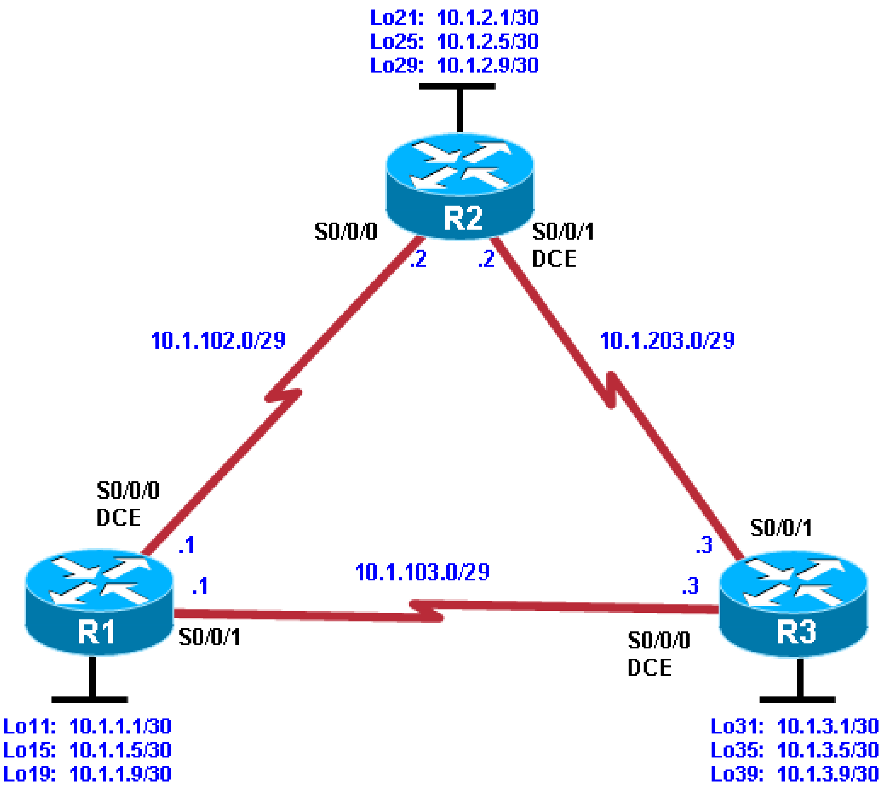
EIGRP Configuration, Bandwidth and Adjacencies

The lab is built on the topology:



Objectives

- Review a basic EIGRP configuration.
- Explore the EIGRP topology table.
- Identify successors, feasible successors, and feasible distances.
- Use **show** and **debug** commands for the EIGRP topology table.
- Configure and verify equal-cost load balancing with EIGRP.
- Configure and verify unequal-cost load balancing with EIGRP.

Background

As a senior network engineer, you are considering deploying EIGRP in your corporation and want to evaluate its ability to converge quickly in a changing environment. You are also interested in equal-cost and unequal-cost load balancing because your network contains redundant links. These links are not often used by other link-state routing protocols because of high metrics. Because you are interested in testing the EIGRP claims that you have read about, you decide to implement and test on a set of three lab routers before deploying EIGRP throughout your corporate network.

Instructions:

Step 1: Configure the addressing and serial links.

a. Create three loopback interfaces on each router and address them as 10.1.*X*.1/30, 10.1.*X*.5/30, and 10.1.*X*.9/30, where *X* is the number of the router. Use the following table or the initial configurations located at the end of the lab.

Router	Interface	IP Address/Mask
R1	Loopback11	10.1.1.1/30
R1	Loopback15	10.1.1.5/30
R1	Loopback19	10.1.1.9/30
R2	Loopback21	10.1.2.1/30
R2	Loopback25	10.1.2.5/30
R2	Loopback29	10.1.2.9/30
R3	Loopback31	10.1.3.1/30
R3	Loopback35	10.1.3.5/30
R3	Loopback39	10.1.3.9/30

```
R1(config)# interface Loopback 11
R1(config-if)# ip address 10.1.1.1 255.255.255.252
R1(config-if)# exit
R1(config)# interface Loopback 15
R1(config-if)# ip address 10.1.1.5 255.255.255.252
R1(config-if)# exit
R1(config)# interface Loopback 19
R1(config-if)# ip address 10.1.1.9 255.255.255.252
R1(config-if)# exit

R2(config)# interface Loopback 21
R2(config-if)# ip address 10.1.2.1 255.255.255.252
R2(config-if)# exit
R2(config)# interface Loopback 25
R2(config-if)# ip address 10.1.2.5 255.255.255.252
R2(config-if)# exit
R2(config)# interface Loopback 29
R2(config-if)# ip address 10.1.2.9 255.255.255.252
R2(config-if)# exit

R3(config)# interface Loopback 31
R3(config-if)# ip address 10.1.3.1 255.255.255.252
R3(config-if)# exit
R3(config)# interface Loopback 35
R3(config-if)# ip address 10.1.3.5 255.255.255.252
R3(config-if)# exit
R3(config)# interface Loopback 39
R3(config-if)# ip address 10.1.3.9 255.255.255.252
R3(config-if)# exit
```

b. Specify the addresses of the serial interfaces as shown in the topology diagram. Set the clock rate to 64 kb/s, and manually configure the interface bandwidth to 64 kb/s.

```
R1(config)# interface Serial 0/0/0
R1(config-if)# description R1-->R2
R1(config-if)# clock rate 64000
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.102.1 255.255.255.248
R1(config-if)# no shutdown
R1(config-if)# exit
```

```
R1(config)# interface Serial 0/0/1
R1(config-if)# description R1-->R3
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.103.1 255.255.255.248
R1(config-if)# no shutdown
R1(config-if)# exit

R2(config)# interface Serial 0/0/0
R2(config-if)# description R2-->R1
R2(config-if)# bandwidth 64
R2(config-if)# ip address 10.1.102.2 255.255.255.248
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# interface Serial 0/0/1
R2(config-if)# description R2-->R3
R2(config-if)# clock rate 64000
R2(config-if)# bandwidth 64
R2(config-if)# ip address 10.1.203.2 255.255.255.248
R2(config-if)# no shutdown
R2(config-if)# exit

R3(config)# interface Serial 0/0/0
R3(config-if)# description R3-->R1
R3(config-if)# clock rate 64000
R3(config-if)# bandwidth 64
R3(config-if)# ip address 10.1.103.3 255.255.255.248
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# interface Serial 0/0/1
R3(config-if)# description R3-->R2
R3(config-if)# bandwidth 64
R3(config-if)# ip address 10.1.203.3 255.255.255.248
R3(config-if)# no shutdown
R3(config-if)# exit
```

c. Verify connectivity by pinging across each of the local networks connected to each router.

d. Issue the **show interfaces description** command on each router. This command displays a brief listing of the interfaces, their status, and a description (if a description is configured). Router R1 is shown as an example.

```
R1# show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Fa0/1                          admin down     down
Se0/0/0                        up             up       R1-->R2
Se0/0/1                        up             up       R1-->R3
Vl1                            up             down
Lo11                           up             up
Lo15                           up             up
Lo19                           up             up
```

e. Issue the **show protocols** command on each router. This command displays a brief listing of the interfaces, their status, and the IP address and subnet mask configured (in prefix format /xx) for each interface. Router R1 is shown as an example.

```
R1# show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
FastEthernet0/1 is administratively down, line protocol is down
Serial0/0/0 is up, line protocol is up
  Internet address is 10.1.102.1/29
Serial0/0/1 is up, line protocol is up
  Internet address is 10.1.103.1/29
Vlan1 is up, line protocol is down
Loopback11 is up, line protocol is up
  Internet address is 10.1.1.1/30
Loopback15 is up, line protocol is up
  Internet address is 10.1.1.5/30
Loopback19 is up, line protocol is up
  Internet address is 10.1.1.9/30
```

Step 2: Configure EIGRP.

a. Enable EIGRP AS 100 for all interfaces on R1 and R2 using the commands used in the previous EIGRP lab. Do not enable EIGRP yet on R3. For your reference, these are the commands which can be used:

```
R1(config)# router eigrp 100
R1(config-router)# network 10.0.0.0

R2(config)# router eigrp 100
R2(config-router)# network 10.0.0.0
```

b. Use the **debug ip eigrp 100** command to watch EIGRP install the routes in the routing table when your routers become adjacent. You get output similar to the following.

```
R3# debug ip eigrp 100
IP-EIGRP Route Events debugging is on
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.

R3(config)# router eigrp 100
```

```
R3(config-router)# network 10.0.0.0

R3(config-router)#
*Feb  4 18:44:57.367: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.103.1
(Serial0/0/0) is up: new adjacency
*Feb  4 18:44:57.367: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.203.2
(Serial0/0/1) is up: new adjacency
*Feb  4 18:44:57.371: IP-EIGRP(Default-IP-Routing-Table:100): Processing
incoming UPDATE packet
*Feb  4 18:44:57.379: IP-EIGRP(Default-IP-Routing-Table:100): Processing
incoming UPDATE packet
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Processing
incoming UPDATE packet
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Int
10.1.102.0/29 M 41024000 - 40000000 1024000 SM 40512000 - 40000000 512000
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.102.0  ()
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Int 10.1.1.0/30
M40640000 - 40000000 640000 SM 128256 - 256 128000
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.0  ()
```

```
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Int 10.1.1.4/30
M 40640000 - 40000000 640000 SM 128256 - 256 128000
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.4  ()
*Feb  4 18:44:57.431: IP-EIGRP(Default-IP-Routing-Table:100): Int 10.1.1.8/30
M40640000 - 40000000 640000 SM 128256 - 256 128000
*Feb  4 18:44:57.431: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.8  ()

<output omitted>
```

Essentially, the EIGRP DUAL state machine has just computed the topology table for these routes and installed them in the routing table.

c. Check to see that these routes exist in the routing table with the **show ip route** command.

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

Gateway of last resort is not set

   10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
D      10.1.3.8/30 [90/40640000] via 10.1.103.3, 00:19:28, Serial0/0/1
D      10.1.2.8/30 [90/40640000] via 10.1.102.2, 00:21:59, Serial0/0/0
C      10.1.1.8/30 is directly connected, Loopback19
D      10.1.3.0/30 [90/40640000] via 10.1.103.3, 00:19:28, Serial0/0/1
D      10.1.2.0/30 [90/40640000] via 10.1.102.2, 00:21:59, Serial0/0/0
C      10.1.1.0/30 is directly connected, Loopback11
D      10.1.3.4/30 [90/40640000] via 10.1.103.3, 00:19:28, Serial0/0/1
D      10.1.2.4/30 [90/40640000] via 10.1.102.2, 00:21:59, Serial0/0/0
C      10.1.1.4/30 is directly connected, Loopback15

C     10.1.103.0/29 is directly connected, Serial0/0/1
C     10.1.102.0/29 is directly connected, Serial0/0/0
D     10.1.203.0/29 [90/41024000] via 10.1.103.3, 00:19:28, Serial0/0/1
                    [90/41024000] via 10.1.102.2, 00:19:28, Serial0/0/0
```

d. After you have full adjacency between the routers, ping all the remote loopbacks to ensure full
You should receive ICMP echo replies for each address pinged. Make sure that you run the Tcl script on each router and verify connectivity before you continue with the lab.

e. Verify the EIGRP neighbor relationships with the **show ip eigrp neighbors** command.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.102.2              Se0/0/0           10 00:00:22    1  5000  2  0
1   10.1.103.3              Se0/0/1           13 00:04:36   24  2280  0  14

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.102.1              Se0/0/0           14 00:00:37    1  5000  1  22
1   10.1.203.3              Se0/0/1           11 00:03:29  143  2280  0  15
```

```
R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H    Address                 Interface       Hold Uptime    SRTT   RTO  Q   Seq
                                             (sec)          (ms)        Cnt Num
1    10.1.203.2              Se0/0/1           14 00:03:43  241   2280  0   18
0    10.1.103.1              Se0/0/0           14 00:05:05   38   2280  0   17
```

Step 3: Examine the EIGRP topology table.

a. EIGRP builds a topology table containing all successor routes. The course content covered the vocabulary for EIGRP routes in the topology table. What is the feasible distance of route 10.1.1.0/30 in the R3 topology table in the following output?

____

____

```
R3# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(10.1.3.9)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.8/30, 1 successors, FD is 128256
    via Connected, Loopback39
P 10.1.2.8/30, 1 successors, FD is 40640000
    via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.8/30, 1 successors, FD is 40640000
    via 10.1.103.1 (40640000/128256), Serial0/0/0
P 10.1.3.0/30, 1 successors, FD is 128256
    via Connected, Loopback31
P 10.1.2.0/30, 1 successors, FD is 40640000
    via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.0/30, 1 successors, FD is 40640000
    via 10.1.103.1 (40640000/128256), Serial0/0/0
P 10.1.3.4/30, 1 successors, FD is 128256
    via Connected, Loopback35
P 10.1.2.4/30, 1 successors, FD is 40640000
    via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.4/30, 1 successors, FD is 40640000
    via 10.1.103.1 (40640000/128256), Serial0/0/0
P 10.1.103.0/29, 1 successors, FD is 40512000
    via Connected, Serial0/0/0
P 10.1.102.0/29, 2 successors, FD is 41024000
    via 10.1.103.1 (41024000/40512000), Serial0/0/0
    via 10.1.203.2 (41024000/40512000), Serial0/0/1
P 10.1.203.0/29, 1 successors, FD is 40512000
    via Connected, Serial0/0/1
```

b. The most important thing is the two successor routes in the passive state on R3. R1 and R2 are both advertising their connected subnet of 10.1.102.0/30. Because both routes have the same feasible distance of 41024000, both are installed in the topology table. This distance of 41024000 reflects the composite metric of more granular properties about the path to the destination network. Can you view the metrics before the composite metric is computed?

c. Use the **show ip eigrp topology 10.1.102.0/29** command to view the information that EIGRP has received about the route from R1 and R2.

```
R3# show ip eigrp topology 10.1.102.0/29
IP-EIGRP (AS 100): Topology entry for 10.1.102.0/29
  State is Passive, Query origin flag is 1, 2 Successor(s), FD is 41024000
  Routing Descriptor Blocks:
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
      Composite metric is (41024000/40512000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  10.1.203.2 (Serial0/0/1), from 10.1.203.2, Send flag is 0x0
      Composite metric is (41024000/40512000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

The output of this command shows the following information regarding EIGRP:

- The bandwidth metric represents the *minimum* bandwidth among all links comprising the path to the destination network.
- The delay metric represents the *total* delay over the path.
- The minimum MTU represents the smallest MTU along the path.
- If you do not have full knowledge of your network, you can use the hop count information to check how many Layer 3 devices are between the router and the destination network.

Step 4: Observe equal-cost load balancing.

EIGRP produces equal-cost load balancing to the destination network 10.1.102.0/29 from R1. Two equal-cost paths are available to this destination per the topology table output above.

a. Use the **traceroute 10.1.102.1** command to view the hops from R3 to this R1 IP address. Notice that both R1 and R2 are listed as hops because there are two equal-cost paths and packets can reach this network via either link.

```
R3# traceroute 10.1.102.1

Type escape sequence to abort.
Tracing the route to 10.1.102.1

  1 10.1.203.2 12 msec
    10.1.103.1 12 msec
    10.1.203.2 12 msec
```

Recent Cisco IOS releases enable Cisco Express Forwarding (CEF), which, by default, performs per-destination load balancing. CEF allows for very rapid switching without the need for route processing. However, if you were to ping the destination network, you would not see load balancing occurring on a packet level because CEF treats the entire series of pings as one flow.

CEF on R3 overrides the per-packet balancing behavior of process switching with per-destination load balancing.

b. To see the full effect of EIGRP equal-cost load balancing, temporarily disable CEF and route caching so that all IP packets are processed individually and not fast-switched by CEF.

```
R3(config)# no ip cef

R3(config)# interface S0/0/0
R3(config-if)# no ip route-cache
R3(config-if)# interface S0/0/1
R3(config-if)# no ip route-cache
```

Note: Typically, you would not disable CEF in a production network. It is done here only to illustrate load balancing. Another way to demonstrate per-packet load balancing, that does not disable CEF, is to use the per-packet load balancing command **ip load-share per-packet** on outgoing interfaces S0/0/0 and S0/0/1.

c. Verify load balancing with the **debug ip packet** command, and then ping 10.1.102.1. You see output similar to the following:

```
R3# debug ip packet
IP packet debugging is on

R3# ping 10.1.102.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.102.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
R3#
*Feb  5 12:58:27.943: IP: tableid=0, s=10.1.103.3 (local), d=10.1.102.1
(Serial0/0/0), routed via RIB
*Feb  5 12:58:27.943: IP: s=10.1.103.3 (local), d=10.1.102.1 (Serial0/0/0),
len 100, sending
*Feb  5 12:58:27.947: IP: tableid=0, s=10.1.102.1 (Serial0/0/0), d=10.1.103.3
(Serial0/0/0), routed via RIB
*Feb  5 12:58:27.947: IP: s=10.1.102.1 (Serial0/0/0), d=10.1.103.3
(Serial0/0/0), len 100, rcvd 3
*Feb  5 12:58:27.947: IP: tableid=0, s=10.1.203.3 (local), d=10.1.102.1
(Serial0/0/1), routed via RIB
*Feb  5 12:58:27.947: IP: s=10.1.203.3 (local), d=10.1.102.1 (Serial0/0/1),
len 100, sending

<output omitted>
```

Notice that EIGRP load-balances between Serial0/0/0 (s=10.1.103.3) and Serial0/0/1 (s=10.1.203.3). This behavior is part of EIGRP. It can help utilize underused links in a network, especially during periods of congestion.

Step 5: Analyze alternate EIGRP paths not in the topology table.

b. Issue the **show ip eigrp topology all-links** command to see all routes that R3 has learned through EIGRP. This command shows all entries that EIGRP holds on this router for networks in the topology, including the exit serial interface and IP address of the next hop to each destination network, and the serial number (serno) that uniquely identifies a destination network in EIGRP.

```
R3# show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(100)/ID(10.1.3.9)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/30, 1 successors, FD is 128256, serno 1
        via Connected, Loopback31
P 10.1.3.4/30, 1 successors, FD is 128256, serno 2
        via Connected, Loopback35
P 10.1.3.8/30, 1 successors, FD is 128256, serno 3
        via Connected, Loopback39
P 10.1.2.8/30, 1 successors, FD is 40640000, serno 24
        via 10.1.203.2 (40640000/128256), Serial0/0/1
        via 10.1.103.1 (41152000/40640000), Serial0/0/0
P 10.1.1.8/30, 1 successors, FD is 40640000, serno 17
        via 10.1.103.1 (40640000/128256), Serial0/0/0
        via 10.1.203.2 (41152000/40640000), Serial0/0/1
P 10.1.2.0/30, 1 successors, FD is 40640000, serno 22
        via 10.1.203.2 (40640000/128256), Serial0/0/1
        via 10.1.103.1 (41152000/40640000), Serial0/0/0
P 10.1.1.0/30, 1 successors, FD is 40640000, serno 15
        via 10.1.103.1 (40640000/128256), Serial0/0/0
        via 10.1.203.2 (41152000/40640000), Serial0/0/1
P 10.1.2.4/30, 1 successors, FD is 40640000, serno 23
        via 10.1.203.2 (40640000/128256), Serial0/0/1
        via 10.1.103.1 (41152000/40640000), Serial0/0/0
P 10.1.1.4/30, 1 successors, FD is 40640000, serno 16
        via 10.1.103.1 (40640000/128256), Serial0/0/0
        via 10.1.203.2 (41152000/40640000), Serial0/0/1
P 10.1.103.0/29, 1 successors, FD is 40512000, serno 13
        via Connected, Serial0/0/0
P 10.1.102.0/29, 2 successors, FD is 41024000, serno 42
        via 10.1.103.1 (41024000/40512000), Serial0/0/0

        via 10.1.203.2 (41024000/40512000), Serial0/0/1
P 10.1.203.0/29, 1 successors, FD is 40512000, serno 12
        via Connected, Serial0/0/1
```

What is the advertised distance of the R1 loopback network routes from R1 and R2?

____

c. Use the **show ip eigrp topology 10.1.2.0/30** command to see the granular view of the alternate paths to 10.1.2.0, including ones with a higher reported distance than the feasible distance.

```
R3# show ip eigrp topology 10.1.2.0/30
IP-EIGRP (AS 100): Topology entry for 10.1.2.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 40640000
  Routing Descriptor Blocks:
```

```
 10.1.203.2 (Serial0/0/1), from 10.1.203.2, 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
 10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
     Composite metric is (41152000/40640000), Route is Internal
     Vector metric:
       Minimum bandwidth is 64 Kbit
       Total delay is 45000 microseconds
       Reliability is 255/255
       Load is 1/255
       Minimum MTU is 1500
       Hop count is 2
```

d. Start a ping with a high repeat count on R3 to the R1 Serial0/0/0 interface 10.1.102.1.

```
R3# ping 10.1.102.1 repeat 10000
```

e. Enter interface configuration mode on R1 and shut down port Serial0/0/1, which is the direct link from R1 to R3.

```
R1(config)# interface serial 0/0/1
R1(config-if)# shutdown
```

f. When the adjacency between R1 and R3 goes down, some pings will be lost. After pings are again being successfully received, stop the ping using Ctrl+Shift+^.

```
R3# ping 10.1.102.1 repeat 10000

Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 10.1.102.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!.
*Dec 11 18:41:55.843: %LINK-3-UPDOWN: Interface Serial0/0/0, changed state to
down
*Dec 11 18:41:55.847: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.103.1
(Serial0/0/0) is down: interface down
*Dec 11 18:41:56.843: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/0, changed state to down
.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 99 percent (374/376), round-trip min/avg/max = 28/39/96 ms
R3#
```

How many packets were dropped?

____

Step 6: Observe unequal-cost load balancing.

a. Review the composite metrics advertised by EIGRP using the **show ip eigrp topology 10.1.2.0/30** command,.

```
R3# show ip eigrp topology 10.1.2.0/30
IP-EIGRP (AS 100): Topology entry for 10.1.2.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 40640000
  Routing Descriptor Blocks:
```

```
  10.1.203.2 (Serial0/0/1), from 10.1.203.2, 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
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
      Composite metric is (41152000/40640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

The reported distance for a loopback network is higher than the feasible distance, so DUAL does not consider it a feasible successor route.

b. To demonstrate unequal-cost load balancing in your internetwork, upgrade the path to the destination network through R1 with a higher bandwidth. Change the clock rate and bandwidth on the R1, R2, and R3 serial interfaces to 128 kb/s.

```
R1(config)# interface serial 0/0/0
R1(config-if)# bandwidth 128
R1(config-if)# clock rate 128000
R1(config-if)# interface serial 0/0/1
R1(config-if)# bandwidth 128

R2(config)# interface serial 0/0/0
R2(config-if)# bandwidth 128


R3(config)# interface serial 0/0/0
R3(config-if)# clock rate 128000
R3(config-if)# bandwidth 128
```

c. Issue the **show ip eigrp topology 10.1.2.0/30** command again on R3 to see what has changed.

```
R3# show ip eigrp topology 10.1.2.0/30
IP-EIGRP (AS 100): Topology entry for 10.1.2.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000
  Routing Descriptor Blocks:
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, 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

10.1.203.2 (Serial0/0/1), from 10.1.203.2, 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
```

After manipulating the bandwidth parameter, the preferred path for R3 to the loopback interfaces of R2 is now through R1. Even though the hop count is two and the delay through R1 is nearly twice that of the R2 path, the higher bandwidth and lower FD results in this being the preferred route.

d. Issue the **show ip route** command to verify that the preferred route to network 10.1.2.0 is through R1 via Serial0/0/0 to next hop 10.1.103.1. There is only one route to this network due to the difference in bandwidth.

```
R3# show ip route eigrp
     10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
D       10.1.2.8/30 [90/21152000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.1.8/30 [90/20640000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.2.0/30 [90/21152000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.1.0/30 [90/20640000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.2.4/30 [90/21152000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.1.4/30 [90/20640000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.102.0/29 [90/21024000] via 10.1.103.1, 00:16:52, Serial0/0/0
```

e. Issue the **debug ip eigrp 100** command on R3 to show route events changing in real time. Then, under the EIGRP router configuration on R3, issue the **variance 2** command, which allows unequal-cost load balancing bounded by a maximum distance of (2) × (FD), where FD represents the feasible distance for each route in the routing table.

```
R3# debug ip eigrp 100
IP-EIGRP Route Events debugging is on

R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
R3(config)# router eigrp 100
R3(config-router)# variance 2
R3(config-router)#
*Feb  5 15:11:45.195: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.3.8/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.195: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.8  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.8  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.8  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.1.8/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.3.0/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.0  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.0  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.0  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.1.0/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.3.4/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.4  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.4  ()
```

```
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.4  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.1.4/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.103.0/29
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.102.0  ()
*Feb  5 15:11:45.203: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.102.0  ()
```

f. Issue the **show ip route** command again to verify that there are now two routes to network 10.1.2.0.

```
R3# show ip route eigrp

     10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
D       10.1.2.8/30 [90/40640000] via 10.1.203.2, 00:02:27, Serial0/0/1
                    [90/21152000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.1.8/30 [90/20640000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.2.0/30 [90/40640000] via 10.1.203.2, 00:02:27, Serial0/0/1
                    [90/21152000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.1.0/30 [90/20640000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.2.4/30 [90/40640000] via 10.1.203.2, 00:02:27, Serial0/0/1
                    [90/21152000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.1.4/30 [90/20640000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.102.0/29 [90/41024000] via 10.1.203.2, 00:02:27, Serial0/0/1
                      [90/21024000] via 10.1.103.1, 00:02:27, Serial0/0/0
```

g. These unequal-cost routes also show up in the EIGRP topology table, even though they are not considered feasible successor routes. Use the **show ip eigrp topology** command to verify this.

```
R3# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(10.1.3.9)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.8/30, 1 successors, FD is 128256
        via Connected, Loopback39
P 10.1.2.8/30, 1 successors, FD is 21152000
        via 10.1.103.1 (21152000/20640000), Serial0/0/0
        via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.8/30, 1 successors, FD is 20640000
        via 10.1.103.1 (20640000/128256), Serial0/0/0
P 10.1.3.0/30, 1 successors, FD is 128256
        via Connected, Loopback31
P 10.1.2.0/30, 1 successors, FD is 21152000
        via 10.1.103.1 (21152000/20640000), Serial0/0/0
        via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.0/30, 1 successors, FD is 20640000
        via 10.1.103.1 (20640000/128256), Serial0/0/0
P 10.1.3.4/30, 1 successors, FD is 128256
        via Connected, Loopback35
P 10.1.2.4/30, 1 successors, FD is 21152000
        via 10.1.103.1 (21152000/20640000), Serial0/0/0
        via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.4/30, 1 successors, FD is 20640000
        via 10.1.103.1 (20640000/128256), Serial0/0/0
P 10.1.103.0/29, 1 successors, FD is 20512000
        via Connected, Serial0/0/0
P 10.1.102.0/29, 1 successors, FD is 21024000
        via 10.1.103.1 (21024000/20512000), Serial0/0/0
        via 10.1.203.2 (41024000/20512000), Serial0/0/1
P 10.1.203.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
```

h. Load balancing over serial links occurs in blocks of packets, the number of which are recorded in the routing table's detailed routing information. Use the **show ip route 10.1.2.0** command to get a detailed view of how traffic is shared between the two links.

```
R3# show ip route 10.1.2.0
Routing entry for 10.1.2.0/30
  Known via "eigrp 100", distance 90, metric 21152000, type internal
  Redistributing via eigrp 100
  Last update from 10.1.203.2 on Serial0/0/1, 00:05:41 ago
  Routing Descriptor Blocks:
    10.1.203.2, from 10.1.203.2, 00:05:41 ago, via Serial0/0/1
      Route metric is 40640000, traffic share count is 25
      Total delay is 25000 microseconds, minimum bandwidth is 64 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 10.1.103.1, from 10.1.103.1, 00:05:41 ago, via Serial0/0/0
      Route metric is 21152000, traffic share count is 48
      Total delay is 45000 microseconds, minimum bandwidth is 128 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

i. Check the actual load balancing using the **debug ip packet** command. Ping from R3 to 10.1.2.1 with a high enough repeat count to view the load balancing over both paths. In the case above, the traffic share is 25 packets routed to R2 to every 48 packets routed to R1.

j. To filter the debug output to make it more useful, use the following extended access list.

```
R3(config)# access-list 100 permit icmp any any echo
R3(config)# end

R3# debug ip packet 100
IP packet debugging is on for access list 100

R3# ping 10.1.2.1 repeat 50

Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*Feb  5 15:20:54.215: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.215: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.231: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.231: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.247: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.247: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.263: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.263: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.279: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.279: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.295: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
*Feb  5 15:20:54.295: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.311: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.311: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
!
<output omitted until the switch to the other path takes place>
!
*Feb  5 15:20:55.395: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
!
```

R3 just switched to load-share the outbound ICMP packets to Serial0/0/1.

```
!
*Feb  5 15:20:55.395: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending
```

```
*Feb  5 15:20:55.423: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
*Feb  5 15:20:55.423: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending
*Feb  5 15:20:55.451: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
*Feb  5 15:20:55.451: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending
*Feb  5 15:20:55.483: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
*Feb  5 15:20:55.483: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending

<output omitted>
```

Note: If a deliberate metric manipulation is necessary on a router to force it to prefer one interface over another for EIGRP-discovered routes, it is recommended to use the interface-level command "delay" for these purposes. While the "bandwidth" command can also be used to influence the metrics of EIGRP-discovered routes through a particular interface, it is discouraged because the "bandwidth" will also influence the amount of bandwidth reserved for EIGRP packets and other IOS subsystems as well. The "delay" parameter specifies the value of the interface delay that is used exclusively by EIGRP to perform metric calculations and does not influence any other area of IOS operation.