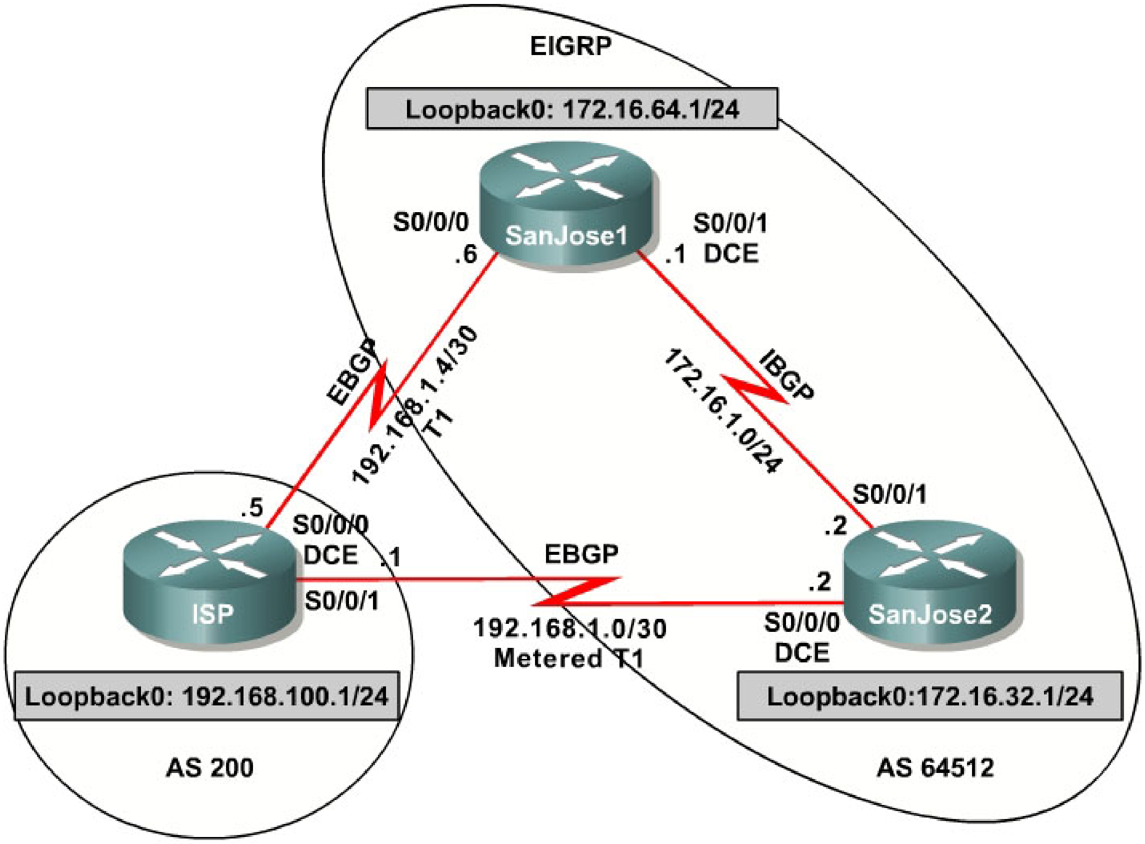
Lab 6-3 Configuring IBGP and EBGP Sessions, Local Preference and MED

Topology Diagram



Learning Objectives

In this lab, you will configure both IBGP and EBGP. For IBGP peers in this lab to correctly exchange routing information, the **next-hop-self** command must be used along with the **Local-Preference** and **MED** attributes. This is to insure that the flat-rate, unlimited-use T1 link is used for sending and receiving data to and from the AS 200 on ISP. 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.

Scenario

The International Travel Agency runs BGP on its SanJose1 and SanJose2 routers externally with ISP, 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.

Step 1: IP Addressing

Build and configure the network according to the diagram, but do not configure a routing protocol. Configure a loopback interface on the SanJose1 and SanJose2 routers as shown. These loopbacks will be used with BGP **neighbor** statements for increased stability.

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

Step 2: Configure EIGRP

Configure EIGRP between the SanJose1 and SanJose2 routers with the same commands:

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

Step 3: Configure IBGP

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 neighbor exist, the router can use any IP interface to communicate by way of BGP. The **update-source lo0** command instructs the router to use interface loopback 0 for TCP connections. This command offers greater fault tolerance if one of the potentially numerous links within the corporate EIGRP WAN cloud fails. For simplicity in the lab environment, only one link needs to be configured.

Step 4: Verify BGP Neighbors

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
```

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.

The link between SanJose1 and SanJose2 should indicate an internal link as shown in the following:

```
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 00:00:01
```

Step 5: Configure EBGP

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.

Step 6: Verify BGP Neighbors

Configure SanJose1 as an EBGP peer to ISP:

```
SanJose1(config)#ip route 172.16.0.0 255.255.0.0 null0
SanJose1(config)#router bgp 64512
SanJose1(config-router)#neighbor 192.168.1.5 remote-as 200
SanJose1(config-router)#network 172.16.0.0
```

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

Step 7: View BGP Summary Output

Configure SanJose2 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
```

In Step 6, 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 2, main routing table version 2
1 network entries and 1 paths using 137 bytes of memory
1 BGP path attribute entries using 60 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 activity 2/1 prefixes, 2/1 paths, scan interval 15 secs
```

```
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
172.16.64.1     4 64512      21      24        2    0    0 00:03:02
0
192.168.1.1     4   200      14      15        2    0    0 00:03:36
0
```

Step 8: Verify Which Path Traffic Takes

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.

Now ping from ISP to the Loopback 0 address of 172.16.32.1 on SanJose2 and the serial link between SanJose1 and SanJose2. This time try 172.16.1.2.

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 is this the case?

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

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

Notice that ISP has two valid routes to the 172.16.0.0 network, as indicated by the *. However, the link to SanJose2, the metered T1, has been selected as the best path. While that may be better for the ISP, a premium is paid for each megabyte transferred across this link.

Was this a malicious attempt by the ISP to get more money? Why did the ISP prefer the link to SanJose2 over SanJose1?

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

BGP operates differently than all other protocols. Unlike other routing protocols that may 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 upon 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 IP address. This was not a malicious attempt by the ISP to get additional funds. In fact, this ISP router was configured from the beginning. The SanJose2 router with address 192.168.1.2 was preferred to the higher IP address of the SanJose1 router, 192.168.1.6.

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.

```
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:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/48/52 ms
ISP#ping
Protocol [ip]:
Target IP address: 172.16.1.1
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 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/48/48 ms
ISP#ping
Protocol [ip]:
Target IP address: 172.16.32.1
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 172.16.32.1, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/33/36 ms
ISP#ping
Protocol [ip]:
Target IP address: 172.16.1.2
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 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/36/56 ms
```

Complete reachability was proven between the ISP router and both SanJose1 and SanJose2.

Why do the following ping requests fail?

```
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#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)
```

Step 9: BGP Next-Hop_Self

Before the ISP can successfully ping the internal serial interfaces of AS 64512, two issues need to be resolved. First, SanJose1 does not know about the link between the ISP and SanJose2. Second, SanJose2 is unaware of the link between the ISP and SanJose1. This can be resolved by an advertisement of these serial links via BGP on the ISP router. This can also be resolved via EIGRP on each of the SanJose routers. The preferred method is for the ISP to advertise these links. If they are advertised and then, at a future date, a BGP link is activated to another ISP in addition to ISP at AS 200, there is a risk of becoming a Transit AS.

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
```

Clear the IP BGP conversation with the **clear ip bgp *** command on ISP. Wait for the conversations to reestablish with each SanJose router. Issue the **show ip bgp** command to verify that the ISP router can see its own WAN links through 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 Origin codes: i - IGP, e - EGP, ? - incomplete

 Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.6                            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
```

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

```
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.32.0 is directly connected, Loopback0
C       172.16.1.0 is directly connected, Serial0/0/1
D       172.16.64.0 [90/20640000] via 172.16.1.1, 00:57:10, Serial0/0/1
     192.168.1.0/30 is subnetted, 2 subnets
C       192.168.1.0 is directly connected, Serial0/0/0
B       192.168.1.4 [20/0] via 192.168.1.1, 00:04:23
B    192.168.100.0/24 [20/0] via 192.168.1.1, 00:04:23
```

The next issue to consider is BGP policy routing between autonomous systems. BGP routers do not increment the next-hop address to their IBGP peers. 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: forwarding 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 only achieved if the **next-hop-self** command is configured on SanJose1 and SanJose2.

This is the output before the **next-hop-self** command was issued:

```
SanJose2#show ip bgp
BGP table version is 11, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i192.168.1.0/30   192.168.1.5              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.1.4/30   192.168.1.5              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.100.0    192.168.1.5              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i

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
```

After issuing these commands, reset BGP operation on either router with the **clear ip bgp *** command.

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

```
SanJose2#show ip bgp
BGP table version is 11, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i192.168.1.0/30   172.16.64.1              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.1.4/30   172.16.64.1              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.100.0    172.16.64.1              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
```

Step 10: 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.

Since 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.

Issue the following commands on SanJose1 and 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
SanJose2(config-route-map)#router bgp 64512
SanJose2(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
```

Do not forget to use the **clear ip bgp *** command after configuring this new policy. Once the conversations have been reestablished, issue the **show ip bgp** command on SanJose1 and SanJose2:

```
SanJose1#show ip bgp

BGP table version is 8, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*>i172.16.0.0       172.16.32.1              0    100      0 i
*> 192.168.1.0/30   192.168.1.5              0    150      0 200 i
*> 192.168.1.4/30   192.168.1.5              0    150      0 200 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i

SanJose2#show ip bgp

BGP table version is 11, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
*>i192.168.1.0/30   172.16.64.1              0    150      0 200 i
*                   192.168.1.1              0    125      0 200 i
*>i192.168.1.4/30   172.16.64.1              0    150      0 200 i
*                   192.168.1.1              0    125      0 200 i
*>i192.168.100.0    172.16.64.1              0    150      0 200 i
*                   192.168.1.1              0    125      0 200 i
```

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.

Step 11: Set BGP MED

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

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

Use an extended **ping** in this situation. Compare your output to the following:

```
SanJose2#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]: 2
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
```

```
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 (48 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 (48 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
```

If the **record** option has not been used prior to this, the important thing to note is that each of the IP addresses 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, 172.16.1.2. It then arrives at the S0/1 interface for SanJose1.
2. SanJose1 S0/0, 192.168.1.6, routes the packet out to arrive at the S0/0 interface of ISP.
3. The target of 192.168.100.1 is reached: 192.168.100.1.
4. The packet is next forwarded out the S0/1, 192.168.1.1, interface for ISP and arrives at the S0/1 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 prefers the link from SanJose2 for all return traffic.

The next step is to create a new policy to force router ISP 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
```

As before, do not forget to use the **clear ip bgp *** 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#show ip bgp
BGP table version is 10, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i172.16.0.0       172.16.32.1              0    100      0 i
*> 192.168.1.0/30   192.168.1.5              0    150      0 200 i
*> 192.168.1.4/30   192.168.1.5              0    150      0 200 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
```

Now reissue an extended **ping** with a **record** command:

```
SanJose2#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]: 2
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
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 (64 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (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 (64 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (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
```

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

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

```
BGP table version is 12, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal   Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop                 Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.2                  75             0 64512 i
*>                  192.168.1.6                  50             0 64512 i
*> 192.168.1.0/30   0.0.0.0                       0         32768 i
*> 192.168.1.4/30   0.0.0.0                       0         32768 i
*> 192.168.100.0    0.0.0.0                       0         32768 i
```

Step 12: Establish a Default Network

The final step is to establish a default route that uses a policy statement that adjusts to changes in the network. Configure both SanJose1 and SanJose2 to use the 192.168.100.0 /24 network as the default network. The following output includes the routing table before the command was issued, the actual command syntax, and then the routing table after the command was issued. Do the same on the SanJose2 router.

```
SanJose1#show ip route     ****Note: Prior to Default-Network Statement
Gateway of last resort is not set
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.32.0/24 [90/20640000] via 172.16.1.2, 02:43:46, Serial0/1
B       172.16.0.0/16 [200/0] via 172.16.32.1, 00:12:32
C       172.16.1.0/24 is directly connected, Serial0/1
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 2 subnets
B       192.168.1.0 [20/0] via 192.168.1.5, 00:14:05
C       192.168.1.4 is directly connected, Serial0/0
B    192.168.100.0/24 [20/0] via 192.168.1.5, 00:14:05

SanJose1(config)#ip default-network 192.168.100.0
SanJose1#show ip route
Gateway of last resort is 192.168.1.5 to network 192.168.100.0
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.32.0/24 [90/20640000] via 172.16.1.2, 02:44:09, Serial0/1
B       172.16.0.0/16 [200/0] via 172.16.32.1, 00:12:55
C       172.16.1.0/24 is directly connected, Serial0/1
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 2 subnets
B       192.168.1.0 [20/0] via 192.168.1.5, 00:14:28
C       192.168.1.4 is directly connected, Serial0/0
B*   192.168.100.0/24 [20/0] via 192.168.1.5, 00:14:29
```

What would be required to add a future T3 link on SanJose2 and for it to have preference for incoming and outgoing traffic?

A newly added route would be as easy as adding another route map for local preference with a value of 175 and a route map referencing a MED (metric) value of 35. Issue the **clear ip bgp *** command to complete the lab is.

Appendix A: TCL Verification

```
tclsh

foreach address {
192.168.100.1
172.16.64.1
172.16.32.1
192.168.1.1
192.168.1.2
192.168.1.5
192.168.1.6
172.16.1.1
172.16.1.2
} {
ping $address }


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

```
+>ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
ISP(tcl)#tclquit

SanJose1#tclsh
SanJose1(tcl)#
SanJose1(tcl)#foreach address {
+>192.168.100.1
+>172.16.64.1
+>172.16.32.1
+>192.168.1.1
+>192.168.1.2
+>192.168.1.5
+>192.168.1.6
+>172.16.1.1
+>172.16.1.2
+>} {
+>ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/43/44 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/68 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
SanJose1(tcl)#tclquit

SanJose2#tclsh
SanJose2(tcl)#
SanJose2(tcl)#foreach address {
+>192.168.100.1
+>172.16.64.1
+>172.16.32.1
+>192.168.1.1
+>192.168.1.2
+>192.168.1.5
+>192.168.1.6
+>172.16.1.1
+>172.16.1.2
+>} {
+>ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/36 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms
SanJose2(tcl)#tclquit
```

Final Configurations

```
ISP#show run
!
hostname ISP
!
interface Loopback0
 ip address 192.168.100.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.5 255.255.255.252
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.1.1 255.255.255.252
 no shutdown
!
router bgp 200
 no synchronization
 network 192.168.1.0 mask 255.255.255.252
 network 192.168.1.4 mask 255.255.255.252
 network 192.168.100.0
 neighbor 192.168.1.2 remote-as 64512
 neighbor 192.168.1.6 remote-as 64512
 no auto-summary
!
end

SanJose1#show run
!
hostname SanJose1
!
interface Loopback0
 ip address 172.16.64.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.6 255.255.255.252
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.1.1 255.255.255.0
 clock rate 64000
 no shutdown
!
router eigrp 64512
 network 172.16.0.0
 no auto-summary
!
router bgp 64512
 no synchronization
 network 172.16.0.0
 neighbor 172.16.32.1 remote-as 64512
 neighbor 172.16.32.1 update-source Loopback0
 neighbor 172.16.32.1 next-hop-self
 neighbor 192.168.1.5 remote-as 200
 neighbor 192.168.1.5 route-map PRIMARY_T1_IN in
 neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out
 no auto-summary
!
ip default-network 192.168.100.0
ip route 172.16.0.0 255.255.0.0 Null0
!
route-map PRIMARY_T1_IN permit 10
 set local-preference 150
!
route-map PRIMARY_T1_MED_OUT permit 10
 set metric 50
!
end

SanJose2#show run
hostname SanJose2
!
interface Loopback0
 ip address 172.16.32.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.2 255.255.255.252
 clock rate 64000
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.1.2 255.255.255.0
 no shutdown
!
router eigrp 64512
 network 172.16.0.0
 no auto-summary
!
router bgp 64512
```

```
 no synchronization
 network 172.16.0.0
 neighbor 172.16.64.1 remote-as 64512
 neighbor 172.16.64.1 update-source Loopback0
 neighbor 172.16.64.1 next-hop-self
 neighbor 192.168.1.1 remote-as 200
 neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
 neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
 no auto-summary
!
ip default-network 192.168.100.0
ip route 172.16.0.0 255.255.0.0 Null0
!
route-map SECONDARY_T1_IN permit 10
 set local-preference 125
!
route-map SECONDARY_T1_OUT permit 10
 set metric 75
!
end
```