.

The ISP has allocated the **209.165.200.233 – 209.165.200.238 (209.165.200.232/29)** pool of public addresses to the HQ site.

The HQ site also has an email server that must be accessible to mobile users and Branch office users. Therefore, static NAT must also be configured to use a public address to reach the email server.

a. On the HQ router, create an extended NAT ACL that matches the 10.10.10.0/24 LAN.

```
HQ(config)# ip access-list extended HQ-NAT-ACL

HQ(config-ext-nacl)# remark Permit Local LAN to use NAT

HQ(config-ext-nacl)# permit ip 10.10.10.0 0.0.0.255 any
```

```
HQ(config-ext-nacl)# exit

HQ(config)#
```

b. The NAT pool must identify addresses 209.165.200.232 /29.

```
HQ(config)# ip nat pool HQ-NAT-POOL 209.165.200.233
209.165.200.237 prefix-length 29

HQ(config)#
```

c. The NAT pool and the ACL must be bound together.

```
HQ(config)# ip nat inside source list HQ-NAT-ACL pool HQ-NAT-
POOL

HQ(config)#
```

d. The email server with private IP address 10.10.20.238 will be statically assigned the last public IP address from the NAT pool, 209.165.200.238. Interface loopback 0 on HQ simulates this server.

```
HQ(config)# ip nat inside source static 10.10.20.238
209.165.200.238

HQ(config)#
```

e. The LAN interface must be identified as an inside NAT interface, and the Internet interface must be identified as an outside NAT interface.

```
HQ(config)# interface Loopback 0

HQ(config-if)# ip nat inside

HQ(config-if)# exit

HQ(config)#

HQ(config)# interface Loopback 1

HQ(config-if)# ip nat inside

HQ(config-if)# exit

HQ(config)#

HQ(config)# interface Serial0/0/0

HQ(config-if)# ip nat outside

HQ(config-if)# exit

HQ(config)#
```

Step 4: Configure NAT on the Branch router.

The ISP has allocated the **209.165.200.249 – 209.165.200.254 (209.165.200.248/29)** pool of public addresses to the Branch site.

a. On the Branch router, create a standard NAT ACL that identifies the 192.168.1.0/24 LAN.

```
Branch(config)# ip access-list extended BRANCH-NAT-ACL
Branch(config-ext-nacl)# remark Permit Local LAN to use NAT
Branch(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 any
Branch(config-ext-nacl)# exit
Branch(config)#
```

b. The NAT pool must identify addresses 209.165.200.232 /29.

```
Branch(config)# ip nat pool BRANCH-NAT-POOL 209.165.200.249
209.165.200.254 prefix-length 29
Branch(config)#
```

c. The NAT pool and the ACL must be bound together.

```
Branch(config)# ip nat inside source list BRANCH-NAT-ACL pool
BRANCH-NAT-POOL
Branch(config)#
```

d. The LAN interface must be identified as an inside NAT interface, and the Internet interface must be identified as an outside NAT interface.

```
Branch(config)# interface Loopback 0
Branch(config-if)# ip nat inside
Branch(config-if)# exit
Branch(config)#
Branch(config)# interface Serial0/0/1
Branch(config-if)# ip nat outside
Branch(config-if)# exit
Branch(config)#
```

Step 5: Verify NAT Configuration.

a. Verify the NAT configuration by using the **show ip nat statistics** and **show ip nat translations** commands.

```
Branch# show ip nat statistics
```

```
Total active translations: 0 (0 static, 0 dynamic; 0
extended)

Peak translations: 0

Outside interfaces:

  Serial0/0/1

Inside interfaces:

  Loopback0

Hits: 0  Misses: 0

CEF Translated packets: 0, CEF Punted packets: 0

Expired translations: 0

Dynamic mappings:

-- Inside Source

[Id: 1] access-list BRANCH-NAT-ACL pool BRANCH-NAT-POOL
refcount 0

 pool BRANCH-NAT-POOL: netmask 255.255.255.248

      start 209.165.200.249 end 209.165.200.254

      type generic, total addresses 6, allocated 0 (0%),
misses 0


Total doors: 0

Appl doors: 0

Normal doors: 0

Queued Packets: 0

Branch#
```

As shown above, the pool has been configured and the interfaces assigned. The output of the **show ip nat translations** command confirms that there are currently no active NAT translations:

```
Branch# show ip nat translations

Branch#
```

b. Initiate NAT traffic by pinging from the Branch LAN to the ISP interface, ISP's loopback, the HQ Internet interface, and this time also include the HQ public email server address. Run the following Tcl script on the Branch router to verify connectivity.

```
foreach address {
209.165.200.241
209.165.202.129
209.165.200.226
209.165.200.238
} { ping $address source 192.168.1.1}
```

```
Branch# tclsh
Branch(tcl)# foreach address {
+>209.165.200.241
+>209.165.202.129
+>209.165.200.226
+>209.165.200.238
+>} { ping $address source 192.168.1.1}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, timeout is
2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/13/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.129, timeout is
2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/13/16 ms
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 209.165.200.226, 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 =
28/28/28 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 209.165.200.238, 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 =
28/28/28 ms

Branch(tcl)#
```

All pings should be successful. Troubleshoot if necessary.

c. Verify that NAT is occurring by using the **show ip nat statistics** and **show ip nat translations** commands.

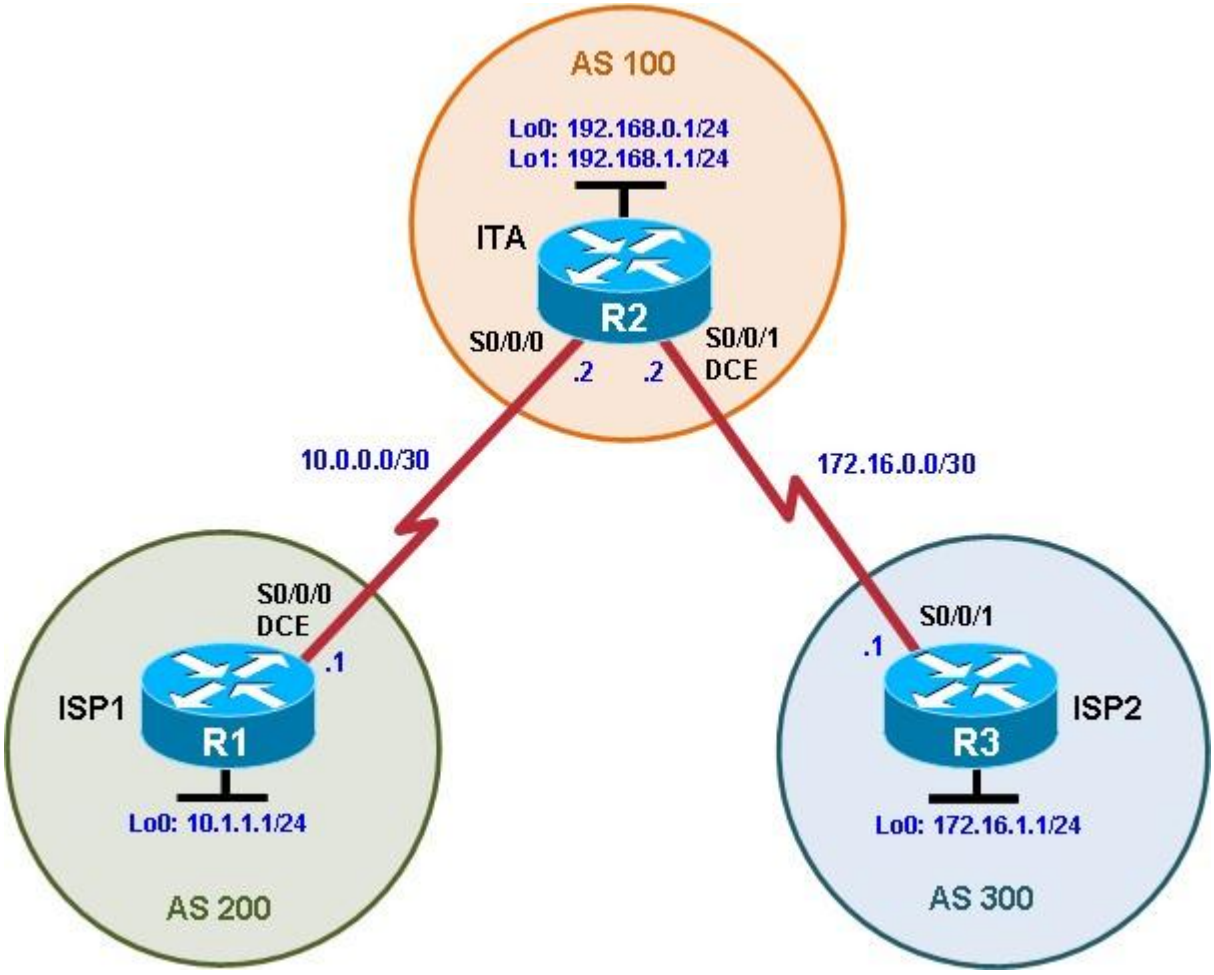
```
Branch# show ip nat statistics

Total active translations: 5 (0 static, 5 dynamic; 4
extended)

Peak translations: 5, occurred 00:00:13 ago

Outside interfaces:

  Serial0/0/1

Inside interfaces:

  Loopback0

Hits: 40  Misses: 0

CEF Translated packets: 20, CEF Punted packets: 0

Expired translations: 0

Dynamic mappings:

-- Inside Source
```

```
[Id: 1] access-list BRANCH-NAT-ACL pool BRANCH-NAT-POOL
refcount 5

 pool BRANCH-NAT-POOL: netmask 255.255.255.248

     start 209.165.200.249 end 209.165.200.254

     type generic, total addresses 6, allocated 1 (16%),
misses 0


Total doors: 0

Appl doors: 0

Normal doors: 0

Queued Packets: 0

Branch#

Branch# show ip nat translations

Pro Inside global      Inside local       Outside local
Outside global

icmp 209.165.200.249:31 192.168.1.1:31    209.165.200.241:31
209.165.200.241:31

icmp 209.165.200.249:32 192.168.1.1:32    209.165.202.129:32
209.165.202.129:32

icmp 209.165.200.249:33 192.168.1.1:33    209.165.200.226:33
209.165.200.226:33

icmp 209.165.200.249:34 192.168.1.1:34    209.165.200.238:34
209.165.200.238:34

--- 209.165.200.249    192.168.1.1        ---
---

Branch#
```

Notice that translations are occurring. The output lists the details of the NAT translations sourced by the 192.168.1.1 Branch LAN IP address, which was translated to public IP address 209.165.200.249.

Chapter 7 Lab 7-1, Configuring BGP with Default Routing

Topology ALVARO ELIU MELO



Objectives

- Configure BGP to exchange routing information with two ISPs.

Background

The International Travel Agency (ITA) relies extensively on the Internet for sales. For this reason, the ITA has decided to create a multihomed ISP connectivity solution and contracted with two ISPs for Internet connectivity with fault tolerance. Because the ITA is connecting to two different service providers, you must configure BGP, which runs between the ITA boundary router and the two ISP routers.

Note: This lab uses Cisco 1941 routers with Cisco IOS Release 15.4 with IP Base. The switches are Cisco WS-C2960-24TT-L with Fast Ethernet interfaces, therefore the router will use routing metrics associated with a 100 Mb/s interface. Depending on the router or

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

Required Resources

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

Step 0: Suggested starting configurations.

ff. 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.

gg. Using the addressing scheme in the diagram, create the loopback interfaces and apply IPv4 addresses to these and the serial interfaces on ISP1 (R1), ISP2 (R3), and ITA (R2). The ISP loopbacks simulate real networks that can be reached through the ISP. The two loopbacks for the ITA router simulate the connections between the ITA boundary router and their core routers. Set a clock rate on the DCE serial interfaces.

```
ISP1(config)# interface Lo0
ISP1(config-if)# description ISP1 Internet Network
ISP1(config-if)# ip address 10.1.1.1 255.255.255.0
ISP1(config-if)# exit
ISP1(config)# interface Serial0/0/0
ISP1(config-if)# description ISP1 -> ITA
ISP1(config-if)# ip address 10.0.0.1 255.255.255.252
ISP1(config-if)# clock rate 128000
ISP1(config-if)# no shutdown
ISP1(config-if)# end
ISP1#

ITA(config)# interface Lo0
ITA(config-if)# description Core router network link 1
ITA(config-if)# ip address 192.168.0.1 255.255.255.0
ITA(config)# exit
ITA(config-if)# interface Lo1
ITA(config-if)# description Core router network link 2
ITA(config-if)# ip address 192.168.1.1 255.255.255.0
ITA(config-if)# exit
ITA(config)# interface Serial0/0/0
ITA(config-if)# description ITA -> ISP1
ITA(config-if)# ip address 10.0.0.2 255.255.255.252
ITA(config-if)# no shutdown
ITA(config-if)# exit
```

```
ITA(config)# interface Serial0/0/1
ITA(config-if)# description ITA -> ISP2
ITA(config-if)# ip address 172.16.0.2 255.255.255.252
ITA(config-if)# clock rate 128000
ITA(config-if)# no shutdown
ITA(config-if)# end
ITA#

ISP2(config)# interface Lo0
ISP2(config-if)# description ISP2 Internet Network
ISP2(config-if)# ip address 172.16.1.1 255.255.255.0
ISP2(config)# exit
ISP2(config-if)# interface Serial0/0/1
ISP2(config-if)# description ISP2 -> ITA
ISP2(config-if)# ip address 172.16.0.1 255.255.255.252
ISP2(config-if)# no shutdown
ISP2(config-if)# end
ISP2#
```

hh. Use **ping** to test the connectivity between the directly connected routers. Note that router ISP1 cannot reach router ISP2.

Step 2: Configure BGP on the ISP routers.

On the ISP1 and ISP2 routers, configure BGP to peer with the ITA boundary router and advertise the ISP loopback networks.

```
ISP1(config)# router bgp 200
ISP1(config-router)# neighbor 10.0.0.2 remote-as 100
ISP1(config-router)# network 10.1.1.0 mask 255.255.255.0

ISP2(config)# router bgp 300
ISP2(config-router)# neighbor 172.16.0.2 remote-as 100
ISP2(config-router)# network 172.16.1.0 mask 255.255.255.0
```

Step 3: Configure BGP on the ITA boundary router.

ii. Configure the ITA router to run BGP with both Internet providers.

```
ITA(config)# router bgp 100
ITA(config-router)# neighbor 10.0.0.1 remote-as 200
ITA(config-router)# neighbor 172.16.0.1 remote-as 300
ITA(config-router)# network 192.168.0.0
ITA(config-router)# network 192.168.1.0
```

You should see BGP neighbor peering messages on the console similar to the following.

```
*Sep  8 16:00:21.587: %BGP-5-ADJCHANGE: neighbor 10.0.0.1 Up
```

jj. To verify the configuration, check the ITA routing table with the **show ip route** command.

```
ITA# 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

      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C        10.0.0.0/30 is directly connected, Serial0/0/0
L        10.0.0.2/32 is directly connected, Serial0/0/0
B        10.1.1.0/24 [20/0] via 10.0.0.1, 00:01:10
      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.2/32 is directly connected, Serial0/0/1
B        172.16.1.0/24 [20/0] via 172.16.0.1, 00:00:53
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, Loopback0
L        192.168.0.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback1
L      192.168.1.1/32 is directly connected, Loopback1
ITA#
```

ITA has BGP routes to the loopback networks at each ISP router.

kk. Run the following Tcl script on all routers to verify connectivity If these pings are not successful, troubleshoot. Use **exit** to exit the Tcl script.

Note: The WAN subnets connecting ITA (R2) to the ISPs (R1 and R3) are not advertised in BGP, so the ISPs will not be able to ping each other's serial interface address.

```
ITA# tclsh

foreach address {
10.0.0.1
10.0.0.2
10.1.1.1
172.16.0.1
172.16.0.2
172.16.1.1
192.168.0.1
192.168.1.1
} {
ping $address }
```

Step 4: Verify BGP on the routers.

ll. To verify the BGP operation on ITA, issue the **show ip bgp** command.

```
ITA# show ip bgp
BGP table version is 5, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24     10.0.0.1                 0             0 200
i
 *>  172.16.1.0/24   172.16.0.1               0             0 300
i
 *>  192.168.0.0     0.0.0.0                  0         32768 i
 *>  192.168.1.0     0.0.0.0                  0         32768 i
ITA#
```

What is the local router ID?

ID del enrutador local es 192.168.1.1

Which table version is displayed?

La versión de la tabla que se muestra es 5.

An asterisk (*) next to a route indicates that it is valid. An angle bracket (>) indicates that the route has been selected as the best route.

mm. To verify the operation of ISP1, issue the **show ip bgp** command.

```
ISP1# show ip bgp
BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24     0.0.0.0                  0         32768 i
 *>  172.16.1.0/24   10.0.0.2                                 0 100
300 i
 *>  192.168.0.0     10.0.0.2                 0             0 100
i
 *>  192.168.1.0     10.0.0.2                 0             0 100
i
ISP1#
```

Which table version is displayed and is it the same as the BGP table version for ITA?

La versión de la tabla que se muestra es 5, que es la misma que se muestra para ITA

From ISP1, what is the path to network 172.16.1.0/24?

La ruta es a través de AS 100 (ITA) y AS 300 (ISP2)

nn. On the ISP1 router, issue the **shutdown** command on Loopback0. Then on ITA, issue the **show ip bgp** command again.

```
ISP1(config)# interface loopback 0
ISP1(config-if)# shutdown
ISP1(config-if)#

ITA# show ip bgp
BGP table version is 6, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  172.16.1.0/24    172.16.0.1               0             0 300
i
 *>  192.168.0.0      0.0.0.0                  0         32768 i
 *>  192.168.1.0      0.0.0.0                  0         32768 i
ITA#
```

Which table version is displayed? Why?

La versión de la tabla mostrada es un incremento de la última, que es 6 en el ejemplo. El comando shutdown provoca una actualización de la tabla de enrutamiento, por lo que la versión debe ser una más alta que la última

What happened to the route for network 10.1.1.0/24?

Ya no está en la tabla BGP porque la interfaz Lo0 en ISP1 está inactiva.

oo. Bring ISP1 router Loopback0 back up by issuing the **no shutdown** command.

```
ISP1(config)# interface loopback 0
ISP1(config-if)# no shutdown
ISP1(config-if)#
```

pp. On ITA, issue the **show ip bgp neighbors** command. The following is a partial sample output of the command showing neighbor 172.16.0.1.

```
ITA# show ip bgp neighbors
BGP neighbor is 10.0.0.1,  remote AS 200, external link
  BGP version 4, remote router ID 10.1.1.1
  BGP state = Established, up for 00:20:47
```

```
  Last read 00:00:49, last write 00:00:41, hold time is 180,
keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                5          1
    Keepalives:            15         17
    Route Refresh:          0          0
    Total:                 21         19
  Default minimum time between advertisement runs is 30 seconds

<output omitted>
```

Based on the output of this command, what is the BGP state between this router and ISP2?

El estado de BGP está establecido.

How long has this connection been up?

La conexión ha estado en 00:15:39.

Step 5: Configure route filters.

qq. Check the ISP2 routing table using the **show ip route** command. ISP2 should have a route that belongs to ISP1, network 10.1.1.0.

```
ISP2# show ip route
<output omitted>

      10.0.0.0/24 is subnetted, 1 subnets
B        10.1.1.0 [20/0] via 172.16.0.2, 00:09:26
      172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.1/32 is directly connected, Serial0/0/1
C        172.16.1.0/24 is directly connected, Loopback0
L        172.16.1.1/32 is directly connected, Loopback0
B     192.168.0.0/24 [20/0] via 172.16.0.2, 00:28:05
B     192.168.1.0/24 [20/0] via 172.16.0.2, 00:28:05
ISP2#
```

If ITA advertises a route belonging to ISP1, ISP2 installs that route in its table. ISP2 might then attempt to route transit traffic through the ITA. This would make ITA a transit router. A traceroute to ISP1's Lo0 interface illustrates this issue.

```
ISP2# traceroute  10.1.1.1
Type escape sequence to abort.
Tracing the route to 10.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.0.2 8 msec 4 msec 8 msec
  2  *  *  *
  3  *  *  *
  4  *  *  *  <control-shift-6 to break>
ISP2#
```

The **traceroute 10.1.1.1** fails because ISP1 does not have a route to the source IPv4 address of the traceroute, 172.16.0.1. It is common in BGP networks not to advertise the links between providers in BGP. A traceroute using the source IPv4 address of ISP2' Lo0 interface is successful, showing that ITA is a transit router for this network.

```
ISP2# traceroute 10.1.1.1 source loopback0
Type escape sequence to abort.
Tracing the route to 10.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.0.2 8 msec 4 msec 8 msec
  2 10.0.0.1 12 msec *  12 msec
ISP2#
```

rr. Configure the ITA router so that it advertises only ITA networks 192.168.0.0 and 192.168.1.0 to both providers. On the ITA router, configure the following access list.

```
ITA(config)# access-list 1 permit 192.168.0.0 0.0.1.255
```

ss. Apply this access list as a route filter using the **distribute-list** keyword with the BGP **neighbor** statement.

```
ITA(config)# router bgp 100
ITA(config-router)# neighbor 10.0.0.1 distribute-list 1 out
ITA(config-router)# neighbor 172.16.0.1 distribute-list 1 out
```

tt. Check the routing table for ISP2 again. The route to 10.1.1.0, ISP1, should still be in the table.

```
ISP2# show ip route
<output omitted>

      10.0.0.0/24 is subnetted, 1 subnets
B        10.1.1.0 [20/0] via 172.16.0.2, 00:25:14
      172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.1/32 is directly connected, Serial0/0/1
C        172.16.1.0/24 is directly connected, Loopback0
L        172.16.1.1/32 is directly connected, Loopback0
B     192.168.0.0/24 [20/0] via 172.16.0.2, 00:43:53
B     192.168.1.0/24 [20/0] via 172.16.0.2, 00:43:53
ISP2#
```

uu. Return to ITA and issue the **clear ip bgp *** command. Wait until the routers reach the established state, which might take several seconds, and then recheck the ISP2 routing table. The route to ISP1, network 10.1.1.0, should no longer be in the routing table for ISP2, and the route to ISP2, network 172.16.1.0, should not be in the routing table for ISP1.

```
ITA# clear ip bgp *
ITA#
*Sep  8 16:47:25.179: %BGP-5-ADJCHANGE: neighbor 10.0.0.1 Down User
reset
*Sep  8 16:47:25.179: %BGP_SESSION-5-ADJCHANGE: neighbor 10.0.0.1
IPv4 Unicast topology base removed from session  User reset
*Sep  8 16:47:25.179: %BGP-5-ADJCHANGE: neighbor 172.16.0.1 Down
User reset
*Sep  8 16:47:25.179: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.0.1
IPv4 Unicast topology base removed from session  User reset
*Sep  8 16:47:25.815: %BGP-5-ADJCHANGE: neighbor 10.0.0.1 Up
*Sep  8 16:47:25.819: %BGP-5-ADJCHANGE
ITA#: neighbor 172.16.0.1 Up
ITA#
```

Note: The **clear ip bgp *** command is disruptive because it completely resets all BGP adjacencies. This is acceptable in a lab environment but could be problematic in a production network. Instead, if only a change of inbound/outbound routing policies is to be performed, it is sufficient to issue the **clear ip bgp * in** or **clear ip bgp * out** commands**.** These commands perform only a new BGP database synchronization without the disruptive effects of a complete BGP adjacency reset. All current Cisco IOS versions support the route refresh capability that replaces the inbound soft reconfiguration feature that previously had to be configured on a per-neighbor basis.

```
ISP2# show ip route
<output omitted>


      172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.1/32 is directly connected, Serial0/0/1
C        172.16.1.0/24 is directly connected, Loopback0
L        172.16.1.1/32 is directly connected, Loopback0
B     192.168.0.0/24 [20/0] via 172.16.0.2, 00:00:06
B     192.168.1.0/24 [20/0] via 172.16.0.2, 00:00:06
ISP2#


ISP1# show ip route
<output omitted>

      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
C        10.0.0.0/30 is directly connected, Serial0/0/0
L        10.0.0.1/32 is directly connected, Serial0/0/0
C        10.1.1.0/24 is directly connected, Loopback0
L        10.1.1.1/32 is directly connected, Loopback0
B      192.168.0.0/24 [20/0] via 10.0.0.2, 00:00:42
B      192.168.1.0/24 [20/0] via 10.0.0.2, 00:00:42
```

```
ISP1#
```

Step 6: Configure primary and backup routes using floating static routes.

With bidirectional communication established with each ISP via BGP, configure the primary and backup routes. This can be done with floating static routes or BGP.

vv. Issue the **show ip route** command on the ITA router.

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

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C        10.0.0.0/30 is directly connected, Serial0/0/0
L        10.0.0.2/32 is directly connected, Serial0/0/0
B        10.1.1.0/24 [20/0] via 10.0.0.1, 00:03:51
      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.2/32 is directly connected, Serial0/0/1
B        172.16.1.0/24 [20/0] via 172.16.0.1, 00:03:51
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, Loopback0
L        192.168.0.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback1
L      192.168.1.1/32 is directly connected, Loopback1
ITA#
```

Notice that there is no gateway of last resort defined. This is a problem because ITA is the border router for the corporate network.

ww. Configure static routes to reflect the policy that ISP1 is the primary provider and that ISP2 acts as the backup by specifying a lower distance metric for the route to ISP1 (210) as compared to the backup route to ISP2 (distance metric 220).

```
ITA(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1 210
ITA(config)# ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

xx. Verify that a default route is defined using the **show ip route** command.

```
ITA# show ip route
<output omitted>
Gateway of last resort is 10.0.0.1 to network 0.0.0.0

S*    0.0.0.0/0 [210/0] via 10.0.0.1
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C        10.0.0.0/30 is directly connected, Serial0/0/0
L        10.0.0.2/32 is directly connected, Serial0/0/0
B        10.1.1.0/24 [20/0] via 10.0.0.1, 00:05:38
      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.2/32 is directly connected, Serial0/0/1
B        172.16.1.0/24 [20/0] via 172.16.0.1, 00:05:38
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
```

```
C        192.168.0.0/24 is directly connected, Loopback0
L        192.168.0.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback1
L        192.168.1.1/32 is directly connected, Loopback1
ITA#
```

yy. Test this default route by creating an unadvertised loopback on the router for ISP1.

```
ISP1# config t
ISP1(config)# interface loopback 100
ISP1(config-if)# ip address 192.168.100.1 255.255.255.0
```

zz. Issue the **show ip route** command to ensure that the newly added 192.168.100.0 /24 network does not appear in the routing table.

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

Gateway of last resort is 10.0.0.1 to network 0.0.0.0

S*    0.0.0.0/0 [210/0] via 10.0.0.1
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C        10.0.0.0/30 is directly connected, Serial0/0/0
L        10.0.0.2/32 is directly connected, Serial0/0/0
B        10.1.1.0/24 [20/0] via 10.0.0.1, 00:07:08
      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.2/32 is directly connected, Serial0/0/1
B        172.16.1.0/24 [20/0] via 172.16.0.1, 00:07:08
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, Loopback0
L        192.168.0.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback1
L      192.168.1.1/32 is directly connected, Loopback1
ITA#
```

aaa. In extended ping mode, ping the ISP1 loopback 1 interface 192.168.100.1 with the source originating from the ITA loopback 1 interface 192.168.1.1.

```
ITA# 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: 192.168.1.1
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 192.168.100.1, timeout is 2
seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms
ITA#
```

Note: You can bypass extended ping prompted mode and ping while specifying a source address using one of these abbreviated commands:

```
ITA# ping 192.168.100.1 source 192.168.1.1
```

or

```
ITA# ping 192.168.100.1 source Lo1
```

Note: Testing the default route by creating an unadvertised network on ISP1 and pinging it works only because the default route also points toward ISP1. If the preferred default route pointed toward ISP2, the ping to that unadvertised network on ISP1 would not succeed. If the link to ISP1 failed, the default route to ISP2 would become active, but the pings would be successful only if ISP1 and ISP2 have another working interconnection and appropriate BGP peering between them, which is currently not the case.

Step 7: Using BGP to propagate a default route.

bbb. ISP router will be used to inject a default route via BGP. First, remove the current default routes on ITA.

```
ITA(config)# no ip route 0.0.0.0 0.0.0.0 10.0.0.1 210
ITA(config)# no ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

ccc. Next, configure the ISP1 router to send a default route to its neighbor, the ITA router. This command does not require the presence of 0.0.0.0 in the local ISP1 router.

```
ISP1(config)# router bgp 200
ISP1(config-router)# neighbor 10.0.0.2 default-originate
ISP1(config-router)#
```

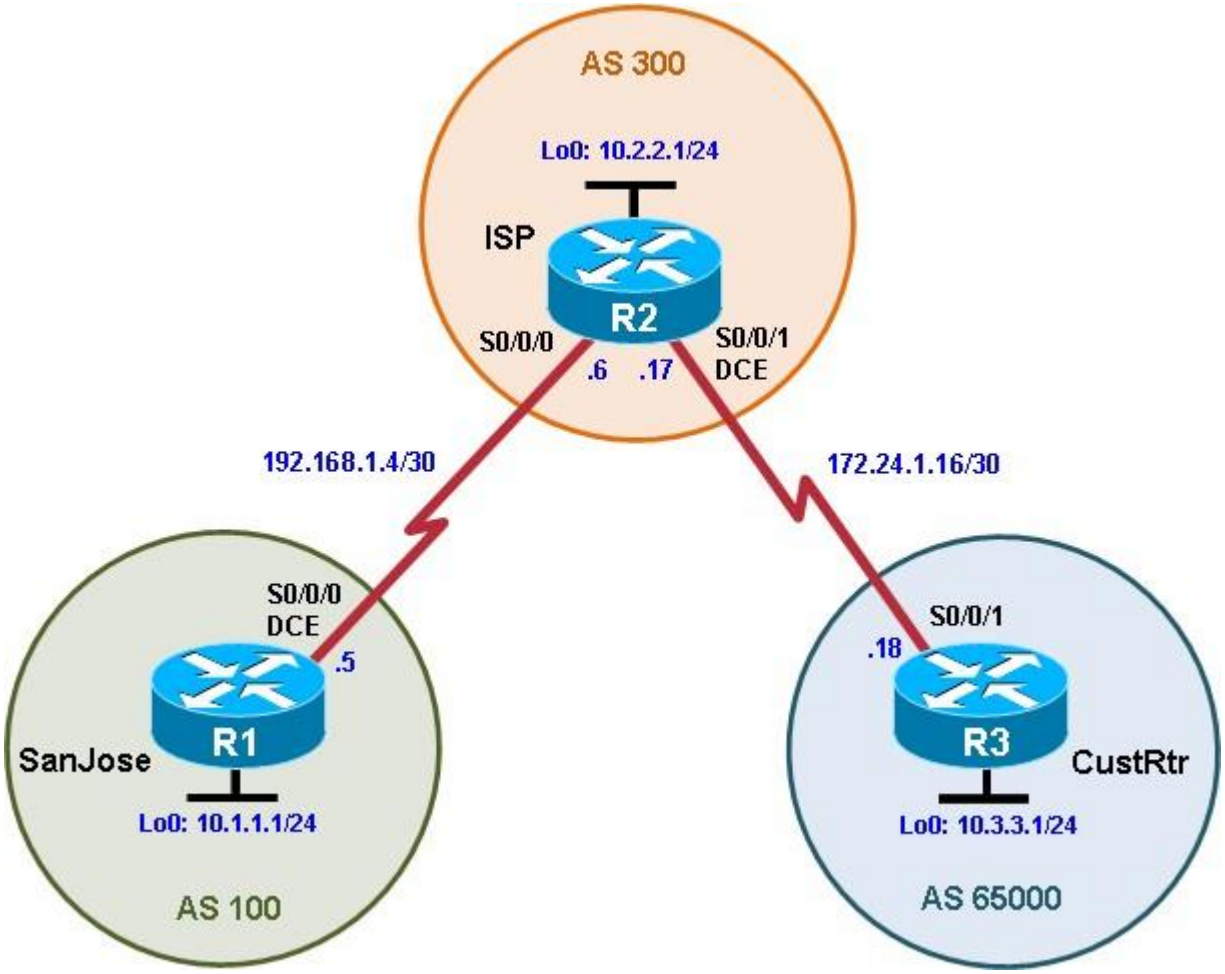
ddd. Verify that the default route was received by ITA using BGP.

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

Gateway of last resort is 10.0.0.1 to network 0.0.0.0

B*    0.0.0.0/0 [20/0] via 10.0.0.1, 00:01:43
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C        10.0.0.0/30 is directly connected, Serial0/0/0
L        10.0.0.2/32 is directly connected, Serial0/0/0
B        10.1.1.0/24 [20/0] via 10.0.0.1, 00:06:51
      172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
C        172.16.0.0/30 is directly connected, Serial0/0/1
L        172.16.0.2/32 is directly connected, Serial0/0/1
B        172.16.1.0/24 [20/0] via 172.16.0.1, 00:06:51
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
```

```
C        192.168.0.0/24 is directly connected, Loopback0
L        192.168.0.1/32 is directly connected, Loopback0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Loopback1
L        192.168.1.1/32 is directly connected, Loopback1
ITA#
```

Chapter 7 Lab 7-2, Using the AS_PATH Attribute

Topology ALVARO ELIU MELO



Objectives

- Use BGP commands to prevent private AS numbers from being advertised to the outside world.

- Use the AS_PATH attribute to filter BGP routes based on their source AS numbers.

Background

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

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

Required Resources

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

Step 0: Suggested starting configurations.

eee. 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.

fff. 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#
```

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

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

Step 2: Configure BGP.

hhh. 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
```

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

```
ISP# show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 65000, external link
  BGP version 4, remote router ID 10.3.3.1
  BGP state = Established, up for 00:00:28
  Last read 00:00:28, last write 00:00:28, hold time is 180, keepalive
interval is 60 seconds
<output omitted>


BGP neighbor is 192.168.1.5,  remote AS 100, external link
  BGP version 4, remote router ID 10.1.1.1
  BGP state = Established, up for 00:01:34
  Last read 00:00:33, last write 00:00:06, hold time is 180, keepalive
interval is 60 seconds
```

```
<output omitted>
```

Step 3: Remove the private AS.

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

```
SanJose#show ip route
Codes: 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

      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, 00:04:22
B        10.3.3.0/24 [20/0] via 192.168.1.6, 00:03:14
      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#
```

kkk.Ping the 10.3.3.1 address from SanJose.

Why does this fail?

Esto falla porque SanJose obtiene el ping con su interfaz más cercana s0 / 0/0 con la dirección IP 192.168.1.5. CustRtr no tiene una ruta de regreso a esa interfaz, por lo que las respuestas de ping no pueden volver a SanJose.

lll. 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 =
28/28/28 ms
SanJose#
```

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

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

or

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

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

```
SanJose# show ip bgp
BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      0.0.0.0                  0         32768 i
 *>  10.2.2.0/24      192.168.1.6              0             0 300
i
 *>  10.3.3.0/24      192.168.1.6                            0 300
65000 i
SanJose#
```

Why is this a problem?

AS 65000 es un AS privado, que no debe publicitarse públicamente en Internet. De lo contrario, los clientes de dos ISP interconectados que tienen el mismo número de AS privado verían sus propios AS en los anuncios de ruta entre ellos. Como resultado, cada cliente concluiría incorrectamente que el anuncio provenía de sí mismo y lo ignoraría.

nnn. 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
```

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

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

```
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 Unicast topology base removed from session  User
reset
*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 Unicast topology base removed from session  User
reset
*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
<output omitted>

      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#
```

Does SanJose still have a route to 10.3.3.0?

Sí, aprendió a través de BGP desde ISP 192.168.1.6.

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

```
SanJose# ping 10.3.3.1 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max =
28/28/32 ms
```

ppp. 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, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      0.0.0.0                  0         32768 i
 *>  10.2.2.0/24      192.168.1.6              0             0 300
i
 *>  10.3.3.0/24      192.168.1.6                            0 300
i
SanJose#
```

Step 4: Use the AS_PATH attribute to filter routes.

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

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

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

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

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

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

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

http://www.cisco.com/c/en/us/td/docs/ios/12_2/termserv/configuration/guide/ftersv_c/tcfaapre.html

http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13754-26.html

rrr. 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.

sss. 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 Unicast topology base removed from session  User
reset
*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 Unicast topology base removed from session  User
reset
*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
<output omitted>

      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#
```

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

```
CustRtr# show ip route
```

```
<output omitted>

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.2.2.0/24 [20/0] via 172.24.1.17, 00:00:32
C        10.3.3.0/24 is directly connected, Loopback0
L        10.3.3.1/32 is directly connected, Loopback0
      172.24.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.24.1.16/30 is directly connected, Serial0/0/1
L        172.24.1.18/32 is directly connected, Serial0/0/1
CustRtr#
```

uuu. 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, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

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

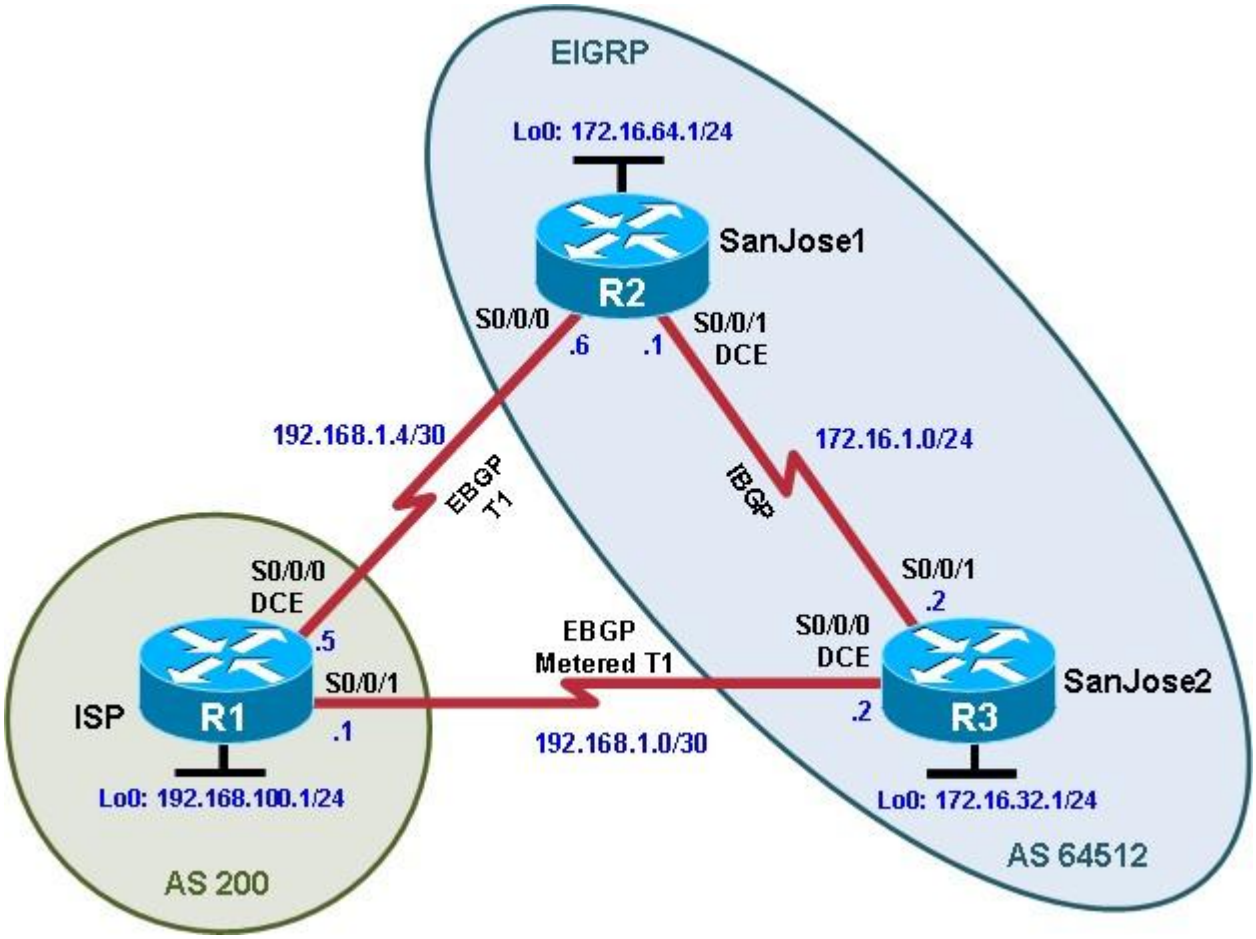
The output of this command shows all matches for the regular expressions that were used in the access list. The path to 10.1.1.0 matches the access list and is filtered from updates to CustRtr.

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

```
ISP# tclsh

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

Chapter 7 Lab 7-3, Configuring IBGP and EBGP Sessions, Local Preference, and MED

Topology ALVARO ELIU MELO



Objectives

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

Background

The International Travel Agency runs BGP on its SanJose1 and SanJose2 routers externally with the ISP router in AS 200. IBGP is run internally between SanJose1 and SanJose2. Your job is to configure both EBGP and IBGP for this internetwork to allow for redundancy. The

metered T1 should only be used in the event that the primary T1 link has failed. Traffic sent across the metered T1 link offers the same bandwidth of the primary link but at a huge expense. Ensure that this link is not used unnecessarily.

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

Required Resources

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

Step 0: Suggested starting configurations.

www. 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.

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

Router R1 (hostname ISP)

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

Router R2 (hostname SanJose1)

```
SanJose1(config)# interface Loopback0
SanJose1(config-if)# ip address 172.16.64.1 255.255.255.0
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/0
SanJose1(config-if)# ip address 192.168.1.6 255.255.255.252
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
```

```
SanJose1(config)# interface Serial0/0/1
SanJose1(config-if)# ip address 172.16.1.1 255.255.255.0
SanJose1(config-if)# clock rate 128000
SanJose1(config-if)# no shutdown
SanJose1(config-if)# end
SanJose1#
```

Router R3 (hostname SanJose2)

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

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

Step 2: Configure EIGRP.

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

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

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

Step 3: Configure IBGP and verify BGP neighbors.

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

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

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

aaaa. 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
```

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

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

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

Step 4: Configure EBGP and verify BGP neighbors.

cccc. 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.

dddd. 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
```

eeee. 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
```

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

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

BGP neighbor is 192.168.1.5,  remote AS 200, external link
  BGP version 4, remote router ID 192.168.100.1
  BGP state = Established, up for 00:06:49
```

```
  Last read 00:00:42, last write 00:00:45, hold time is 180,
keepalive interval is 60 seconds
<output omitted>
```

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

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

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

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

Step 5: View BGP summary output.

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

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

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

Step 6: Verify which path the traffic takes.

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

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

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

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

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

jjjj. 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#
```

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

El ping falla porque SanJose1 no tiene una ruta de regreso a la fuente. La fuente es la interfaz más cercana conectada del ISP de acuerdo con BGP, que en este caso es el enlace s0 / 0/0 con SanJose1. La ruta a la red 172.16.0.0 desde el ISP es a través de

SanJose2, por lo que el ISP puede hacer ping a las interfaces SanJose2 conectadas directamente, pero no a las interfaces SanJose1 directamente conectadas

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

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

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

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

Debido a que todas las demás métricas eran las mismas, la ruta anunciada por el vecino con la ID de enrutador BGP inferior ganó el proceso de selección de ruta BGP. El proceso de identificación del enrutador BGP es el mismo que se utiliza tanto para EIGRP como para OSPF. En ausencia de un comando router-id, los enrutadores utilizan las direcciones de bucle de retorno más altas para sus ID de enrutador. Las ID de enrutador vecinas se muestran usando el comando show ip bgp neighbor. SanJose2 tiene una ID de enrutador BGP inferior de 172.16.32.1 que SanJose1 con una ID de enrutador 172.16.64.1.

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

No, porque BGP no verifica el ancho de banda del enlace en su proceso de selección de ruta.

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

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

```
ISP# ping 172.16.1.1 source 192.168.100.1

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

ISP# ping 172.16.32.1 source 192.168.100.1

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

ISP# ping 172.16.1.2 source 192.168.100.1

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

ISP# ping 172.16.64.1 source 192.168.100.1

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

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

```
ISP# ping
Protocol [ip]:
Target IP address: 172.16.64.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.100.1
Type of service [0]:
Set DF bit in IP header? [no]:
```

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

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

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

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

mmmm. 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
```

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

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

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

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

```
SanJose2# show ip route
```

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

Gateway of last resort is not set

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

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

pppp. 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)#
```

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

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

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


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

Gateway of last resort is not set

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

Notice that SanJose2 has 192.168.100.0 in it's BGP table but not in its routing table. The BGP table shows the next hop to 192.168.100.0 as 192.168.1.5. Because SanJose2 does not have a route to this next hop address of 192.168.1.5 in its routing table, it will not install

the 192.168.100.0 network into the routing table. It won't install a route if it doesn't know how to get to the next hop.

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

rrrr. 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
```

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

```
SanJose1# clear ip bgp *
SanJose1#

SanJose2# clear ip bgp *
SanJose2#
```

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

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

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

uuuu. 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#
```

vvvv. 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.

wwww. 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
```

xxxx. 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.

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

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

     Network          Next Hop            Metric LocPrf Weight Path
 *   172.16.0.0       192.168.1.6              0             0
64512 i
```

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

Gateway of last resort is not set

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

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

El tráfico de retorno seguirá la ruta al enrutador con la ID de enrutador BGP más baja que es SanJose2. Las rutas que se anuncian al ISP tienen las mismas características, por lo que el ISP elige la ruta a través del vecino con la ID del router BGP más baja

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

Como se describe a continuación, puede usar un tipo especial de ping extendido en esta situación. También puede ver qué paquetes de interfaz vienen con el comando del paquete de depuración ip

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

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

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

Reply to request 1 (20 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
    (172.16.1.2)
    (192.168.1.6)
```

```
    (192.168.100.1)
    (192.168.1.1)
    (172.16.32.1) <*>
    (0.0.0.0)
    (0.0.0.0)
    (0.0.0.0)
    (0.0.0.0)
  End of list

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

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

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

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

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

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

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

aaaaa. 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
```

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

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

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

     Network          Next Hop            Metric LocPrf Weight Path
 * i 172.16.0.0       172.16.32.1              0    100      0 i
 *>                   0.0.0.0                  0         32768 i
```

```
 *>  192.168.100.0    192.168.1.5              0    150      0 200
i
SanJose1#

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

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

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

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

```
   (0.0.0.0)
   (0.0.0.0)

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

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

Reply to request 2 (28 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

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

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

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

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

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

Sí. Ahora el ISP prefiere que San José 1 envíe su tráfico de retorno

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

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

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

Step 10: Establish a default route.

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

ddddd. 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)#
```

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

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

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

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

SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
- BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
```

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

Gateway of last resort is 172.16.64.1 to network 0.0.0.0

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

fffff. 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#
```

ggggg. 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#
```

hhhhh. 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)#
```

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

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

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

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

SanJose2# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
- BGP
```

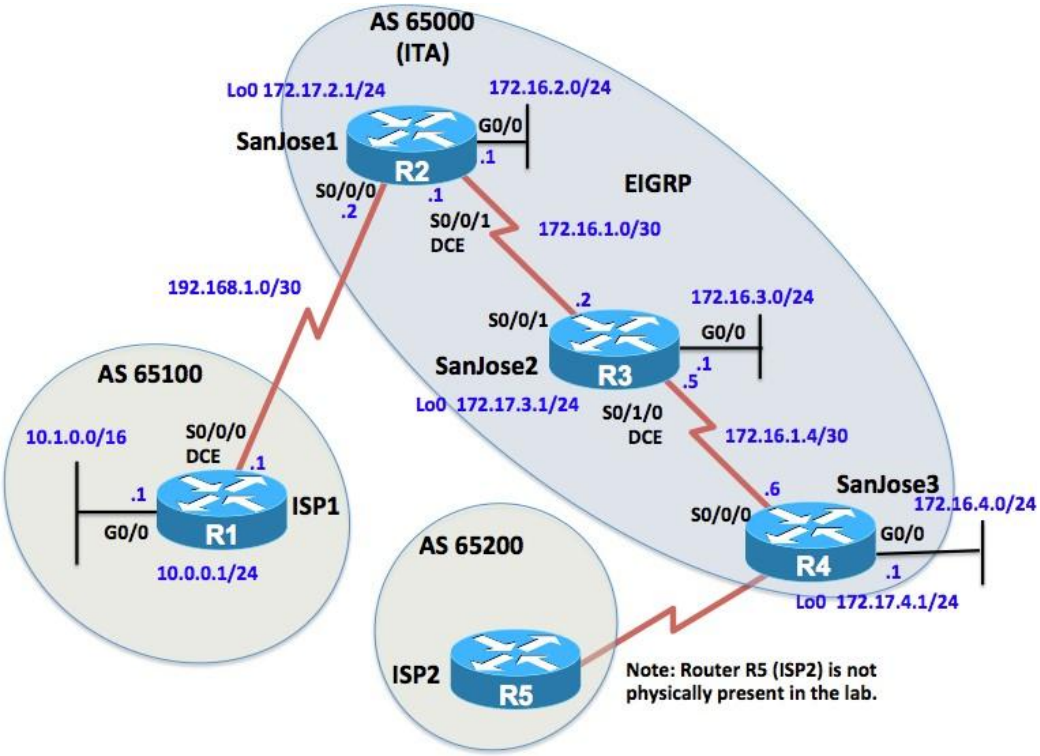
```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user
static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l -
LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

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

jjjjj. 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#
```

Chapter 7 Lab 7-4, IBGP, Next Hop and Synchronization

Topology ALVARO ELIU MELO



Objectives

- Configure EBGP and IBGP .
- Configure EIGRP in the ITA domain.
- Troubleshoot and resolve next hop issues in IBGP.
- Configure full-mesh IBGP to resolve routing issue within ITA domain.
- Configure ITA so it is not a transit AS.
- Verify connectivity.

Background

The International Travel Agency (ITA) runs BGP on its SanJose1 and SanJose3 routers in AS 65000. SanJose1 in AS 65000 is running EBGP with the ISP1 router in AS 65100. SanJose3 in AS 65000 is running EBGP with the ISP2 router in AS 65200. ITA routers need to receive IPv4 networks from both ISPs. To ensure AS 65000 is not a transit AS, SanJose1 and SanJose3 will only include ITA networks 172.16.2.0/24 and 172.16.4.0/24 in its BGP updates to the ISP routers. Your job is to configure EIGRP BGP for this internetwork.

Note: The topology shows SanJose3 in AS 65000 is running EBGP with the ISP2 router in AS 65200. ISP2 (router R5) does not actually exist in the physical lab topology. This is done due to the limitations of four routers in our CCNP NetLab topologies.

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

Required Resources

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

Step 0: Suggested starting configurations.

kkkkk. 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 on all routers and EBGP on ISP1.

llll. 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 ISP1)

```
ISP(config)# interface Loopback0
ISP(config-if)# ip address 10.0.0.1 255.255.255.0
ISP(config-if)# exit
ISP(config)# interface GigabitEthernet0/0
ISP(config-if)# ip address 10.1.0.1 255.255.0.0
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface Serial0/0/0
ISP(config-if)# ip address 192.168.1.1 255.255.255.252
ISP(config-if)# clock rate 64000
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# router bgp 65100
ISP(config-router)# bgp router-id 1.0.0.1
ISP(config-router)# neighbor 192.168.1.2 remote-as 65000
ISP(config-router)# network 10.1.0.0 mask 255.255.0.0
ISP(config-router)#
```

ISP1 has an EBGP peering session with SanJose1. ISP1 is advertising the 10.1.0.0/16 network. A similar BGP configuration is assumed on ISP2, which does not physically exist in this lab topology.

Router R2 (hostname SanJose1)

```
SanJose1(config)# interface Loopback0
SanJose1(config-if)# ip address 172.17.2.1 255.255.255.0
SanJose1(config-if)# exit
SanJose1(config)# interface GigabitEthernet0/0
SanJose1(config-if)# ip address 172.16.2.1 255.255.255.0
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/0
SanJose1(config-if)# ip address 192.168.1.2 255.255.255.252
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/1
SanJose1(config-if)# ip address 172.16.1.1 255.255.255.252
SanJose1(config-if)# clock rate 64000
SanJose1(config-if)# no shutdown
SanJose1(config-if)#
```

Router R3 (hostname SanJose2)

```
SanJose2(config)# interface Loopback0
SanJose2(config-if)# ip address 172.17.3.1 255.255.255.0
SanJose2(config-if)# exit
SanJose2(config)# interface GigabitEthernet0/0
SanJose2(config-if)# ip address 172.16.3.1 255.255.255.0
SanJose2(config-if)# no shutdown
SanJose2(config-if)# exit
SanJose2(config)# interface Serial0/0/1
SanJose2(config-if)# ip address 172.16.1.2 255.255.255.252
SanJose2(config-if)# no shutdown
SanJose2(config-if)# exit
SanJose2(config)# interface Serial0/1/0
SanJose2(config-if)# ip address 172.16.1.5 255.255.255.252
SanJose2(config-if)# clock rate 64000
SanJose2(config-if)# no shutdown
SanJose2(config-if)#
```

Router R4 (hostname SanJose3)

```
SanJose3(config)# interface Loopback0
SanJose3(config-if)# ip address 172.17.4.1 255.255.255.0
SanJose3(config-if)# no shutdown
SanJose3(config-if)# exit
SanJose3(config)# interface Serial0/0/0
SanJose3(config-if)# ip address 172.16.1.6 255.255.255.252
SanJose3(config-if)# no shutdown
SanJose3(config-if)# exit
SanJose3(config)# interface GigabitEthernet0/0
SanJose3(config-if)# ip address 172.16.4.1 255.255.255.0
SanJose3(config-if)# no shutdown
SanJose3(config-if)#
```

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

Step 2: Configure EIGRP on ITA routers.

Configure EIGRP on the SanJose1, SanJose2, and SanJose3 routers. Both routers should be able to ping the other router's LAN and loopback interfaces. (Note: If using an IOS prior to 15.0, use the no auto-summary router configuration command to disable automatic summarization. This command is the default beginning with IOS 15.)

Configure EIGRP for IPv4 and IPv6 on SanJose1.

```
SanJose1(config)# router eigrp 1
SanJose1(config-router)# eigrp router-id 1.1.1.1
SanJose1(config-router)# network 172.16.0.0
SanJose1(config-router)# network 172.17.0.0

SanJose2(config)# router eigrp 1
SanJose2(config-router)# eigrp router-id 2.2.2.2
SanJose2(config-router)# network 172.16.0.0
SanJose2(config-router)# network 172.17.0.0

SanJose3(config)# router eigrp 1
SanJose3(config-router)# eigrp router-id 3.3.3.3
SanJose3(config-router)# network 172.16.0.0
SanJose3(config-router)# network 172.17.0.0
```

nnnnn. Use **ping** to test the reachability between the ITA routers. For example, SanJose3's G0/0 interface should be able to ping SanJose1's G0/0 interface.

```
SanJose3# ping 172.16.2.0 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.0, timeout is 2 seconds:
Packet sent with a source address of 172.16.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
52/55/56 ms
SanJose3#
```

Step 3: Configure BGP on SanJose1 and SanJose3.

ooooo. On SanJose1, configure EBGP to peer with ISP1. ISP1 has already been configured to peer with SanJose1. Configure SanJose1 to IBGP peer with SanJose3 using its loopback0 address. SanJose1 will be advertising the 172.16.2.0/24 network in BGP.

```
SanJose1(config)# router bgp 65000
SanJose1(config-router)# bgp router-id 1.1.1.1
SanJose1(config-router)# neighbor 192.168.1.1 remote-as 65100
SanJose1(config-router)# neighbor 172.17.4.1 remote-as 65000
SanJose1(config-router)# neighbor 172.17.4.1 update-source
Loopback0
SanJose1(config-router)# network 172.16.2.0 mask 255.255.255.0
SanJose1(config-router)#
```

ppppp. Configure SanJose3 to IBGP peer with SanJose1 using its loopback0 address. SanJose3 will be advertising the 172.16.4.0/24 network in BGP.

```
SanJose3(config)# router bgp 65000
SanJose3(config-router)# bgp router-id 3.3.3.3
SanJose3(config-router)# neighbor 172.17.2.1 remote-as 65000
```

```
SanJose3(config-router)# neighbor 172.17.2.1 update-source
Loopback0
SanJose3(config-router)# network 172.16.4.0 mask 255.255.255.0
SanJose3(config-router)#
```

Step 4: Verify BGP on SanJose1.

qqqqq. Examine SanJose1's BGP table using the **show ip bgp** command.

```
SanJose1# show ip bgp
BGP table version is 7, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.0.0/16     192.168.1.1              0             0
65100 i
 *>  172.16.2.0/24   0.0.0.0                  0         32768 i
 r>i 172.16.4.0/24   172.17.4.1               0    100      0 i
SanJose1#
```

Examine Notice that there are three entries in SanJose1's BGP table.

- 10.1.0.0/16 – The status codes "*>" indicate that this network is reachable using the next hop IPaddress 192.168.1.1.
- 172.16.2.0/24 – The status codes "*>" indicate that this network is reachable. The next hop address 0.0.0.0 indicates that this router is originating the network.
- 172.16.4.0/24 - The status "r>i" indicate that this network is reachable. The "r" indicates a RIB failure and the "i" means this entry was learned via IBGP.

Why is there a RIB failure for the 172.16.4.0/24 network? What command would help you determine the cause?

La falla de la RIB se debe a SanJose1 que tiene una mejor fuente de enrutamiento, una menor distancia administrativa, a esta red. El comando show ip bgp rib-failure se puede usar para ayudar a determinar la causa

rrrrr. Use the **show ip bgp rib-failure** command to examine the cause of the RIB failure.

```
SanJose1# show ip bgp rib-failure
  Network            Next Hop                        RIB-failure
RIB-NH Matches
```

```
172.16.4.0/24      172.17.4.1          Higher admin distance
n/a
SanJose1#
```

As you might have answer in the previous question, the RIB failure is due to SanJose1 having a better routing source to this destination. SanJose routers are using EIGRP to share internal ITA networks. IBGP has a higher administrative distance (200) than EIGRP (90), so the EIGRP router is preferred.

sssss. Verify SanJose1's routing table using the **show ip route** command.

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

Gateway of last resort is not set

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [20/0] via 192.168.1.1, 00:14:14
      172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
C        172.16.1.0/30 is directly connected, Serial0/0/1
L        172.16.1.1/32 is directly connected, Serial0/0/1
D        172.16.1.4/30 [90/2681856] via 172.16.1.2, 00:30:41,
Serial0/0/1
C       172.16.2.0/24 is directly connected, GigabitEthernet0/0
L       172.16.2.1/32 is directly connected, GigabitEthernet0/0
D 172.16.3.0/24 [90/2172416] via 172.16.1.2, 00:30:41,
Serial0/0/1
D        172.16.4.0/24 [90/2684416] via 172.16.1.2, 00:29:42,
Serial0/0/1
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.17.2.0/24 is directly connected, Loopback0
L        172.17.2.1/32 is directly connected, Loopback0
D        172.17.3.0/24 [90/2297856] via 172.16.1.2, 00:30:41,
Serial0/0/1
D        172.17.4.0/24 [90/2809856] via 172.16.1.2, 00:29:42,
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
SanJose1#
```

Notice that SanJose1 has a BGP route to 10.1.0.0/16 on ISP1 and an EIGRP route to the 172.16.4.0/24 network on SanJose3.

ttttt. Verify SanJose1's reachability to 10.1.0.0/16 on ISP1.

```
SanJose1# ping 10.1.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
24/27/28 ms
SanJose1#
```

Step 5: Examine and troubleshoot IBGP next hop reachability on SanJose3.

uuuuu. Examine the routing table on SanJose3 using the **show ip route** command.

```
SanJose3# 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, 7 subnets, 3 masks
D        172.16.1.0/30 [90/2681856] via 172.16.1.5, 01:00:03,
Serial0/0/0
C        172.16.1.4/30 is directly connected, Serial0/0/0
L        172.16.1.6/32 is directly connected, Serial0/0/0
D        172.16.2.0/24 [90/2684416] via 172.16.1.5, 01:00:03,
Serial0/0/0
```

```
D        172.16.3.0/24 [90/2172416] via 172.16.1.5, 01:00:03,
Serial0/0/0
C        172.16.4.0/24 is directly connected, GigabitEthernet0/0
L        172.16.4.1/32 is directly connected, GigabitEthernet0/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.17.2.0/24 [90/2809856] via 172.16.1.5, 01:00:03,
Serial0/0/0
D        172.17.3.0/24 [90/2297856] via 172.16.1.5, 01:00:03,
Serial0/0/0
C        172.17.4.0/24 is directly connected, Loopback0
L  172.17.4.1/32  is  directly  connected,  Loopback0
SanJose3#
```

Notice that SanJose3 does not include a route to the 10.1.0.0/16 network on ISP1.

vvvvv. Examine the BGP table on SanJose3 using the show ip bgp command to try and determine the reason why the 10.1.0.0/16 network is not in its routing table.

```
SanJose3# show ip bgp
BGP table version is 3, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric  LocPrf Weight Path
 * i 10.1.0.0/16      192.168.1.1              0     100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0     100      0 i
 *>  172.16.4.0/24    0.0.0.0                  0          32768 i
SanJose3#
```

The output shows that the 10.1.0.0/16 network is in the BGP table but is missing the “>” status code indicating that it is not being offered to the IP routing table. The next hop address used for this route is 192.168.1.1. SanJose3’s routing table in Step 3a shows that SanJose3 does not have a route to this next hop address. If the router does not have a route to the next hop address then the route will not be included in the IP routing table.

In routing, the term “next hop” does not always mean the next hop is a physically adjacent interface. The next hop, as in this case, can be more than one router away.

BGP specifies that routes learned through IBGP are never propagated to other IBGP peers. SanJose1 has learned via EBGP about the 10.1.0.0/16 network from ISP1 with a next hop address of 192.168.1.1, the IP address of ISP1. SanJose1 uses this same next hop address of 192.168.1.1 in its IBGP update to SanJose3.

What are two solutions to this problem?

Una solución es que SanJose1 anuncie la red 192.168.1.0/30 dentro del dominio de enrutamiento EIGRP. Esto se puede hacer mediante la inclusión de un comando de red EIGRP para 192.168.1.0 o redistribuir la red conectada en EIGRP. La segunda solución es que SanJose1 modifique la dirección del próximo salto para las rutas aprendidas de ISP1. Esto necesita ser la dirección propagada por EIGRP, como su dirección loopback0.

wwwww. The decision is made to modify the behavior on SanJose1 so that it uses its loopback0 interface as the next hop address in its IBGP updates.

```
SanJose1(config)# router bgp 65000
SanJose1(config-router)# neighbor 172.17.4.1 next-hop-self
SanJose1(config-router)#
```

Note: For consistency, a similar configuration for SanJose3 is shown. You do **not** need to configure this. This would need to be done if ISP2 router actually existed in our lab topology.

```
SanJose3(config)# router bgp 65000
SanJose3(config-router)# neighbor 172.17.2.1 next-hop-self
SanJose3(config-router)#
```

xxxxx. Re-examine the BGP table on SanJose3 using the **show ip bgp** command to see if SanJose3 now has a valid next hop to the 10.1.0.0/16 network.

```
SanJose3# show ip bgp
BGP table version is 5, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, 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 10.1.0.0/16      172.17.2.1               0    100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0    100      0 i
 *>  172.16.4.0/24    0.0.0.0                  0         32768 i
SanJose3#
```

Notice that the next hope address has been changed to SanJose1's loopback0 address 172.17.2.1 which is reachable because it being advertised in EIGRP updates from SanJose1.

yyyyy. Re-examine the routing table on SanJose3 using the **show ip route** command to see if SanJose3 now has a route to the 10.1.0.0/16 network.

```
SanJose3# 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

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 00:03:17
      172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
D        172.16.1.0/30 [90/2681856] via 172.16.1.5, 01:26:06,
Serial0/0/0
C        172.16.1.4/30 is directly connected, Serial0/0/0
L        172.16.1.6/32 is directly connected, Serial0/0/0
D        172.16.2.0/24 [90/2684416] via 172.16.1.5, 01:26:06,
Serial0/0/0
D        172.16.3.0/24 [90/2172416] via 172.16.1.5, 01:26:06,
Serial0/0/0
C        172.16.4.0/24 is directly connected, GigabitEthernet0/0
L        172.16.4.1/32 is directly connected, GigabitEthernet0/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.17.2.0/24 [90/2809856] via 172.16.1.5, 01:26:06,
Serial0/0/0
D        172.17.3.0/24 [90/2297856] via 172.16.1.5, 01:26:06,
Serial0/0/0
C       172.17.4.0/24 is directly connected, Loopback0
L  172.17.4.1/32  is  directly  connected,  Loopback0
SanJose3#
```

zzzzz. In the previous output, SanJose3 shows a route to the 10.1.0.0/16 network. Verify reachability to this network by pinging ISP1's G0/0 interface.

```
SanJose3# ping 10.1.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

```
SanJose3#
```

Notice that the ping was **not** successful. One reason is because SanJose3 is not advertising the network used as the source IP address in the ping, the 172.16.1.4/30 network.

SanJose3 is advertising its 172.16.4.0/24 network in its BGP updates using the **network** command in its initial BGP configuration. Use the ping command changing the source IP address for the ping to use SanJose3's G0/0 IP address 172.16.4.1.

```
SanJose3# ping 10.1.0.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.4.1
U.U.U
Success rate is 0 percent (0/5)
SanJose3#
```

Even with the correct source IP address the ping does not succeed.

Even though SanJose3 has a route to ISP1's 10.1.0.0/16 network, why do the pings from SanJose3 fail to 10.1.0.1?

El problema radica en SanJose2. SanJose2 desconoce por completo la red 10.1.0.0/16.

Step 4: Examine the behavior of BGP synchronization being disabled.

aaaaaa. The output below reminds us that SanJose3 has an entry in its BGP table and a route in its IP routing table for 10.1.0.0/16.

```
SanJose3# show ip bgp
BGP table version is 5, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, 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 10.1.0.0/16      172.17.2.1               0    100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0    100      0 i
 *>  172.16.4.0/24    0.0.0.0                  0         32768 i
SanJose3#

SanJose3# show ip route bgp
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

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 00:26:43
SanJose3#
```

bbbbbb. Use the ping command to see if SanJose3 can ping the 10.1.0.1 address on ISP1. Notice that the ping fails.

```
SanJose3# ping 10.1.0.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.4.1
U.U.U
```

cccccc. The problem is on SanJose2. SanJose2 does not have a route for 10.1.0.0/16 network as shown using the **show ip route** command.

```
SanJose2# show ip route
<output omitted>

      172.16.0.0/16 is variably subnetted, 8 subnets, 3 masks
C        172.16.1.0/30 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
C        172.16.1.4/30 is directly connected, Serial0/1/0
L        172.16.1.5/32 is directly connected, Serial0/1/0
D        172.16.2.0/24 [90/2172416] via 172.16.1.1, 01:56:50,
Serial0/0/1
C        172.16.3.0/24 is directly connected, GigabitEthernet0/0
L        172.16.3.1/32 is directly connected, GigabitEthernet0/0
D        172.16.4.0/24 [90/2172416] via 172.16.1.6, 01:55:52,
Serial0/1/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.17.2.0/24 [90/2297856] via 172.16.1.1, 01:56:50,
Serial0/0/1
C        172.17.3.0/24 is directly connected, Loopback0
```

```
L        172.17.3.1/32 is directly connected, Loopback0
D        172.17.4.0/24 [90/2297856] via 172.16.1.6, 01:55:52,
Serial0/1/0
SanJose2#
```

Even though SanJose2 does not have any knowledge of the 10.1.0.0/16 network, SanJose3 has the network in its IP routing table because it learned the route via IBGP from SanJose1 and has a valid next hop address to SanJose1 for the route. Even though there is not complete reachability in the ITA for 10.1.0.0/16, SanJose3 still has a IBGP route for this network because the default BGP behavior is **no synchronization**. Beginning with IOS 12.2(8)T, the default BGP behavior is **no synchronization**.

What is the BGP synchronization rule? The BGP synchronization rule states that a router will not include in its routing table nor advertise routes learned by IBGP unless that route is directly connected or learned from an IGP. In other words, with synchronization enabled, SanJose3 will not include the BGP route to 10.1.0.0/16 in its routing table unless it already knows about it via EIGRP. SanJose3 having the 10.1.0.0/16 network in its IP routing table as an EIGRP route would be mean other routers in the domain, SanJose2, most likely have this route also.

Prior to IOS 12.2(8)T **synchronization** was the default behavior. What this meant was that

ddddddd. The affect of this behavior can be examined by enabling synchronization on SanJose3 using the BGP **synchronization** command. The **clear ip bgp *** command is used to rest the neighbor adjacencies.

```
SanJose3(config)# router bgp 65000
SanJose3(config-router)# synchronization
SanJose3(config-router)# end
SanJose3#clear ip bgp *
*Sep 28 18:13:53.007: %BGP-5-ADJCHANGE: neighbor 172.17.2.1 Down
User reset
*Sep 28 18:13:53.007: %BGP_SESSION-5-ADJCHANGE: neighbor 172.17.2.1
IPv4 Unicast topology base removed from session  User reset
*Sep 28 18:13:53.335: %BGP-5-ADJCHANGE: neighbor 172.17.2.1 Up
SanJose3#
```

eeeeee. Using the **show ip bgp** command verify that SanJose3 is still receiving the IBGP update for 10.1.0.0/16 from SanJose1 but no longer is valid.

```
SanJose3# show ip bgp
BGP table version is 3, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
     Network          Next Hop            Metric  LocPrf  Weight  Path
 * i 10.1.0.0/16      172.17.2.1               0     100       0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0     100       0 i
 *>  172.16.4.0/24    0.0.0.0                  0           32768 i
SanJose3#
```

ffffff. With synchronization enable, using the **show ip route** command verify that SanJose3 no longer includes the 10.1.0.0/16 network in its routing table. SanJose3 does not include the 10.1.0.0/16 network in its IP routing table because it does not have this network as an IGP (EIGRP) route in its routing table.

```
SanJose3# show ip route
<output omitted>

      172.16.0.0/16 is variably subnetted, 7 subnets, 3 masks
D        172.16.1.0/30 [90/2681856] via 172.16.1.5, 02:09:59,
Serial0/0/0
C        172.16.1.4/30 is directly connected, Serial0/0/0
L        172.16.1.6/32 is directly connected, Serial0/0/0
D        172.16.2.0/24 [90/2684416] via 172.16.1.5, 02:09:59,
Serial0/0/0
D        172.16.3.0/24 [90/2172416] via 172.16.1.5, 02:09:59,
Serial0/0/0
C        172.16.4.0/24 is directly connected, GigabitEthernet0/0
L        172.16.4.1/32 is directly connected, GigabitEthernet0/0
      172.17.0.0/16 is variably subnetted, 4 subnets, 2 masks
D        172.17.2.0/24 [90/2809856] via 172.16.1.5, 02:09:59,
Serial0/0/0
D        172.17.3.0/24 [90/2297856] via 172.16.1.5, 02:09:59,
Serial0/0/0
C        172.17.4.0/24 is directly connected, Loopback0
L  172.17.4.1/32  is  directly  connected,  Loopback0
SanJose3#
```

With synchronization enable, routes learned via EBGP would need to redistributed into the IGP (EIGRP). SanJose2 would then include the 10.1.0.0/16 network in its IP routing table. Because of the size of Internet routing tables and the potential for this to consume a lot of memory and CPU resources, this is no longer considered best practice .

The better solution is to configure full-mesh IBGP on the transit BGP routers. This is done in the next step.

ggggg. To return to the default configuration, disable synchronization on SanJose3 using the **no synchronization** command and reset the BGP peering. Verify that SanJose3 has returned to this behavior using the **show ip bgp** and **show ip route bgp** commands.

```
SanJose3(config)# router bgp 65000
SanJose3(config-router)# no synchronization
SanJose3(config-router)# end
SanJose3# clear ip bgp *
SanJose3#
*Sep 28 18:25:39.415: %BGP-5-ADJCHANGE: neighbor 172.17.2.1 Down
User reset
*Sep 28 18:25:39.415: %BGP_SESSION-5-ADJCHANGE: neighbor 172.17.2.1
IPv4 Unicast topology base removed from session  User reset
*Sep 28 18:25:40.155: %BGP-5-ADJCHANGE: neighbor 172.17.2.1 Up
SanJose3#
SanJose3# show ip bgp
BGP table version is 4, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 10.1.0.0/16      172.17.2.1               0    100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0    100      0 i
 *>  172.16.4.0/24    0.0.0.0                  0         32768 i
SanJose3# show ip route bgp
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

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 00:09:43
SanJose3#
```

Step 5: Configure and verify full-mesh IBGP on all ITA transit routers.

hhhhhh. With **no synchronization** being the default behavior with IOS, it is important that network administrators ensure IBGP reachability amongst all routers in the transit path. Looking at the topology, notice that SanJose3 has an EBGP peering relationship with ISP2.

(This only exists in the topology but was not configured.) Without any further configuration to filter the 10.1.0.0/16 network, SanJose3 would advertise this network to ISP2. The routing policy on ISP2 could mean that it would forward packets for 10.1.0.0/16 to SanJose3. SanJose2 would then forward them to SanJose2, where they would be dropped because it does not have a route to this network.

The solution is to configure fully-meshed IBGP amongst all routers in the transit path, SanJose1, SanJose2, and SanJose3.

Configure BGP on SanJose2 to have peering relationships with both SanJose1 and SanJose3. Configure SanJose1 and SanJose3 to have a peering relationship with SanJose2. Configure the peering to use the loopback0 addresses.

```
SanJose2(config)# router bgp 65000
SanJose2(config-router)# bgp router-id 2.2.2.2
SanJose2(config-router)# neighbor 172.17.2.1 remote-as 65000
SanJose2(config-router)# neighbor 172.17.2.1 update-source
Loopback0
SanJose2(config-router)# neighbor 172.17.4.1 remote-as 65000
SanJose2(config-router)# neighbor 172.17.4.1 update-source
Loopback0
SanJose2(config-router)#

SanJose1(config)# router bgp 65000
SanJose1(config-router)# neighbor 172.17.3.1 remote-as 65000
SanJose1(config-router)# neighbor 172.17.3.1 update-source
Loopback0
SanJose1(config-router)# neighbor 172.17.3.1 next-hop-self
SanJose1(config-router)#

SanJose3(config)# router bgp 65000
SanJose3(config-router)# neighbor 172.17.3.1 remote-as 65000
SanJose3(config-router)# neighbor 172.17.3.1 update-source
Loopback0
SanJose3(config-router)#
```

Note: Notice that SanJose1 is configured as the next hop for IBGP routes advertised to SanJose2. For consistency, a similar configuration for SanJose3 is shown. Once again, you do **not** need to configure this because the ISP2 router does not actually existed in our lab topology.

```
SanJose3(config)# router bgp 65000
SanJose3(config-router)# neighbor 172.17.3.1 next-hop-self
SanJose3(config-router)#
```

iiiiii. Use the **show bgp summary** command on each router to verify the neighbor adjacencies.

```
ISP1# show bgp summary
BGP router identifier 1.0.0.0, local AS number 65100
BGP table version is 18, main routing table version 18
3 network entries using 432 bytes of memory
3 path entries using 240 bytes of memory
```

```
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1176 total bytes of memory
BGP activity 6/3 prefixes, 8/5 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
192.168.1.2     4        65000      20      21       18    0    0
00:14:25        2
ISP1#

SanJose1# show bgp summary
BGP router identifier 1.1.1.1, local AS number 65000
BGP table version is 4, main routing table version 4
3 network entries using 432 bytes of memory
3 path entries using 240 bytes of memory
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1176 total bytes of memory
BGP activity 7/4 prefixes, 8/5 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
172.17.3.1      4        65000      18      18        4    0    0
00:12:12        0
172.17.4.1      4        65000      19      18        4    0    0
00:12:12        1
192.168.1.1     4        65100      18      18        4    0    0
00:12:12        1
SanJose1#

SanJose2# show bgp summary
BGP router identifier 2.2.2.2, local AS number 65000
BGP table version is 9, main routing table version 9
3 network entries using 432 bytes of memory
3 path entries using 240 bytes of memory
2/2 BGP path/bestpath attribute entries using 320 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1016 total bytes of memory
BGP activity 5/2 prefixes, 5/2 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
172.17.2.1      4        65000      20      20        9    0    0
00:14:30        2
172.17.4.1      4        65000      68      66        9    0    0
00:53:14        1
SanJose2#
```

```
SanJose3# show bgp summary
BGP router identifier 3.3.3.3, local AS number 65000
BGP table version is 10, main routing table version 10
3 network entries using 432 bytes of memory
3 path entries using 240 bytes of memory
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1176 total bytes of memory
BGP activity 11/8 prefixes, 11/8 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
172.17.2.1      4        65000      20      22       10    0    0
00:14:35        2
172.17.3.1      4        65000      66      68       10    0    0
00:53:19        0
SanJose3#
```

jjjjj. Verify that SanJose2 now has the 10.1.0.0/16 network in its BGP table and in its IP routing table, using the **show ip bgp** and **show ip route bgp** commands.

```
SanJose2# show ip bgp
BGP table version is 5, local router ID is 2.2.2.2
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 10.1.0.0/16      172.17.2.1               0    100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0    100      0 i
 r>i 172.16.4.0/24    172.17.4.1               0    100      0 i
SanJose2#

SanJose2# show ip route bgp
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

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 00:06:53
SanJose2#
```

kkkkkk. Verify that SanJose3 still has the 10.1.0.0/16 network in its BGP table and in its IP routing table, using the **show ip bgp** and **show ip route bgp** commands.

```
SanJose3# show ip bgp
BGP table version is 5, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, 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 10.1.0.0/16      172.17.2.1               0    100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0    100      0 i
 *>  172.16.4.0/24    0.0.0.0                  0         32768 i
SanJose3#

SanJose3# show ip route bgp
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

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 00:54:55
SanJose3#
```

llllll. Verify that SanJose3 and ISP1 can now ping their BGP advertised networks from their G0/0 interfaces.

```
SanJose3# ping 10.1.0.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
80/82/84 ms
SanJose3#

ISP1# ping 172.16.4.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
80/83/84 ms
ISP1#
```

We now have complete reachability!

Would the pings succeeded if they were not sourced from their G0/0 interfaces? Why or why not?

Los pings no habrían tenido éxito porque ISP1 no tiene una ruta a la dirección 172.16.4.1 y SanJose3 no tiene una ruta a 192.168.1.1. BGP no está anunciando ninguna red.

Step 6: Configure AS 65000 as a non-transit AS.

mmmmmm.The configuration on SanJose1 and ISP1 allow both routers to exchange BGP learned route. Although this router does not actually exist in our topology, ISP2 and SanJose3 could also be configured to exchange BGP learned routes. This would cause AS 65000 to be a transit AS. BGP routes learned from ISP1 would be advertised by SanJose3 to ISP2, and BGP routes learned from ISP2 would be advertised to by SanJose1 to ISP1.

To avoid being a transit AS, on SanJose1, configure an AS-path filter using an AS-path access list. The access list will only permit locally sourced routes are sent to the provider, ISP1. Routes learned from another AS will be filtered and not included in its updates. This filter is applied to a set of routes announced to the ISP1 neighbor.

The regular expression “^$” matches only routes that are locally sourced, do not contain an AS in its AS-path.

```
SanJose1(config)# router bgp 65000
```

```
SanJose1(config-router)# neighbor 192.168.1.1 filter-list 1 out
SanJose1(config-router)# exit
SanJose1(config)# ip as-path access-list 1 permit ^$
SanJose1(config)#
```

Note: A similar configuration would be done on SanJose3 with its neighbor ISP2.

nnnnnn. Because our topology is not actually include the ISP2 network, we won't see any difference in our outputs. If ISP2 did actually exist, ISP1 and ISP2 would only receive BGP updates from SanJose1 and SanJose3 respectively for the 172.16.2.0/24 and 172.16.4.0/24 networks. For example, ISP2 would not receive a BGP update for the 10.1.0.0/16 from SanJose3.

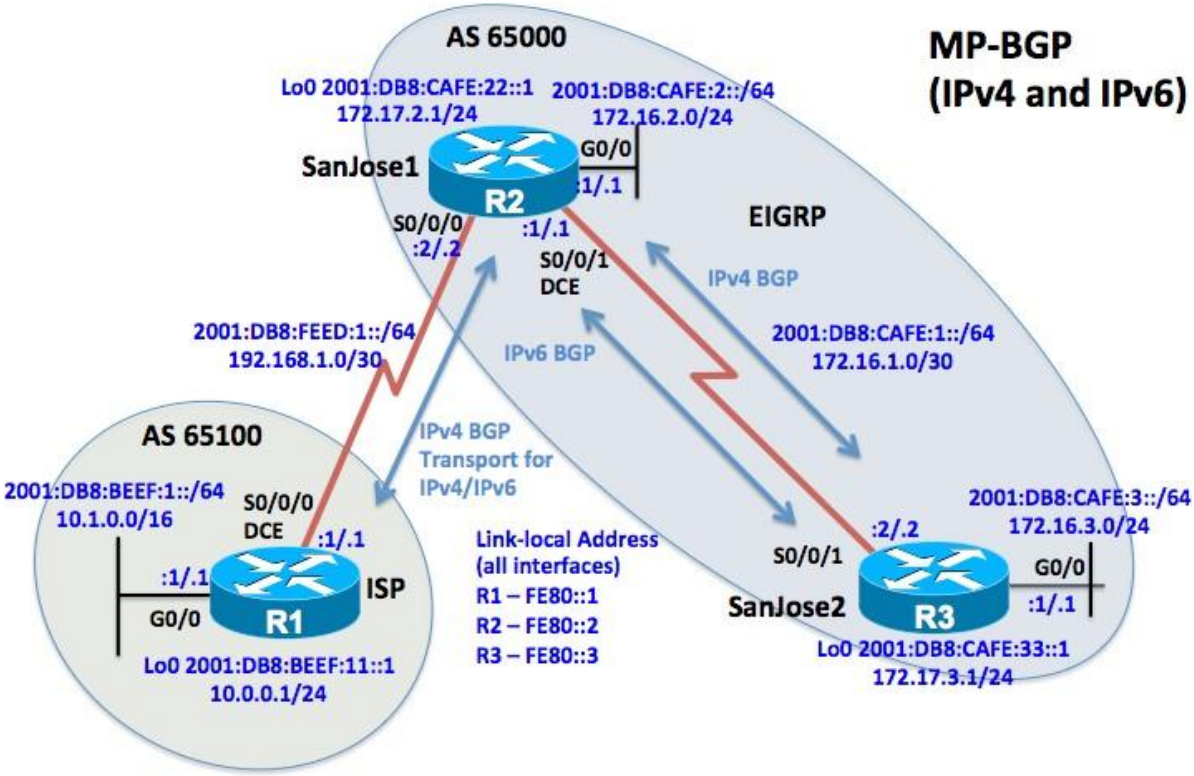
To verify that we have not removed any reachability between AS 65100 and AS 65000, once again use the ping command between ISP1 and SanJose3.

```
ISP1# ping 172.16.4.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.4.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
80/83/84 ms
ISP1#

SanJose3# ping 10.1.0.1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.4.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
80/83/84 ms
SanJose3#
```

FABIAN BERMUDEZ

Chapter 7 Lab 7-5, Configuring MP-BGP

Topology



Objectives

- Configure EIGRP on ITA network.
- Using MP-BGP, configure EBGP for IPv4 and IPv6 between ISP and SanJose1, using IPv4 BGP transport for both protocols.
- Configure MP-BGP IBGP between SanJose1 and SanJose2.
- Verify BGP neighbors, BGP tables and routing tables for IPv4 and IPv6.

Background

SanJose1 in AS 65000 is running MP-BGP with the ISP router in AS 65100. The International Travel Agency runs MP-BGP on its SanJose1 and SanJose2 routers in AS 65000. The International Travel Agency and the ISP need to share both IPv4 and IPv6 prefixes. Your job is to configure MP-BGP for this internetwork. You will need to configure internal and external BGP sessions and advertise IPv6 network prefixes via BGP. You will deploy IPv4 and IPv6 transport. You will use route-maps to set the next-hop attribute to an IPv6 address when exchanging the IPv6 networks over a IPv4 transport session between ISP and SanJose1.

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

Required Resources

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

Step 0: Suggested starting configurations.

oooooo. 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.

pppppp. 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 gigabitethernet 0/0
ISP(config-if)# ip address 10.1.0.1 255.255.0.0
ISP(config-if)# ipv6 address 2001:db8:beef:1::1/64
ISP(config-if)# ipv6 address fe80::1 link-local
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface loopback 0
ISP(config-if)# ip address 10.0.0.1 255.255.255.0
ISP(config-if)# ipv6 address 2001:db8:beef:11::1/64
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface serial 0/0/0
ISP(config-if)# ip address 192.168.1.1 255.255.255.252
ISP(config-if)# ipv6 address 2001:db8:feed:1::1/64
ISP(config-if)# ipv6 address fe80::1 link-local
ISP(config-if)# clock rate 64000
ISP(config-if)# no shutdown
```

Router R2 (hostname SanJose1)

```
SanJose1(config)# interface gigabitethernet 0/0
SanJose1(config-if)# ip address 172.16.2.1 255.255.255.0
SanJose1(config-if)# ipv6 address 2001:db8:cafe:2::1/64
SanJose1(config-if)# ipv6 address fe80::2 link-local
```

```
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface serial 0/0/0
SanJose1(config-if)# ip address 192.168.1.2 255.255.255.252
SanJose1(config-if)# ipv6 address 2001:db8:feed:1::2/64
SanJose1(config-if)# ipv6 address fe80::2 link-local
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface serial 0/0/1
SanJose1(config-if)# ip address 172.16.1.1 255.255.255.252
SanJose1(config-if)# ipv6 address 2001:db8:cafe:1::1/64
SanJose1(config-if)# ipv6 address fe80::2 link-local
SanJose1(config-if)# clock rate 64000
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface loopback 0
SanJose1(config-if)# ip address 172.17.2.1 255.255.255.0
SanJose1(config-if)# ipv6 address 2001:db8:cafe:22::1/64
SanJose1(config-if)# no shutdown
```

Router R3 (hostname SanJose2)

```
SanJose2(config)# interface gigabitethernet 0/0
SanJose2(config-if)# ip address 172.16.3.1 255.255.255.0
SanJose2(config-if)# ipv6 address 2001:db8:cafe:3::1/64
SanJose2(config-if)# ipv6 address fe80::3 link-local
SanJose2(config-if)# no shutdown
SanJose2(config-if)# exit
SanJose2(config)# interface serial 0/0/1
SanJose2(config-if)# ip address 172.16.1.2 255.255.255.252
SanJose2(config-if)# ipv6 address 2001:db8:cafe:1::2/64
SanJose2(config-if)# ipv6 address fe80::3 link-local
SanJose2(config-if)# no shutdown
SanJose2(config)# interface loopback 0
SanJose2(config-if)# ip address 172.17.3.1 255.255.255.0
SanJose2(config-if)# ipv6 address 2001:db8:cafe:33::1/64
```

qqqqqq. Use **ping** to test the connectivity between the directly connected routers for both IPv4 and IPv6. Both SanJose routers should be able to ping each other and SanJose1 should be able to ping the ISP on the serial link IP addresses. The ISP router cannot reach the segment between SanJose1 and SanJose2, or their LAN interfaces.

Step 2: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 routers. Both routers should be able to ping the other router's LAN and loopback interfaces. (Note: If using an IOS prior to 15.0, use the no auto-summary router configuration command to disable automatic summarization. This command is the default beginning with IOS 15.)

Configure EIGRP for IPv4 and IPv6 on SanJose1.

```
SanJose1(config)# ipv6 unicast-routing
SanJose1(config)# router eigrp 1
SanJose1(config-router)# eigrp router-id 2.2.2.2
SanJose1(config-router)# network 172.16.0.0
```

```
SanJose1(config-router)# network 172.17.0.0

SanJose1(config)# ipv6 router eigrp 2
SanJose1(config-rtr)# eigrp router-id 2.2.2.2

SanJose1(config)# interface gigabitethernet 0/0
SanJose1(config-if)# ipv6 eigrp 2
SanJose1(config-if)# exit
SanJose1(config)# interface serial 0/0/1
SanJose1(config-if)# ipv6 eigrp 2
SanJose1(config-if)# exit
SanJose1(config)# interface loopback 0
SanJose1(config-if)# ipv6 eigrp 2
```

Configure EIGRP for IPv4 and IPv6 on SanJose2.

```
SanJose2(config)# ipv6 unicast-routing
SanJose2(config)# router eigrp 1
SanJose2(config-router)# eigrp router-id 3.3.3.3
SanJose2(config-router)# network 172.16.0.0
SanJose2(config-router)# network 172.17.0.0


SanJose2(config)# ipv6 router eigrp 2
SanJose2(config-rtr)# eigrp router-id 3.3.3.3

SanJose2(config)# interface gigabitethernet 0/0
SanJose2(config-if)# ipv6 eigrp 2
SanJose2(config-if)# exit
SanJose2(config)# interface serial 0/0/1
SanJose2(config-if)# ipv6 eigrp 2
SanJose2(config-if)# exit
SanJose2(config)# interface loopback 0
SanJose2(config-if)# ipv6 eigrp 2
```

Step 3: Configure MP-BGP on ISP – EBGP.

rrrrrr. Configure EBGP between the ISP and SanJose1. ISP and SanJose1 will be using IPv4 as the BGP transport for both IPv4 and IPv6 sessions. After enabling IPv6 routing on ISP, configure BGP for AS 65100 with a router ID of 1.1.1.1. In its peering with SanJose1, the IPv4 address of SanJose1 will be used for the IPv4 BGP transport session.

```
ISP(config)# ipv6 unicast-routing
ISP(config)# router bgp 65100
ISP(config-router)# bgp router-id 1.1.1.1
ISP(config-router)# neighbor 192.168.1.2 remote-as 65000
```

ssssss. Enter the router configuration mode for the IPv4 address family. Enter the commands to advertise the 10.1.0.0/16 network and activate the IPv4 neighbor 192.1681.2 within the IPv4 AF.

```
ISP(config-router)# address-family ipv4 unicast
ISP(config-router-af)# network 10.1.0.0 mask 255.255.0.0
ISP(config-router-af)# neighbor 192.168.1.2 activate
ISP(config-router-af)# exit-address-family
```

ttttt. Enter the router configuration mode for the IPv6 address family and enter the command to advertise the 2001:DB8:BEEF:1::/64 prefix. Since you are using IPv4 as the BGP transport, you must also activate the IPv4 neighbor 192.168.1.2 within the IPv6 AF. Configure the route-map NEXT-HOP-IPV6 to attach to the BGP neighbor in the outbound direction. Outbound direction means that this information in the route-map will be applied to IPv6 BGP updates as they are sent to SanJose1.

```
ISP(config-router)# address-family ipv6 unicast
ISP(config-router-af)# network 2001:DB8:BEEF:1::/64
ISP(config-router-af)# neighbor 192.168.1.2 activate
ISP(config-router-af)# neighbor 192.168.1.2 route-map NEXT-HOP-IPV6
out
ISP(config-router-af)# exit-address-family
```

The route-map is applied in the outbound direction. What will this do?

Dirección de salida significa que esta información en el mapa de rutas se aplicará a las actualizaciones IPv6 BGP a medida que se envían a SanJose1.

---

uuuuuu. The route-map NEXT-HOP-IPV6 is configured to overwrite the next-hop parameter with the appropriate IPv6 next-hop address. Notice that the next-hop address is the local IPv6 address of this router, ISP. The neighbor, SanJose1, will use this IPv6 address as it's next-hop address in its IPv6 BGP table.

```
ISP(config)# route-map NEXT-HOP-IPV6 permit 10
ISP(config-route-map)# set ipv6 next-hop 2001:DB8:FEED:1::1
```

Step 4: Configure MP-BGP on SanJose1 – EBGP and IBGP.

vvvvvv. Enable IPv6 routing on SanJose1 and then configure BGP for AS 65000 with a router ID of 2.2.2.2. The IPv4 address of ISP will be used for the IPv4 BGP transport session with ISP.

```
SanJose1(config)# router bgp 65000
SanJose1(config-router)# bgp router-id 2.2.2.2
SanJose1(config-router)# neighbor 192.168.1.1 remote-as 65100
```

wwwwww. Configure IBGP on SanJose1 to peer with SanJose2 for both IPv4 and IPv6. The **update-source loopback 0** command instructs the router to use the IP address of the interface loopback 0 as the source IP address for all BGP messages sent to that neighbor. The IP address of the loopback interface is used in the **neighbor** command.

```
SanJose1(config-router)# neighbor 2001:DB8:CAFE:33::1 remote-as
65000
SanJose1(config-router)# neighbor 2001:DB8:CAFE:33::1 update-source
Loopback0
SanJose1(config-router)# neighbor 172.17.3.1 remote-as 65000
SanJose1(config-router)# neighbor 172.17.3.1 update-source
Loopback0
```

xxxxxx. Enter the router configuration mode for the IPv4 address family for SanJose1. Enter the command to advertise the 172.16.2.0/24 network. Activate the IPv4 neighbor within the IPv4 AF for the EBGP peering session with ISP.

```
SanJose1(config-router)# address-family ipv4 unicast
SanJose1(config-router-af)# network 172.16.2.0 mask 255.255.255.0
SanJose1(config-router-af)# neighbor 192.168.1.1 activate
```

yyyyyy. Activate the IPv4 neighbor within the IPv4 AF for the IBGP peering session with SanJose2. Configure the **next-hop-self** parameter so SanJose1 uses its own IPv4 address as the next-hop address in its IBGP updates to SanJose2. By default, SanJose1 would include the next-hop address the ISP in it IBGP updates to SanJose2. This would be for any routes learned from ISP using EBGP.

```
SanJose1(config-router-af)# neighbor 172.17.3.1 activate
SanJose1(config-router-af)# neighbor 172.17.3.1 next-hop-self
SanJose1(config-router-af)# exit-address-family
```

zzzzzz. Enter the router configuration mode for the IPv6 address family and enter the command to advertise the 2001:DB8:CAFE:2::/64 prefix. Similar to ISP, since you are using IPv4 as our BGP transport, you must also activate the IPv4 neighbor within the IPv6 AF. Configure the route-map NEXT-HOP-IPV6 to attach to the BGP neighbor in the outbound direction. Outbound direction means that this information in the route-map will be applied to IPv6 BGP updates as they are sent to ISP.

```
SanJose1(config-router)# address-family ipv6 unicast
SanJose1(config-router-af)# network 2001:DB8:CAFE:2::/64
SanJose1(config-router-af)# neighbor 192.168.1.1 activate
SanJose1(config-router-af)# neighbor 192.168.1.1 route-map NEXT-
HOP-IPV6 out
```

aaaaaaa. Activate the IPv6 neighbor within the IPv6 AF for the IBGP peering session with SanJose2. Similar to BGP for IPv4, configure the **next-hop-self** parameter so SanJose1 uses its own IPv6 address as the next-hop address in its IBGP updates to SanJose2.

```
SanJose1(config-router-af)# neighbor 2001:DB8:CAFE:33::1 activate
SanJose1(config-router-af)# neighbor 2001:DB8:CAFE:33::1 next-hop-self
SanJose1(config-router-af)# exit-address-family
```

If the **next-hop-self** parameter is not used, what needs to be done to ensure reachability to routes advertised by the ISP router?

SanJose2 necesita tener una ruta a la siguiente dirección de salto, interfaz S0 / 0/0 del ISP

bbbbbbb. Because SanJose1 is using IPv4 transport in its peering with ISP, the route-map NEXT-HOP-IPV6 is configured to overwrite the next-hop parameter with the appropriate IPv6 next-hop address. Notice that the next-hop address is the IPv6 address of SanJose1. Using its local IPv6 address in the route-map, neighbor ISP will use this IPv6 address as it's next-hop address in its IPv6 BGP table.

```
SanJose1(config)# route-map NEXT-HOP-IPV6 permit 10
SanJose1(config-route-map)# set ipv6 next-hop 2001:DB8:FEED:1::2
```

Step 4: Configure MP-BGP on SanJose2 – IBGP.

ccccccc. Enable IPv6 routing on SanJose2 and then configure BGP for AS 65000 with a router ID of 3.3.3.3.

```
SanJose2(config)# router bgp 65000
SanJose2(config-router)# bgp router-id 3.3.3.3
```

ddddddd. Configure IBGP on SanJose2 to peer with SanJose1 for both IPv4 and IPv6. SanJose2's loopback 0 interface will be used in the peering for both IPv4 and IPv6.

```
SanJose2(config-router)# neighbor 2001:DB8:CAFE:22::1 remote-as 65000
SanJose2(config-router)# neighbor 2001:DB8:CAFE:22::1 update-source Loopback0
SanJose2(config-router)# neighbor 172.17.2.1 remote-as 65000
SanJose2(config-router)# neighbor 172.17.2.1 update-source Loopback0
```

eeeeeee. Enter the router configuration mode for the IPv4 address family for SanJose2. Enter the command to advertise the 172.16.3.0/24 network. Activate the IPv4 neighbor within the IPv4 AF for the IBGP peering session with SanJose1.

```
SanJose2(config-router)# address-family ipv4 unicast
SanJose2(config-router-af)# network 172.16.3.0 mask 255.255.255.0
SanJose2(config-router-af)# neighbor 172.17.2.1 activate
SanJose2(config-router-af)# exit-address-family
```

fffffff. Enter the router configuration mode for the IPv6 address family and enter the command to advertise the 2001:DB8:CAFE:3::/64 prefix. Activate the IPv6 neighbor within the IPv6 AF for the IBGP peering session with SanJose1.

```
SanJose2(config-router)# address-family ipv6 unicast
SanJose2(config-router-af)# network 2001:DB8:CAFE:3::/64
SanJose2(config-router-af)# neighbor 2001:DB8:CAFE:22::1 activate
SanJose2(config-router-af)# exit-address-family
```

Step 5: Verifying BGP neighbor peering relationships for IPv4 and IPv6.

ggggggg. Use the **show bgp all neighbors** command on SanJose1 to display information about BGP connections to neighbors for all (IPv4 and IPv6) address families. Each neighbor shows that it is in the "Established" state indicating the router can send and receive BGP messages.

SanJose1 has two neighbor addresses, ISP and SanJose2, for each address family, IPv4 and IPv6. The internal link is the IBGP neighbor relationship with SanJose2 whereas the external link is the EBGP neighbor relationship with ISP. Notice for the IPv6 address family, there is SanJose2's IPv6 address and ISP's IPv4 address. SanJose1 employs IPv6 as the IBGP transport with SanJose2, using the IPv6 address with 2001:DB8:CAFE:33::1. IPv4 is used as the EBGP transport with ISP, so the IPv4 address 192.168.1.1 is shown for the IPv6 address family.

```
SanJose1# show bgp all neighbors
For address family: IPv4 Unicast
BGP neighbor is 172.17.3.1,  remote AS 65000, internal link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 03:47:09
  Last read 00:00:33, last write 00:00:53, hold time is 180,
keepalive interval is 60 seconds
  <output omitted>

BGP neighbor is 192.168.1.1,  remote AS 65100, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 03:47:17
  Last read 00:00:19, last write 00:00:53, hold time is 180,
keepalive interval is 60 seconds
  <output omitted>


For address family: IPv6 Unicast
BGP neighbor is 2001:DB8:CAFE:33::1,  remote AS 65000, internal
link
  BGP version 4, remote router ID 3.3.3.3
  BGP state = Established, up for 03:47:25
  Last read 00:00:38, last write 00:00:04, hold time is 180,
keepalive interval is 60 seconds
 <output omitted>
```

```
BGP neighbor is 192.168.1.1,  remote AS 65100, external link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 03:47:43
  Last read 00:00:46, last write 00:00:19, hold time is 180,
keepalive interval is 60 seconds
 <output omitted>

SanJose1#
```

What is the relationship between the “remote AS” and whether it is an internal or external link?

Un enlace interno tendrá el mismo AS que el vecino. Un enlace externo tendrá un AS diferente que el vecino.

---

hhhhhhh. Another option to the **show bgp all neighbors** command, are the **show ip bgp neighbors** and **show bgp ipv6 unicast neighbors** commands which can be used for their respective address families. An excerpt from **show ip bgp neighbors** command is displayed below. In this command, the IPv6 address family information not only displays the IPv4 address used as the transport, but the name of the route map that was used on SanJose1 end of the connection.

```
SanJose1# show ip bgp neighbors
<output omitted>

 For address family: IPv6 Unicast
  Session: 192.168.1.1
  BGP table version 12, neighbor version 12/0
  Output queue size : 0
  Index 11, Advertise bit 0
  11 update-group  member
  Outbound path policy configured
  Route map for outgoing advertisements is NEXT-HOP-IPV6
  Slow-peer detection is disabled
  Slow-peer split-update-group dynamic is disabled
  Interface associated: Serial0/0/0
  <output omitted>

SanJose1#
```

iiiiiii. Use the **show bgp ipv4 unicast summary** and **show bgp ipv6 unicast summary** commands on ISP to display a summary of IPv4/IPv6 peering information with SanJose1. The **show bgp ipv4 unicast summary** is the equivalent of **show ip bgp** and either command can be used. Notice that BGP connectivity for both IPv4 and IPv6 is over an IPv4 BGP transport session, using the neighbor address of 192.168.1.2.

```
ISP# show bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 65100
BGP table version is 21, main routing table version 21
3 network entries using 432 bytes of memory
3 path entries using 240 bytes of memory
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1176 total bytes of memory
BGP activity 9/3 prefixes, 18/12 paths, scan interval 60 secs

Neighbor        V       AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down State/PfxRcd
192.168.1.2     4    65000      80      78       21    0    0
01:03:46        2
ISP#


ISP# show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 65100
BGP table version is 23, main routing table version 23
3 network entries using 504 bytes of memory
3 path entries using 312 bytes of memory
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1320 total bytes of memory
BGP activity 9/3 prefixes, 18/12 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd
192.168.1.2     4        65000      86      85       23    0    0
01:09:28        2
ISP#
```

jjjjjj. Use the **show bgp ipv4 unicast summary** command on SanJose1 to display a summary of IPv4 peering information with ISP and SanJose2. Notice that SanJose1 has two IPv4 peers, one in each AS. Also notice, that the IBGP peering relationship with SanJose2 uses SanJose2's loopback address 172.17.3.1. This is why this network was included in the EIGRP configuration on SanJose1 and SanJose2.

```
SanJose1# show bgp ipv4 unicast summary
BGP router identifier 2.2.2.2, local AS number 65000
BGP table version is 6, main routing table version 6
3 network entries using 432 bytes of memory
3 path entries using 240 bytes of memory
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
```

```
BGP using 1176 total bytes of memory
BGP activity 23/17 prefixes, 25/19 paths, scan interval 60 secs

Neighbor        V           AS  MsgRcvd  MsgSent    TblVer   InQ  OutQ
Up/Down  State/PfxRcd
172.17.3.1      4        65000       93       93         6     0     0
01:19:50        1
192.168.1.1     4        65100       98       98         6     0     0
01:19:50        1
SanJose1#
```

kkkkkkk. Use the **show bgp ipv6 unicast summary** command on SanJose1 to display a summary of IPv6 peering information with ISP and SanJose2. Similar to IPv4, notice that SanJose1 has two peers, one in each AS. However, the IPv6 peering session with ISP in AS 65100 uses IPv4 as its transport, so the IPv4 neighbor address 192.168.1.1 is displayed.

```
SanJose1# show bgp ipv6 unicast summary
BGP router identifier 2.2.2.2, local AS number 65000
BGP table version is 8, main routing table version 8
3 network entries using 504 bytes of memory
3 path entries using 312 bytes of memory
3/3 BGP path/bestpath attribute entries using 480 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1320 total bytes of memory
BGP activity 23/17 prefixes, 25/19 paths, scan interval 60 secs

Neighbor        V           AS  MsgRcvd  MsgSent    TblVer   InQ  OutQ
Up/Down  State/PfxRcd
2001:DB8:CAFE:33::1
                4        65000       93       97         8     0     0
01:19:59        1
192.168.1.1     4        65100       98       98         8     0     0
01:19:59        1
SanJose1#
```

Step 6: Verifying the BGP tables for IPv4 and IPv6.

lllllll. Use the **show bgp ipv4 unicast** command on ISP to display its IPv4 BGP table. This command is equivalent to the **show ip bgp** command and either command can be used. Notice that ISP shows three IPv4 networks in its IPv4 BGP table. Each network is valid "*" and has one path which is the best path ">". Amongst other information, the next hop IPv4 address and the AS path are included.

```
ISP# show bgp ipv4 unicast
BGP table version is 22, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
```

```
                r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
                x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.0.0/16      0.0.0.0                  0         32768 i
 *>  172.16.2.0/24    192.168.1.2              0             0
65000 i
 *>  172.16.3.0/24    192.168.1.2                            0
65000 i
ISP#
```

mmmmmmm. Use the **show bgp ipv6 unicast** command on ISP to display its IPv6 BGP table. Similar to the BGP table for IPv4, notice that ISP shows three IPv6 prefixes in its IPv6 BGP table. Each network is a valid "*" and has one path which is the best path ">". The next hop IPv6 address and AS path are also included.

Notice that the next-hop address for the prefixes 2001:DB8:CAFE:2::/64 and 2001:DB8:CAFE:3::/64, advertised by SanJose1, is using the address from SanJose1's NEXT-HOP-IPV6 route-map, 2001:DB8:FEED:1::2.

```
ISP# show bgp ipv6 unicast
BGP table version is 26, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
                r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
                x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  2001:DB8:BEEF:1::/64
                      ::                       0         32768 i
 *>  2001:DB8:CAFE:2::/64
                      2001:DB8:FEED:1::2
                                               0             0
65000 i
 *>  2001:DB8:CAFE:3::/64
                      2001:DB8:FEED:1::2
                                                             0
65000 i
ISP#
```

nnnnnnn. Using the **show bgp ipv4 unicast** command on SanJose1 displays information for its IPv4 BGP table. Both the 10.1.0.0/24 network learned via EBGP from the ISP, and its own advertised network of 172.16.2.0/24 are included.

Notice that the 172.16.3.0/24 and 2001:DB8:CAFE:3::/64 prefixes do not include the "*" indicating best path, but rather the "r" signifying a RIB (routing information base) failure. Although these prefixes are being advertised by IBGP with an administrative distance of

200, the router is preferring the EIGRP source with a lower administrative distance of 90. Therefore, the EIGRP route is the preferred source and will be the one added to the IPv4 routing table.

```
SanJose1# show bgp ipv4 unicast
BGP table version is 6, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.0.0/16      192.168.1.1              0             0
65100 i
 *>  172.16.2.0/24    0.0.0.0                  0         32768 i
 r>i 172.16.3.0/24    172.17.3.1               0    100      0 i
SanJose1#
```

ooooooo. Similarly, the **show bgp ipv6 unicast** command on SanJose1 displays its IPv6 BGP table. ISP shows three IPv6 prefixes in its IPv6 BGP table. Each network is a valid "*" and has one path which is the best path ">". Amongst other information, the next hop IPv6 address and AS path are included.

Similar to its IPv4 BGP table, notice that the next-hop address for the prefix 2001:DB8:BEEF:1::/64, advertised by SanJose1, is using the address from ISP's NEXT-HOP-IPV6 route-map, 2001:DB8:FEED:1::1.

```
SanJose1# show bgp ipv6 unicast
BGP table version is 8, local router ID is 2.2.2.2
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
 *>  2001:DB8:BEEF:1::/64
                      2001:DB8:FEED:1::1
                                               0             0
65100 i
 *>  2001:DB8:CAFE:2::/64
                      ::                       0         32768 i
 r>i 2001:DB8:CAFE:3::/64
                      2001:DB8:CAFE:33::1
                                               0    100      0 i
SanJose1#
```

ppppppp. Similar BGP table output is shown for SanJose2. Notice that the next hop address is the loopback interface of SanJose1. In SanJose1's peering configuration with SanJose2, SanJose1 uses the **next-hop-self** option and its loopback address 172.17.2.1. Remember, without this option IBGP carries EBGP routes into the domain with the next hop address unchanged – the next hop address of the ISP in this case.

```
SanJose2# show bgp ipv4 unicast
BGP table version is 22, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 10.1.0.0/16      172.17.2.1               0    100      0
65100 i
 r>i 172.16.2.0/24    172.17.2.1               0    100      0 i
 *>  172.16.3.0/24    0.0.0.0                  0         32768 i
SanJose2#

SanJose2# show bgp ipv6 unicast
BGP table version is 24, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i
- internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f
RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 2001:DB8:BEEF:1::/64
                      2001:DB8:CAFE:22::1
                                               0    100      0
65100 i
 r>i 2001:DB8:CAFE:2::/64
                      2001:DB8:CAFE:22::1
                                               0    100      0 i
 *>  2001:DB8:CAFE:3::/64
                      ::                       0         32768 i
SanJose2#
```

Step 6: Verifying the IP routing tables for IPv4 and IPv6.

qqqqqqq. By examining the IPv4 and IPv6 routing tables on ISP you can verify that BGP is receiving the IPv4 and IPv6 prefixes from SanJose1.

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

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
C        10.0.0.0/24 is directly connected, Loopback0
L        10.0.0.1/32 is directly connected, Loopback0
C        10.1.0.0/16 is directly connected, GigabitEthernet0/0
L        10.1.0.1/32 is directly connected, GigabitEthernet0/0
      172.16.0.0/24 is subnetted, 2 subnets
B        172.16.2.0 [20/0] via 192.168.1.2, 02:59:37
B        172.16.3.0 [20/0] via 192.168.1.2, 03:00:10
      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        192.168.1.1/32 is directly connected, Serial0/0/0
ISP#


ISP# show ipv6 route
IPv6 Routing Table - default - 9 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static
route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D -
EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF
ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext
2
       a - Application
C   2001:DB8:BEEF:1::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:DB8:BEEF:1::1/128 [0/0]
     via GigabitEthernet0/0, receive
C   2001:DB8:BEEF:11::/64 [0/0]
     via Loopback0, directly connected
L   2001:DB8:BEEF:11::1/128 [0/0]
```

```
     via Loopback0, receive
B    2001:DB8:CAFE:2::/64 [20/0]
     via FE80::2, Serial0/0/0
B    2001:DB8:CAFE:3::/64 [20/0]
     via FE80::2, Serial0/0/0
C    2001:DB8:FEED:1::/64 [0/0]
     via Serial0/0/0, directly connected
L    2001:DB8:FEED:1::1/128 [0/0]
     via Serial0/0/0, receive
L    FF00::/8 [0/0]
     via Null0, receive
ISP#
```

Are the BGP routes learned via EBGP or EBGP? How can you tell by just the information in the routing table?

Las rutas se aprenden a través de EBGP. La distancia administrativa de 20 indica EBGP como la información de enrutamiento de origen.

rrrrrrr. Examine the IPv4 and IPv6 routing tables on SanJose1. SanJose1 is EIGRP receiving routes from SanJose2 for SanJose2's LAN and networks. Using EBGP, SanJose1 is receiving IPv4 and IPv6 prefixes from the ISP.

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

Gateway of last resort is not set

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [20/0] via 192.168.1.1, 03:01:19
      172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
C        172.16.1.0/30 is directly connected, Serial0/0/1
L        172.16.1.1/32 is directly connected, Serial0/0/1
C        172.16.2.0/24 is directly connected, GigabitEthernet0/0
L        172.16.2.1/32 is directly connected, GigabitEthernet0/0
D 172.16.3.0/24 [90/2172416] via 172.16.1.2, 04:17:47,
Serial0/0/1
```

```
      172.17.0.0/16 is variably subnetted, 3 subnets, 2 masks
C        172.17.2.0/24 is directly connected, Loopback0
L        172.17.2.1/32 is directly connected, Loopback0
D        172.17.3.0/24 [90/2297856] via 172.16.1.2, 04:17:47,
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
SanJose1#


SanJose1# show ipv6 route
IPv6 Routing Table - default - 12 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static
route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D -
EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF
ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext
2
       a - Application
B   2001:DB8:BEEF:1::/64 [20/0]
     via FE80::1, Serial0/0/0
C   2001:DB8:CAFE:1::/64 [0/0]
     via Serial0/0/1, directly connected
L   2001:DB8:CAFE:1::1/128 [0/0]
     via Serial0/0/1, receive
C   2001:DB8:CAFE:2::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:DB8:CAFE:2::1/128 [0/0]
     via GigabitEthernet0/0, receive
D   2001:DB8:CAFE:3::/64 [90/2172416]
     via FE80::3, Serial0/0/1
C   2001:DB8:CAFE:22::/64 [0/0]
     via Loopback0, directly connected
L   2001:DB8:CAFE:22::1/128 [0/0]
     via Loopback0, receive
D   2001:DB8:CAFE:33::/64 [90/2297856]
     via FE80::3, Serial0/0/1
C   2001:DB8:FEED:1::/64 [0/0]
     via Serial0/0/0, directly connected
L   2001:DB8:FEED:1::2/128 [0/0]
     via Serial0/0/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
SanJose1#
```

sssssss. Looking at the IPv4 and IPv6 routing tables on SanJose2 shows that SanJose2 is receiving EIGRP and BGP routes from SanJose1. SanJose1's LAN and loopback interfaces

are being advertised to SanJose2 using EIGRP and for ISPs IPv4/IPv6 prefixes, SanJose1 is advertising them using IBGP.

```
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

      10.0.0.0/16 is subnetted, 1 subnets
B        10.1.0.0 [200/0] via 172.17.2.1, 03:02:16
      172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
C        172.16.1.0/30 is directly connected, Serial0/0/1
L        172.16.1.2/32 is directly connected, Serial0/0/1
D        172.16.2.0/24 [90/2172416] via 172.16.1.1, 04:18:45,
Serial0/0/1
C        172.16.3.0/24 is directly connected, GigabitEthernet0/0
L        172.16.3.1/32 is directly connected, GigabitEthernet0/0
      172.17.0.0/16 is variably subnetted, 3 subnets, 2 masks
D        172.17.2.0/24 [90/2297856] via 172.16.1.1, 04:18:45,
Serial0/0/1
C        172.17.3.0/24 is directly connected, Loopback0
L  172.17.3.1/32  is  directly  connected,  Loopback0
SanJose2#
```

```
SanJose2# show ipv6 route
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static
route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D -
EIGRP
       EX - EIGRP external, ND - ND Default, NDp - ND Prefix, DCE -
Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF
ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext
2
       a - Application
```

```
B    2001:DB8:BEEF:1::/64 [200/0]
      via  2001:DB8:CAFE:22::1
C    2001:DB8:CAFE:1::/64 [0/0]
      via Serial0/0/1, directly connected
L    2001:DB8:CAFE:1::2/128 [0/0]
      via Serial0/0/1, receive
D    2001:DB8:CAFE:2::/64 [90/2172416]
      via FE80::2, Serial0/0/1
C    2001:DB8:CAFE:3::/64 [0/0]
      via GigabitEthernet0/0, directly connected
L    2001:DB8:CAFE:3::1/128 [0/0]
      via GigabitEthernet0/0, receive
D    2001:DB8:CAFE:22::/64 [90/2297856]
      via FE80::2, Serial0/0/1
C    2001:DB8:CAFE:33::/64 [0/0]
      via Loopback0, directly connected
L    2001:DB8:CAFE:33::1/128 [0/0]
      via Loopback0, receive
L    FF00::/8 [0/0]
      via Null0, receive
SanJose2#
```

Are the BGP routes learned via EBGP or EBGP? How can you tell by just the information in the routing table?

Ellos son aprendidas via EBGP.

---

tttttt. Verify IPv4 and IPv6 reachability pinging ISP's LAN interface from the LAN interface on SanJose2.

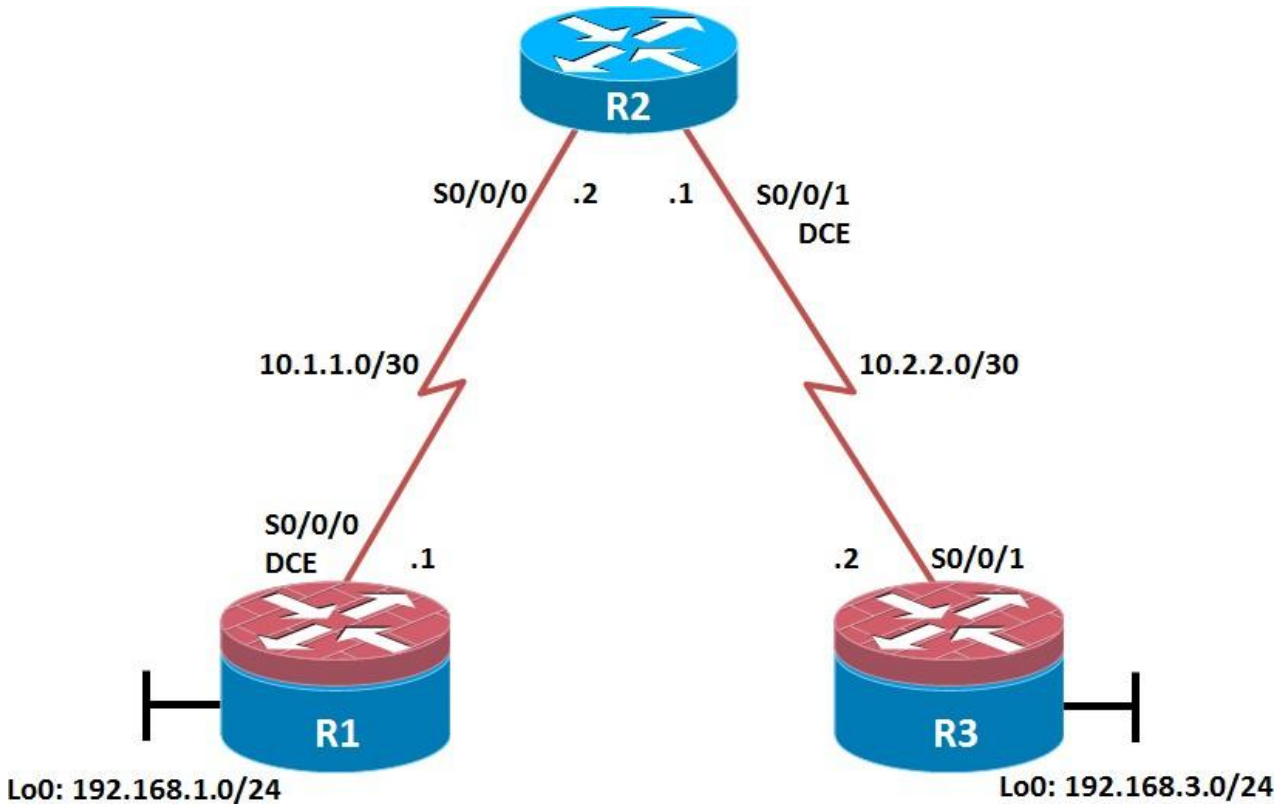
```
SanJose2# ping 10.1.0.1 source 172.16.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.0.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.3.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
56/56/60 ms
SanJose2#
```

```
SanJose2# ping 2001:db8:beef:1::1 source gig 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:BEEF:1::1, timeout is 2
seconds:
Packet sent with a source address of 2001:DB8:CAFE:3::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
52/55/56 ms
SanJose2#
```

CCNPv7 ROUTE

Chapter 8 Lab 8-1, Secure the Management Plane

Topology MARILIN VELASCO



Objectives

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

Background

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

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

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

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

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

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

R1

```
hostname R1


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

```
exit
!
end
```

R2

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

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

R3

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

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

Step 2: Configure static routes.

d. 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
```

e. 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
```

f. 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
```

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

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

R1# tclsh

R1(tcl)#foreach address {
```

```
+>(tcl)#192.168.1.1

+>(tcl)#10.1.1.1

+>(tcl)#10.1.1.2

+>(tcl)#10.2.2.1

+>(tcl)#10.2.2.2

+>(tcl)#192.168.3.1

+>(tcl)#} { ping $address }

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
1/2/4 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/15/16 ms

R1(tcl)#
```

Are the pings now successful?

_____

_____

Yes. If not, troubleshoot.

Step 3: Secure management access.

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

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

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

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

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

_____
_____

_____
_____

_____
_____

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

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

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

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

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

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

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

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

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

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

_____

_____

_____

_____

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

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

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

Yes. They are all in clear text.

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

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

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

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

No. The passwords are now encrypted.

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

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

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

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

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

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

uuuuuuu.

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

Step 4: Configure enhanced username password security.

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

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

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

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

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

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

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

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

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

vvvvvvv.

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

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

```
User Access Verification

Username: ADMIN
Password:
R3>
```

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

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

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

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

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

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

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

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

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

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

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

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

d. Assign both RADIUS servers to a server group.

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

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

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

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

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

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

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

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

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

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

wwwwwww.

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

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

User Access Verification

Username: admin
Password:

% Authentication failed

Username: ADMIN
Password:

R3>
```

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

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

Step 6: Enabling secure remote management using SSH.

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

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

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

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

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

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

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

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

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

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

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

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

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

xxxxxxx.

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).

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

zzzzzzz. Configure SSH version 2 on R1.

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

aaaaaaaa.

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

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

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

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

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

```
R1# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQC3Lehh7ReYlgyDzls6wq+mFzxqzoaZFr9XGx+
Q/yio
dFYw00hQo80tZy1W1Ff3Pz6q7Qi0y00urwddHZ0kBZceZK9EzJ6wZ+9a87KKDETCWrG
SLi6c8lE/y4K+
Z/oVrMMZk7bpTM1MFdP41YgkTf35utYv+TcqbsYo++KJiYk+xw==
R1#
```

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

bbbbbbbb.

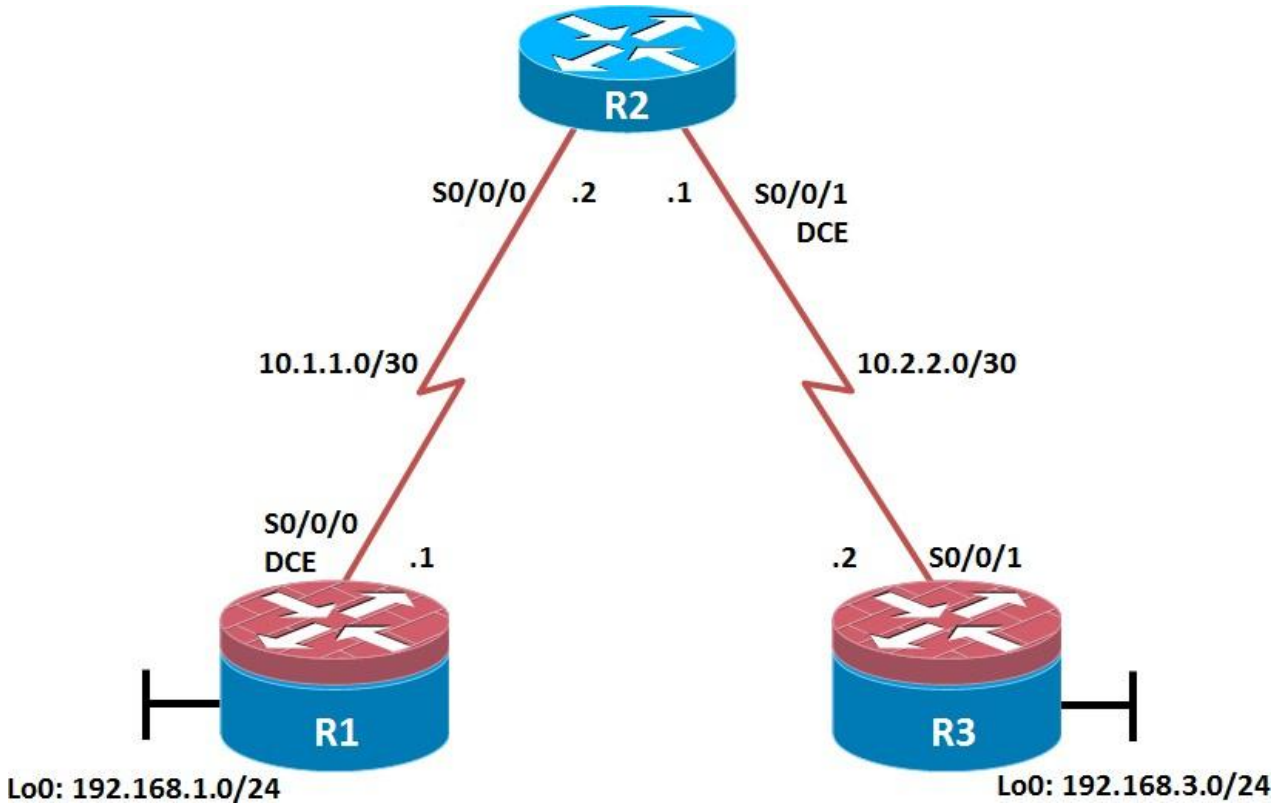
h. Although a user can SSH from a host using the SSH option of TeraTerm of PuTTY, a router can also SSH to another SSH enabled device. SSH to R3 from R1.

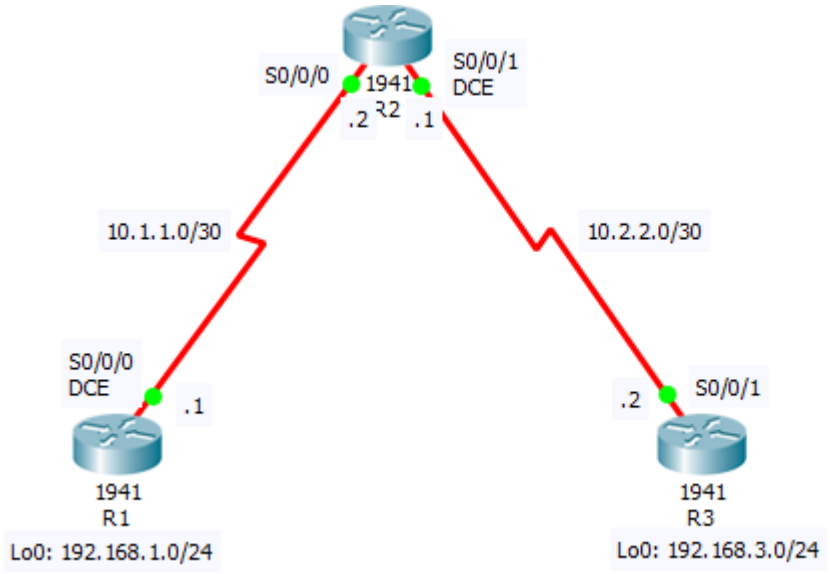
```
R1# ssh -l ADMIN 10.2.2.2
Password:
Unauthorized access strictly prohibited!
R3>
R3> en
Password:
R3#
```

CCNPv7 ROUTE

Chapter 8 Lab 8-2, Routing Protocol Authentication

Topology MARILIN VELASCO





Objectives

- Secure EIGRP routing protocol using SHA authentication.
- Secure OSPF routing protocol using SHA authentication.

Background

The

In this lab, you build a multi-router network and secure the routing protocols used between R1, R2, and R3.

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

Required Resources

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

Step 1: Configure loopbacks and assign addresses.

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

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

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

R1

```
hostname R1


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

```
!
end
```

R2

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

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

R3

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


interface Serial0/0/1
```

```
 description R3 --> R2
 ip address 10.2.2.2 255.255.255.252
 no shutdown
exit
!
end
```

Step 2: Configure named EIGRP routing.

EIGRP SHA authentication can only be configured when using the named EIGRP method. In this step, you will configure named EIGRP.

h. On R1, configure named EIGRP.

```
R1(config)# router eigrp ROUTE
R1(config-router)# address-family ipv4 autonomous-system 1
R1(config-router-af)# network 10.1.1.0 0.0.0.3
R1(config-router-af)# network 192.168.1.0 0.0.0.255
R1(config-router-af)#
```

i. On R2, configure named EIGRP.

```
R2(config)# router eigrp ROUTE
R2(config-router)# address-family ipv4 autonomous-system 1
R2(config-router-af)# network 10.1.1.0 0.0.0.3
R2(config-router-af)# network 10.2.2.0 0.0.0.3
R2(config-router-af)#
Jan 10 10:10:59.823: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1:
Neighbor 10.1.1.1 (Serial0/0/0) is up: new adjacency
R2(config-router-af)#
```

j. On R3, configure named EIGRP.

```
R3(config)# router eigrp ROUTE
R3(config-router)# address-family ipv4 autonomous-system 1
```

```
R3(config-router-af)# network 10.2.2.0 0.0.0.3

R3(config-router-af)# network 192.168.3.0 0.0.0.255

R3(config-router-af)#

Jan 10 10:10:58.795: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1:
Neighbor 10.2.2.1 (Serial0/0/1) is up: new adjacency

R3(config-router-af)#
```

k. Verify the routing table of R1.

```
R1# show ip route eigrp | begin Gateway

Gateway of last resort is not set


      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

D        10.2.2.0/30 [90/23796062] via 10.1.1.2, 00:05:56,
Serial0/0/0

D     192.168.3.0/24 [90/23796702] via 10.1.1.2, 00:05:44,
Serial0/0/0

R1#
```

cccccccc.

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

```
foreach address {

192.168.1.1

10.1.1.1

10.1.1.2

10.2.2.1

10.2.2.2

192.168.3.1

} { ping $address }


R1(tcl)#foreach address {

+>(tcl)#192.168.1.1

+>(tcl)#10.1.1.1
```

```
+>(tcl)#10.1.1.2

+>(tcl)#10.2.2.1

+>(tcl)#10.2.2.2

+>(tcl)#192.168.3.1

+>(tcl)#} { ping $address }

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2
seconds:

!!!!!
```

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2
seconds:

!!!!!

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

R1(tcl)#
```

Are the pings now successful?

Step 3: Secure the named EIGRP routing process.

m. On R1, create the key chain to be used for authentication.

```
R1(config)# key chain NAMED-R1-Chain
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-1
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)#
```

n. Next, enable authentication on the serial 0/0/0 interface of R1.

```
R1(config)# router eigrp ROUTE
R1(config-router)# address-family ipv4 autonomous-system 1
R1(config-router-af)# af-interface S0/0/0
R1(config-router-af-interface)# authentication key-chain NAMED-R1-
Chain
R1(config-router-af-interface)# authentication mode hmac-sha-256
secret-2
R1(config-router-af-interface)#
Jan 10 10:19:35.035: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor
10.1.1.2 (Serial0/0/0) is down: authentication HMAC-SHA-256
configured
R1(config-router-af-interface)#
```

Notice how the adjacency with R2 has changed to down. This is because R1 no longer accepts the updates from R2 because they are not authenticated.

ddddddddd.

o. On R2, create the key chain to be used for authentication.

```
R2(config)# key chain NAMED-R1-Chain
R2(config-keychain)# key 1
R2(config-keychain-key)# key-string secret-1
```

```
R2(config-keychain-key)# exit
R2(config-keychain)# exit
R2(config)#
```

p. Next, enable authentication on the serial 0/0/0 and serial 0/0/1 interfaces of R2.

```
R2(config)# router eigrp ROUTE
R2(config-router)# address-family ipv4 autonomous-system 1
R2(config-router-af)# af-interface S0/0/0
R2(config-router-af-interface)# authentication key-chain NAMED-R2-Chain
R2(config-router-af-interface)# authentication mode hmac-sha-256 secret-2
R2(config-router-af-interface)# exit
R2(config-router-af)# af-interface S0/0/1
R2(config-router-af-interface)# authentication key-chain NAMED-R2-Chain
R2(config-router-af-interface)# authentication mode hmac-sha-256 secret-2
R2(config-router-af-interface)#
Jan 10 10:22:03.299: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.2.2.2 (Serial0/0/1) is down: authentication HMAC-SHA-256 configured
R2(config-router-af-interface)#
Jan 10 10:22:05.503: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.1.1.1 (Serial0/0/0) is up: new adjacency
R2(config-router-af-interface)#
```

Notice how the first informational message is saying that the adjacency with R3 has changed to down. This is because R2 no longer accepts the updates from R3 because they are not authenticated.

However, the second information message is saying that the adjacency with R1 has been restored because they are now authenticating each other's routing updates.

eeeeeeee.

q. On R3, create the key chain to be used for authentication.

```
R3(config)# key chain NAMED-R1-Chain
R3(config-keychain)# key 1
R3(config-keychain-key)# key-string secret-1
R3(config-keychain-key)# exit
R3(config-keychain)# exit
R3(config)#
```

r. Next, enable authentication on the serial 0/0/1 interface of R3.

```
R3(config)# router eigrp ROUTE
```

```
R3(config-router)# address-family ipv4 autonomous-system 1
R3(config-router-af)# af-interface S0/0/1
R3(config-router-af-interface)# authentication key-chain NAMED-R3-Chain
R3(config-router-af-interface)# authentication mode hmac-sha-256 secret-2
R3(config-router-af-interface)#
Jan 10 10:28:17.455: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 10.2.2.1 (Serial0/0/1) is up: new adjacency
R3#
```

m. Verify the routing table of R1.

```
R1#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D        10.2.2.0/30 [90/23796062] via 10.1.1.2, 00:08:18, Serial0/0/0
D     192.168.3.0/24 [90/23796702] via 10.1.1.2, 00:01:56, Serial0/0/0
R1#
```

ffffffff.

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

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

```
R1(tcl)#foreach address {
```

```
+>(tcl)#192.168.1.1

+>(tcl)#10.1.1.1

+>(tcl)#10.1.1.2

+>(tcl)#10.2.2.1

+>(tcl)#10.2.2.2

+>(tcl)#192.168.3.1

+>(tcl)#} { ping $address }

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2
seconds:

!!!!!

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

R1(tcl)#
```

Are the pings now successful?

o. Next we will configure OSPF routing protocol authentication. Therefore, remove EIGRP from R1, R2, and R3 using the **no router eigrp ROUTE** command on all three routers.

```
R1(config)# no router eigrp ROUTE

R1(config)
```

Step 4: Configure OSPF routing.

Since Cisco IOS Software Release 15.4(1)T, OSPFv2 supports SHA hashing authentication using key chains. Cisco refers to this as OSPFv2 Cryptographic Authentication feature. The feature prevents unauthorized or invalid routing updates in a network by authenticating OSPFv2 protocol packets using HMAC-SHA algorithms.

a. On R1, configure OSPF.

```
R1(config)# router ospf 1
R1(config-router)# network 192.168.1.0 0.0.0.255 area 0
R1(config-router)# network 10.1.1.0 0.0.0.3 area 0
R1(config-router)#
```

b. On R2, configure OSPF.

```
R2(config)# router ospf 1
R2(config-router)# network 10.1.1.0 0.0.0.3 area 0
R2(config-router)# network 10.2.2.0 0.0.0.3 area 0
R2(config-router)#
```

c. On R3, configure OSPF.

```
R1(config)# router ospf 1
R1(config-router)# network 192.168.3.0 0.0.0.255 area 0
R1(config-router)# network 10.2.2.0 0.0.0.3 area 0
R1(config-router)#
```

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

```
foreach address {
192.168.1.1
10.1.1.1
10.1.1.2
10.2.2.1
10.2.2.2
```

```
192.168.3.1
} { ping $address }


R1(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#10.1.1.1
+>(tcl)#10.1.1.2
+>(tcl)#10.2.2.1
+>(tcl)#10.2.2.2
+>(tcl)#192.168.3.1
+>(tcl)#} { ping $address }
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
24/27/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2
seconds:
```

```
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2
seconds:

!!!!!

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

R1(tcl)#
```

Are the pings now successful?

Step 5: Secure the OSPF routing protocol.

OSPF will use the OSPFv2 Cryptographic Authentication.

a. On R1, create the key chain to be used for OSPF authentication.

```
R1(config)# key chain SHA-CHAIN
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string secret-1
R1(config-keychain-key)# cryptographic-algorithm hmac-sha-256
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)#
```

b. Next, enable authentication on the serial 0/0/0 interface of R1.

```
R1(config)# interface s0/0/0
R1(config-if)# ip ospf authentication key-chain SHA-CHAIN
R1(config-if)#
Jan 10 11:08:34.075: %OSPF-5-ADJCHG: Process 1, Nbr 10.2.2.1 on
Serial0/0/0 from FULL to DOWN, Neighbor Down: Dead timer expired
```

Notice how the adjacency with R2 has changed to down. This is because R1 no longer accepts the updates from R2 because they are not authenticated.

ggggggg.

c. On R2, create the key chain to be used for authentication.

```
R2(config)# key chain SHA-CHAIN
R2(config-keychain)# key 1
R2(config-keychain-key)# key-string secret-1
R2(config-keychain-key)# cryptographic-algorithm hmac-sha-256
R2(config-keychain-key)# exit
R2(config-keychain)# exit
R2(config)#
```

d. Next, enable authentication on the serial 0/0/0 and serial 0/0/1 interfaces of R2.

```
R2(config)# interface s0/0/0
R2(config-if)# ip ospf authentication key-chain SHA-CHAIN
R2(config-if)# exit
R2(config)#
R2(config)# interface s0/0/1
R2(config-if)# ip ospf authentication key-chain SHA-CHAIN
R2(config-if)#
Jan 10 11:08:42.523: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on
Serial0/0/0 from LOADING to FULL, Loading Done
R2(config-if)#
Jan 10 11:09:14.487: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.3.1 on
Serial0/0/1 from FULL to DOWN, Neighbor Down: Dead timer expired
```

Notice how the first informational message is saying that the adjacency with R1 has been restored because they are now authenticating each other's routing updates.

However, the second information message is saying that the adjacency with R3 has changed to down. This is because R2 no longer accepts the updates from R3 because they are not authenticated.

hhhhhhhh.

e. On R3, create the key chain to be used for authentication.

```
R3(config-router)# key chain SHA-CHAIN
R3(config-keychain)# key 1
R3(config-keychain-key)# key-string secret-1
R3(config-keychain-key)# cryptographic-algorithm hmac-sha-256
R3(config-keychain-key)# exit
R3(config-keychain)# exit
R3(config)#
```

f. Next, enable authentication on the serial 0/0/1 interface of R3.

```
R3(config)#interface s0/0/1
R3(config-if)#ip ospf authentication key-chain SHA-CHAIN
R3(config-if)#
Jan 10 11:09:20.223: %OSPF-5-ADJCHG: Process 1, Nbr 10.2.2.1 on
Serial0/0/1 from LOADING to FULL, Loading Done
R3#
```

e. Verify the routing table of R1.

```
R1# show ip route ospf | begin Gateway

Gateway of last resort is not set


      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

O        10.2.2.0/30 [110/128] via 10.1.1.2, 00:05:23,
Serial0/0/0

      192.168.3.0/32 is subnetted, 1 subnets

O        192.168.3.1 [110/129] via 10.1.1.2, 00:04:23,
Serial0/0/0

R1#
```

iiiiiiii.

f. Verify the routing table of R1.

jjjjjjjj.

```
R1# show ip ospf interface s0/0/0 | section Crypto

  Cryptographic authentication enabled

    Sending SA: Key 1, Algorithm HMAC-SHA-256 - key chain
SHA-CHAIN

R1#
```

kkkkkkkk.

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

```
foreach address {

192.168.1.1

10.1.1.1

10.1.1.2
```

```
10.2.2.1

10.2.2.2

192.168.3.1

} { ping $address }


R1(tcl)#foreach address {

+>(tcl)#192.168.1.1

+>(tcl)#10.1.1.1

+>(tcl)#10.1.1.2

+>(tcl)#10.2.2.1

+>(tcl)#10.2.2.2

+>(tcl)#192.168.3.1

+>(tcl)#} { ping $address }

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2
seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max =
12/14/16 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2
seconds:

!!!!!

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

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2
seconds:

!!!!!

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

R1(tcl)#
```

Are the pings now successful?

CONCLUSIONES

1. Este trabajo nos ha permitido evidenciar el avance de nuestros conocimientos en lo que se refiere a los protocolos de enrutamiento a nivel de WAN.
2. Los laboratorios prácticos nos han permitido afianzar los procedimientos necesarios para las configuraciones a nivel de WAN (ISP)
3. Las actividades practica que hemos ejecutado en esta fase del curso nos permitió fortalecer y complementar las habilidades en las configuraciones a nivel de Routing.

BIBLIOGRAFIA

Amberg, E. (2014). CCNA 1 Powertraining : ICND1/CCENT (100-101). Heidleberg: MITP. Recuperado de: http://bibliotecavirtual.unad.edu.co:2051/login.aspx?direct=true&db=e000xww&AN=979032&lang=es&site=ehost-live

Lammle, T. (2010). CISCO Press (Ed). Cisco Certified Network Associate Study Guide. Recuperado de: http://gonda.nic.in/swangonda/pdf/ccna1.pdf