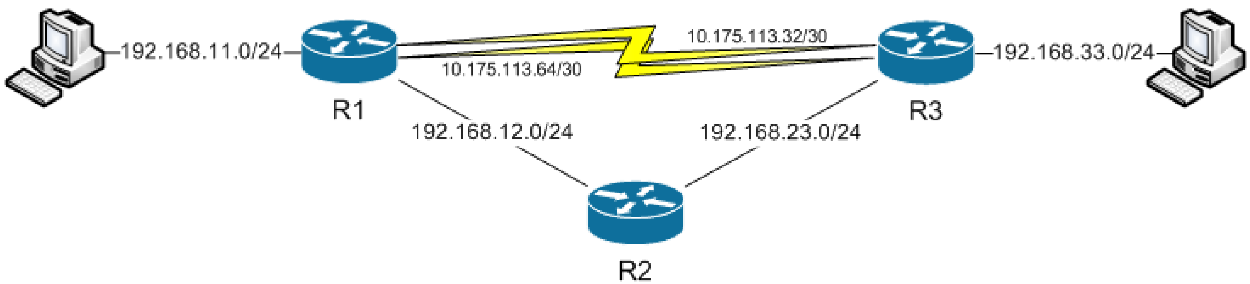
EIGRP Routing



The purpose of this lab is to explore the functionality of the EIGRP routing protocol.

Hardware & Configuration Required for this Lab

- Two Cisco routers with two Fast Ethernet interfaces and two serial ports
- One Cisco router with two Fast Ethernet interfaces
- Two crossover Cat 5 cables for router to router
- Two DTE/DCE back to back cables
- Two PCs to connect to the routers
- Two crossover Cat 5 cables for PC to Router or four straight through Cat 5 cables if you put a switch in between them
- **Special Note:** If you do not have three routers with dual Ethernet ports and two PCs, that is ok. Simply do not configure the 192.168.11.0 subnet on R1 and the 192.168.33.0 subnet on R3. Then do your pings from R1 & R3 respectively in place of the PCs and the lab will still work fine.

Commands Used in this Lab

router eigrp – Enables eigrp on the router
show ip eigrp topology - Displays the eigrp topology table and route information
clock rate – Sets clock speed on a WAN serial link
bandwidth – Logial setting of the bandwidth metric on a link
delay – Logical setting of the delay metric on a link
variance <multiplier> - Used to tell the router what multiple of the feasible distance should be considered for unequal load balancing
ip hello-interval eigrp – sets the hello time on an eigrp interface
ip hold-time eigrp – sets the hold time on an eirgrp interface
debug ip eigrp – Displays route table updates and associated messages

---

Initial Configs - Where you see *Initial Configs,* these are basic configuration steps that by now you should be able to perform on the devices by yourself without us detailing them step by step. Generally you simply go into enable and then configuration mode and start the configuration.

R1

line con 0
logging synch
exit
int loopback 0
ip add 150.123.11.11 255.255.255.255
int fa0/0
ip add 192.168.12.1 255.255.255.0
no shut
int fa0/1
ip add 192.168.11.1 255.255.255.0
no shut
int s0/0
ip add 10.175.113.33 255.255.255.252
clock rate 128000 (if you have a WIC-1DSU-TI module(don’t confuse this with a WIC-1T, use the *service-module T1 clock source internal* command instead)).
no shut
int s0/1
ip add 10.175.113.65 255.255.255.252
clock rate 128000 (if you have a WIC-1DSU-TI module(don’t confuse this with a WIC-1T, use the *service-module T1 clock source internal* command instead)).
no shut

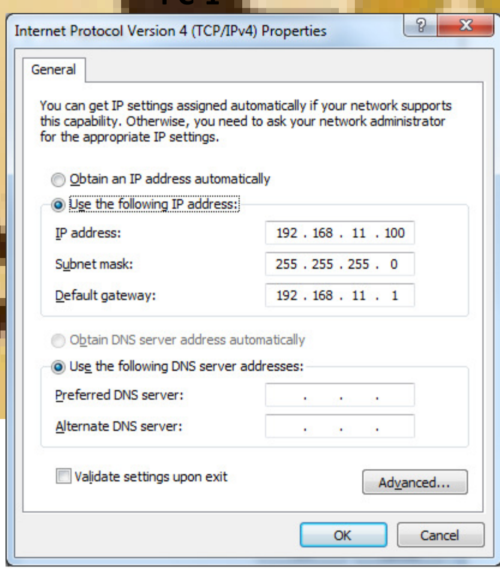
R2

line con 0
logging synch
exit
int loopback 0
ip add 150.123.22.22 255.255.255.255
int fa0/0
ip add 192.168.12.2 255.255.255.0
no shut
int fa0/1
ip add 192.168.23.2 255.255.255.0
no shut

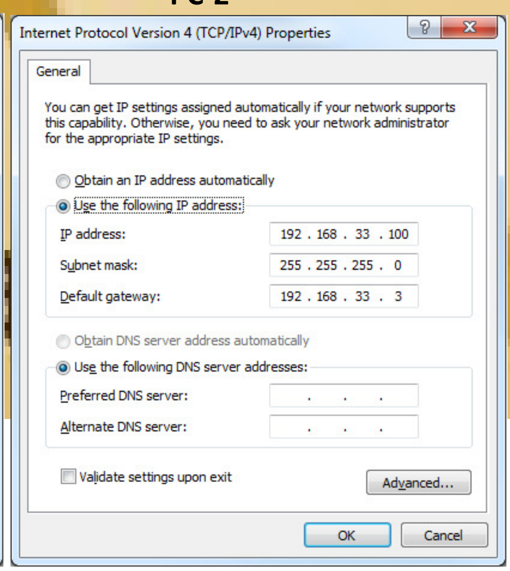
R3

line con 0
logging synch
exit
int loopback 0
ip add 150.123.33.33 255.255.255.255
int fa0/0
ip add 192.168.23.3 255.255.255.0
no shut
int fa0/1
ip add 192.168.33.1 255.255.255.0
no shut
int s0/0
ip add 10.175.113.34 255.255.255.252
(If you have a WIC-1DSU-TI module(don't confuse this with a WIC-1T, use the *service-module T1 clock source line* command here)).
no shut
int s0/1
ip add 10.175.113.66 255.255.255.252
(If you have a WIC-1DSU-TI module(don't confuse this with a WIC-1T, use the *service-module T1 clock source line* command here)).
no shut

PC-1



PC-2



EIGRP is Cisco's proprietary routing protocol designed to replace IGRP (no longer on the CCNA exam) and provide an easier alternative to OSPF. It is a hybrid protocol meaning it is a mix between distance vector (RIP) and link state protocols (OSPF, ISIS).

To enable EIGRP on a router you use it's router command **router eigrp <AS>**. EIGRP requires an Autonomous system number because that is how it determines what routers belong to the same EIGRP routing domain. For example a router with an EIGRP AS of 1 and another with an AS of 150 will not form adjacency they will simply ignore each other.

Note: AS numbers are unique to your own network, so you pick whatever number makes sense to you.

Like RIP, EIGRP is easy to configure basic functionality once again it uses the **network** statement to control what interfaces are going to run EIGRP.

The network statement uses the following syntax:
network <classful network>

The network statement is still classful so if you enter **network 10.0.0.0** any interface with an IP in the range of 10.0.0.1 - 10.255.255.254 will be added to EIGRP. You can optionally add a wildcard mask to the network statement to be more selective. **network 10.1.1.1 0.0.0.0** for example will only enable EIGRP on an interface with the IP 10.1.1.1 instead of the whole range.

Wildcard masks are used with Access-lists and several other functions as well. They are the inverse of a subnet mask excluding the network bit. In a subnet mask the 1 bit is ignored, in a wildcard the 0 bit is ignored. Here is a table showing the common Class C wildcard masks.

Subnet Mask	Size of Network	Wildcard Mask
255.255.255.0	256	0.0.0.255
255.255.255.128	128	0.0.0.127
255.255.255.192	64	0.0.0.63
255.255.255.224	32	0.0.0.31
255.255.255.240	16	0.0.0.15
255.255.255.248	8	0.0.0.7
255.255.255.252	4	0.0.0.3
255.255.255.254	2	0.0.0.1
255.255.255.255	1	0.0.0.0

Let's enable EIGRP on all of routers for all interfaces. We will use EIGRP AS 123. For now let's just use the classful statements. To prove it's using classful statements lets enter R1's 10 network as 10.10.10.0.

R1(config)#**router eigrp 123**
R1(config-router)#**network 10.10.10.0**
R1(config-router)#**network 192.168.11.0**
R1(config-router)#**network 192.168.12.0**

R1(config-router)#**network 150.123.0.0**

R2(config)#**router eigrp 123**
R2(config-router)#**network 10.0.0.0**
R2(config-router)#**network 192.168.12.0**
R2(config-router)#**network 192.168.23.0**
R2(config-router)#**network 150.123.0.0**

R3(config)#**router eigrp 123**
R3(config-router)#**network 10.0.0.0**
R3(config-router)#**network 192.168.23.0**
R3(config-router)#**network 192.168.33.0**
R3(config-router)#**network 150.123.0.0**

One thing you will notice is that EIGRP is a very fast protocol, it almost immediately forms adjacency once it's configured on both ends, you will see something like this message once adjacency forms.

*Mar 1 05:02:03.846: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123: Neighbor 10.175.113.34 (Serial0/0) is up: new adjacency
*Mar 1 05:02:03.850: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123: Neighbor 10.175.113.66 (Serial0/1) is up: new adjacency

EIGRP also can use the **show ip protocol** command to show a quick summary of EIGRP information.

R1#**show ip protocol**
Routing Protocol is "eigrp 123"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Output truncated

A handy command to see what adjacencies are up and running is: **show ip eigrp neighbors**. It will show the neighbor IP, what interface it was learned on. Troubleshooting wise the most useful column is the **Q Cnt** which is the Queue Count, basically if the number is not zero something is wrong in your network.

R1#**show ip eigrp neighbors**

```
IP-EIGRP neighbors for process 123
H   Address            Interface     Hold Uptime   SRTT   RTO  Q  Seq
                                     (sec)         (ms)        Cnt Num
2   192.168.12.2       Fa0/0           14 00:24:03  128   768  0  12
1   10.175.113.66      Se0/1           12 00:25:11  109   654  0  23
0   10.175.113.34      Se0/0           13 00:25:11  109   654  0  24
```

You can also use **show ip eigrp interfaces** to see a quick summary of what interfaces are running EIGRP and how many peers are learned on each interface.

R1#**show ip eigrp interfaces**

IP-EIGRP interfaces for process 123

Interface	Peers	Xmit Queue Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Se0/0	1	0/0	51	0/15	199	0
Se0/1	1	0/0	50	0/15	187	0
Fa0/1	0	0/0	0	0/1	0	0
Lo0	0	0/0	0	0/1	0	0
Fa0/0	1	0/0	56	0/1	212	0

Another somewhat helpful show command is: **show ip eigrp traffic** which shows traffic statistics for EIGRP.

R3#**show ip eigrp traffic 123**

IP-EIGRP Traffic Statistics for AS 123
Hellos sent/received: 71/49
Updates sent/received: 5/9
Queries sent/received: 1/1
Replies sent/received: 1/1
Acks sent/received: 7/6
SIA-Queries sent/received: 0/0
SIA-Replies sent/received: 0/0
Hello Process ID: 175
PDM Process ID: 136
IP Socket queue: 0/2000/4/0 (current/max/highest/drops)
Eigrp input queue: 0/2000/4/0 (current/max/highest/drops)

Finally let's have a look at the routing table, we can see by default EIGRP is auto-summarizing routes to their classful boundary, let's take a minute and see what kind of fun this causes. EIGRP routes will start with a **D** you can also just show EIGRP routes with **show ip route eigrp**. **Important Note:** If you do not have as many interfaces on your routers, your output may be slightly different.

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
C 192.168.12.0/24 is directly connected, FastEthernet0/0
C 192.168.11.0/24 is directly connected, FastEthernet0/1
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
D 10.0.0.0/8 is a summary, 01:23:11, Null0
C 10.175.113.64/30 is directly connected, Serial0/1

```
C      10.175.113.32/30 is directly connected, Serial0/0
D    192.168.23.0/24 [90/30720] via 192.168.12.2, 01:21:00, FastEthernet0/0
150.123.0.0/16 is variably subnetted, 2 subnets, 2 masks
C      150.123.11.11/32 is directly connected, Loopback0
D      150.123.0.0/16 is a summary, 01:21:29, Null0
D    192.168.33.0/24 [90/33280] via 192.168.12.2, 01:20:57, FastEthernet0/0
```

Let's focus on 2 routes, the 192.168.23.0 route and the loopback network. From R1 we can see that we can reach the 192.168.23.0 network with a ping which is what we would expect.

R1#**ping 192.168.23.3**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/33/76 ms

However we can see that we can't reach either of the loopback networks from R1.

R1#**ping 150.123.22.22**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.123.22.22, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R1#**ping 150.123.33.33**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.123.33.33, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

To figure out why this is happening, let's have another look at the routing table, this time, just the loopback networks. We see two routes, one for our loopback0 interface and the other a summary route pointing to Null0.

Null0 is a special interface that simply discards anything sent to it, EIGRP uses it when making summaries because the idea is that with longest match routing. The router will never use its own summary routes but instead will use any shorter route. The problem is that with auto-summary, we don't have any longer match routes. In the next few show commands, we are going to just show you the portion of the output to focus on starting now.

R1#**show ip route | begin 150.123.0.0** *(on most 12.4 and later IOS, use the | sec command)*

```
    150.123.0.0/16 is variably subnetted, 2 subnets, 2 masks
C      150.123.11.11/32 is directly connected, Loopback0
D      150.123.0.0/16 is a summary, 01:29:23, Null0
```

Since the loopback is directly connected, the router will make a summary route to advertise to R2 and R3 and point it to Null0. When it receives the 150.123.0.0/16 route from the other routes, R1 sees it already has a directly connected network with the address 150.123.0.0/16 (The Null0 interface) and it simply ignores the update. The 192.168.23.0 route works because R1 doesn't have an interface with the 192.168.23.0 network configured on it, so there isn't a summary route for it.

R1#**show ip route 150.123.22.0**
Routing entry for 150.123.0.0/16
  Known via "eigrp 123", distance 5, metric 128256, type internal
  Redistributing via eigrp 123
  Routing Descriptor Blocks:
  * directly connected, via Null0
      Route metric is 128256, traffic share count is 1
      Total delay is 5000 microseconds, minimum bandwidth is 10000000 Kbit
      Reliability 255/255, minimum MTU 1514 bytes
      Loading 1/255, Hops 0

Here's a debug showing R1 routing the packets to oblivion.
*Mar 1 06:31:51.382: IP: tableid=0, s=150.123.11.11 (local), d=150.123.33.33 (Null0), routed via RIB
*Mar 1 06:31:51.386: IP: s=150.123.11.11 (local), d=150.123.33.33 (Null0), len 100, sending.
*Mar 1 06:31:53.378: IP: tableid=0, s=150.123.11.11 (local), d=150.123.33.33 (Null0), routed via RIB
*Mar 1 06:31:53.382: IP: s=150.123.11.11 (local), d=150.123.33.33 (Null0), len 100, sending.
*Mar 1 06:31:55.378: IP: tableid=0, s=150.123.11.11 (local), d=150.123.33.33 (Null0), routed via RIB

Note: When you try this on our own your output might not be exactly the same.

While the above is a bit of an extreme case, it is another reason why summaries need to carefully considered as it has the potential for routing loops and the loss of route visibility. This means that if one of R3's serial interfaces had a problem (10.0.0.0 network) the other routes wouldn't realize there was an issue. The solution is to disable auto-summary with the **no auto-summary** command. It is a best practice to always immediately disable auto-summary unless you have a good reason for using it.

R1(config)#**router eigrp 123**
R1(config-router)#**no auto**
*Mar 1 06:42:51.302: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 10.175.113.34 (Serial0/0) is resync:
*Mar 1 06:42:51.306: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 10.175.113.66 (Serial0/1) is resync:
*Mar 1 06:42:51.310: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 192.168.12.2 (FastEthernet0/0) is resync:

R2(config)#**router eigrp 123**

R2(config-router)#**no auto**
*Mar 1 06:42:58.698: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 192.168.12.1 (FastEthernet0/0) is resync:
*Mar 1 06:42:58.702: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 192.168.23.3 (FastEthernet1/0) is resync:

R3(config)#**router eigrp 123**
R3(config-router)#**no auto**
*Mar 1 06:42:45.202: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 10.175.113.33 (Serial0/0) is resync:
*Mar 1 06:42:45.206: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 10.175.113.65 (Serial0/1) is resync:
*Mar 1 06:42:45.210: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123:
Neighbor 192.168.23.2 (FastEthernet0/0) is resync:

After the EIGRP resyncs the routing table, we can now see the individual /32 routes for the loopbacks. Please focus on the route below in the output.

R1#**show ip route | sec 150.123.0.0** *(on most 12.4 and later IOS, use the | sec command)*
150.123.0.0/32 is subnetted, 3 subnets
D 150.123.33.33 [90/158720] via 192.168.12.2, 00:00:22, FastEthernet0/0
D 150.123.22.22 [90/156160] via 192.168.12.2, 00:00:21, FastEthernet0/0
C 150.123.11.11 is directly connected, Loopback0

And we can reach all of them from R1.

R1#**ping 150.123.22.22**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.123.22.22, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/20/56 ms

R1#**ping 150.123.33.33**
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.123.33.33, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/20/40 ms

As mentioned above, EIGRP is a very fast protocol that can detect issues very quickly. It does this with the concept of successors and feasible successors; the successor is the route that is chosen for the routing table. This is chosen by the best metric (Bandwidth + Delay by default) routes also have to pass EIGRPs loop prevention rule which says that the Advertised Distance of a route (R2 -> R3) will be lower than the Feasible Distance (R1 -> R2 -> R3). EIGRP also stores a number of feasible successors so that in case something goes wrong with the successor it can switch routes as soon as it knows there is a problem.

R1#**show ip eigrp topology**

IP-EIGRP Topology Table for AS(123)/ID(150.123.11.11)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - reply Status, s - sia Status
P 192.168.33.0/24, 1 successors, FD is 33280
via 192.168.12.2 (33280/30720), FastEthernet0/0
via 10.175.113.34 (2172416/28160), Serial0/0
via 10.175.113.66 (2172416/28160), Serial0/1
P 192.168.11.0/24, 1 successors, FD is 28160
via Connected, FastEthernet2/0
P 192.168.12.0/24, 1 successors, FD is 28160
via Connected, FastEthernet1/0
P 192.168.23.0/24, 1 successors, FD is 30720
via 192.168.12.2 (30720/28160), FastEthernet0/0
via 10.175.113.66 (2172416/28160), Serial0/1
via 10.175.113.34 (2172416/28160), Serial0/0
P 10.175.113.64/30, 1 successors, FD is 2169856
via Connected, Serial0/1
P 150.123.33.33/32, 1 successors, FD is 158720
via 192.168.12.2 (158720/156160), FastEthernet0/0
via 10.175.113.34 (2297856/128256), Serial0/0
via 10.175.113.66 (2297856/128256), Serial0/1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - reply Status, s - sia Status
P 150.123.22.22/32, 1 successors, FD is 156160
via 192.168.12.2 (156160/128256), FastEthernet0/0
P 150.123.11.11/32, 1 successors, FD is 128256
via Connected, Loopback0
P 10.175.113.32/30, 1 successors, FD is 2169856
via Connected, Serial0/0

Let's test this out, according to the topology table R1 is prefering the Fa0/0 interface to reach R3's loopback. After a bit, I disabled R3's Fa0/0 interface by simply unplugging it. Notice how it takes about 7 packets to figure out there is a problem and switch over?

R1#**ping 150.123.33.33 repeat 100000** *(if the repeat command does not work for you as you are running an older version of IOS, simply type ping with no ip address. You will then be prompted for the protocol, IP address and how many times to send with the repeat option where you can enter 100000).*
Type escape sequence to abort.
Sending 100000, 100-byte ICMP Echos to 150.123.33.33, timeout is 2 seconds:
!!
Success rate is **99 percent** (**840/847**), round-trip min/avg/max = 1/9/80 ms

To demonstrate how fast the switch over actually is lets debug the routing table with **debug ip routing.** Debug ip routing is not really a CCNA command but it is pretty useful to see any changes to the routing table in real time. Notice that as soon as EIGRP detects the R3 interface is down and removes the route, within the same second it has switched to the serial links to route the traffic.

R1# **debug ip routing**
IP routing debugging is on
R1#**ping 150.123.33.33 repeat 100000**
Type escape sequence to abort.
Sending 100000, 100-byte ICMP Echos to 150.123.33.33, timeout is 2 seconds:
!!
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!......
*Mar 1 07:26:27.174: RT: delete route to 150.123.33.33 via 192.168.12.2, eigrp metric [90/158720]
*Mar 1 07:26:27.178: RT: SET_LAST_RDB for 150.123.33.33/32
*Mar 1 07:26:27.190: RT: add 150.123.33.33/32 via 10.175.113.34, eigrp metric [90/2297856]
*Mar 1 07:26:27.190: RT: NET-RED 150.123.33.33/32
*Mar 1 07:26:27.190: RT: add 150.123.33.33/32 via 10.175.113.66, eigrp metric [90/2297856]!!!!!!!!!!!!

Checking the table now we can see that EIGRP is now load balancing between both serial links because they have equal metrics. Focus on this portion of the output.

R1# **show ip route | begin 150.123.33.33** *(on most 12.4 and later IOS, use the | sec command)*
D 150.123.33.33 [90/2297856] via 10.175.113.66, 00:13:21, Serial0/1
[90/2297856] via 10.175.113.34, 00:13:21, Serial0/0

Let's bring R3's Ethernet link back up by plugging the cable back in.

R3
*Mar 1 07:40:43.874: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 07:40:44.070: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123: Neighbor 192.168.23.2 (FastEthernet0/0) is up: new adjacency
*Mar 1 07:40:44.874: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Let's examine the topology table a bit more closely. Specifically 150.123.33.33/32.

R1#**show ip eigrp topology | begin 150.123.33.33/32** *(on most 12.4 and later IOS, use the | sec command)*
P 150.123.33.33/32, 1 successors, FD is 158720
via 192.168.12.2 (158720/156160), FastEthernet0/0
via 10.175.113.34 (2297856/128256), Serial0/0
via 10.175.113.66 (2297856/128256), Serial0/1

If you want to load balance between unequal links (like for example a FastEthernet and a Serial interface) you have two options. One is to modify the bandwidth and delay on the interface so that its equal with the other interfaces the other is to use the **variance** command. In either case its helpful to look at the detailed topology info for the routes you're interested in. You can do this with the **show ip eigrp topology <route>** command. The command will show you the complete metric info for each interface that knows about the route and also the FD/AD.

```
R1#show ip eigrp topology 150.123.33.33/32
IP-EIGRP (AS 123): Topology entry for 150.123.33.33/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 158720
  Routing Descriptor Blocks:
  192.168.12.2 (FastEthernet0/0), from 192.168.12.2, Send flag is 0x0
      Composite metric is (158720/156160), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 5200 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
  10.175.113.34 (Serial0/0), from 10.175.113.34, Send flag is 0x0
      Composite metric is (2297856/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  10.175.113.66 (Serial0/1), from 10.175.113.66, Send flag is 0x0
      Composite metric is (2297856/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

To change the metric info on interface, we can do this with the **bandwidth** and **delay** commands. It is important to remember that the bandwidth command is purely logical - meaning it's only used by protocols such as EIGRP or QoS to determine the link bandwidth. It will not affect the actual **speed** of a link, only the speed command can do that.

However, it's a bit of a painful trial and error process to fine tune the metrics on the interfaces and can affect other protocols that rely on the bandwidth command. Below shows how to adjust the metrics. But do **not** run these commands now.

```
R1(config)#int s0/0
R1(config-if)#bandwidth 100000
R1(config-if)#delay 55
```

The better way to do it is using the **variance** command. The variance command is used to tell the router what multiple of feasible distance is should be considered for unequal load

balancing.
The syntax is: **variance <multiplier>**

For example if R1's Fa0/0 interface has a Feasible Distance of 158720 if we were to configure a variance of 2 then R1 would accept anything with a Feasible Distance between the range of 158720 - 317440 will be used for load balancing. It's worth noting that unequal load balancing isn't 1 to 1 between the interfaces but instead quoting Cisco, "the router distributes traffic proportionately to the ratios of the metrics that are associated with different routes." Which simply means the router will intelligently send more traffic across the faster links then the slower ones.

So looking at the FD for the Ethernet link is 158720 and the Serial links have a FD of 2297856. To find the value of variance you take the highest interface FD and divide it by the lowest interface FD (the successor). Look for the info in the output below.

R1#**show ip eigrp topology | begin 150.123.33.33/32** *(on most 12.4 and later IOS, use the | sec command)*
P 150.123.33.33/32, 1 successors, FD is 158720
    via 192.168.12.2 (158720/156160), FastEthernet0/0
    via 10.175.113.34 (2297856/128256), Serial0/0
    via 10.175.113.66 (2297856/128256), Serial0/1

The highest FD is 2297856 and the successor FD is 158720.

R1(config)#**router eigrp 123**
R1(config-router)#**variance 14**

You'll notice that nothing changed in the routing table, this is because 14 is a clean divide. It's actually too small, we will need to increase the variance to 15 instead.

R1#**show ip route eigrp**
D   192.168.23.0/24 [90/30720] via 192.168.12.2, 00:03:14, FastEthernet0/0
     150.123.0.0/32 is subnetted, 3 subnets
D   150.123.33.33 [90/158720] via 192.168.12.2, 00:03:14, FastEthernet0/0
D   150.123.22.22 [90/156160] via 192.168.12.2, 00:03:14, FastEthernet0/0
D   192.168.33.0/24 [90/33280] via 192.168.12.2, 00:03:14, FastEthernet0/0

After we change the variance to 15 we see the routes being added to the routing table since we still have **debug ip routing** on.

R1(config)#**router eigrp 123**
R1(config-router)#**variance 15**
R1(config-router)#**exit**
R1#**clear ip eigrp neighbors**
*Mar 2 01:52:31.553: RT: add 150.123.33.33/32 via 10.175.113.34, eigrp metric [90/2297856]
*Mar 2 01:52:31.557: RT: NET-RED 150.123.33.33/32
*Mar 2 01:52:31.561: RT: add 150.123.33.33/32 via 10.175.113.66, eigrp metric [90/2297856]
*Mar 2 01:52:31.565: RT: NET-RED 150.123.33.33/32

You **must** run the *clear ip eigrp neighbors* command every time you change the variance on most pre 12.4 IOS routers. You will then see the route changes. Now we can we see the other paths for R3's loopback. Look for this in the output.

R1#**show ip route**
D   150.123.33.33 [90/158720] via 192.168.12.2, 00:04:00, FastEthernet0/0
                [90/2297856] via 10.175.113.66, 00:04:00, Serial0/1
                [90/2297856] via 10.175.113.34, 00:04:00, Serial0/0

EIGRP communicates with its peers by sending hello packets every 5 seconds for high speed broadcast links and every 60 seconds for slow speed NBMA links (Frame-relay). If you need to adjust the timers (perhaps you want EIGRP to detect link failures more quickly) you can use the **ip hello-interval eigrp <AS> <seconds>** command. When you change the hello time, you'll also need to adjust the **hold-timer** with the **ip hold-time eigrp <AS> <seconds>** As a rule of thumb the hold time should be 3 times as much as the hello interval.

Note: If you are going to change timers it is important to make sure the other side of the link also is configured for the same value to avoid any issues.

Let's change the hello time on the Fast Ethernet links to be 1 second instead of 5, we'll also adjust the hold time.

R1(config)#**int fa0/0**
R1(config-if)#**ip hello-interval eigrp 123 1**
R1(config-if)#**ip hold-time eigrp 123 3**

R2(config)#**int fa0/1**
R2(config-if)#**ip hello-interval eigrp 123 1**
R2(config-if)#**ip hold-time eigrp 123 3**
R2(config-if)#**int fa0/0**
R2(config-if)#**ip hello-interval eigrp 123 1**
R2(config-if)#**ip hold-time eigrp 123 3**

R3(config)#**int fa0/0**
R3(config-if)#**ip hello-interval eigrp 123 1**
R3(config-if)#**ip hold-time eigrp 123 3**

Now let's try our ping test again with a timeout of 1 second which is our hello time. I'll ping R3's loopback then disconnect R3's Fa0/0 cable. Notice that this time we only lost 3 ping packets which is what we would expect, after R1 misses 3 hellos from the Fa0/0 path it switches over to the serial links.

R1#**ping 150.123.33.33 repeat 100000 timeout 1**
Type escape sequence to abort.
    Sending 100000, 100-byte ICMP Echos to 150.123.33.33, timeout is 1 seconds:

!!
!!
!!
Success rate is 98 percent (278/281), round-trip min/avg/max = 1/12/64 ms

Failover time is great when there is a feasible successor for a route because EIGRP simply switches over to the new one. But what happens when EIGRP doesn't have a feasible successor and there is a problem? When EIGRP doesn't have a backup route it will send a query out to all of its neighbors asking if they know of a way to get to the downed network. This process is called the Active state because the router is actively trying to solve reachability problems, this is the opposite of passive state where everyone is in sync and there is not a lot of chatter in the network. When neighbors receive a query they will do one of two things: Either they will have a route and respond with an UPDATE packet or if the router doesn't it will send out its own query to all of its neighbors to see if it can figure out the route.

The downside of this system is that a router will only send out one query packet per neighbor in an effort to minimize redundant chatter, while it is waiting the router will keep the router an Active state until it receives either a reply or an update packet from each neighbor. During this time route is still kept in the routing table. In large EIGRP networks it may take awhile for all the query packets to be answered or worse yet if the QUERY or the REPLY packet is lost in transit due for whatever reason every router looking for the route will need to wait for the hold time to expire and everyone to resync before the network turns passive again. This can be a massive waste of bandwidth as well as be a lengthy outage considering the hold-time on NBMA links is 180 seconds or 3 minutes. This issue is called Stuck in Active.

To help explore this we'll change the timers on the Fast Ethernet links to 60/180 and we'll change the S0/0 links to 1/3.

R1(config-if)#**int s0/0**
R1(config-if)# **ip hello-interval eigrp 123 1**
R1(config-if)# **ip hold-time eigrp 123 3**
R1(config-if)#**int fa0/0**
R1(config-if)# **ip hello-interval eigrp 123 60**
R1(config-if)# **ip hold-time eigrp 123 180**

R2(config-if)#**int fa0/0**
R2(config-if)# **ip hello-interval eigrp 123 60**
R2(config-if)# **ip hold-time eigrp 123 180**
R2(config-if)#**int fa0/1**
R2(config-if)# **ip hello-interval eigrp 123 60**
R2(config-if)# **ip hold-time eigrp 123 180**

R3(config-if)#**int s0/0**
R3(config-if)# **ip hello-interval eigrp 123 1**
R3(config-if)# **ip hold-time eigrp 123 3**
R3(config-if)#**int fa0/0**

R3(config-if)# **ip hello-interval eigrp 123 60**
R3(config-if)# **ip hold-time eigrp 123 180**

Next we'll apply a standard ACL on R1's Fa0/0 to block all traffic and shut down R3's S0/0.

R1(config)#**access-list 5 deny any**
R1(config)#**int fa0/0**
R1(config-if)#**ip access-group 5 in**

R3(config-if)#**int s0/0**
R3(config-if)#**shut**

If we turn on **debug ip eigrp** we can see looking for the serial network.

R1#**debug ip eigrp**
*Mar 2 04:54:00.048: IP-EIGRP(Default-IP-Routing-Table:123): 10.175.113.32/30 - not in IP routing table
*Mar 2 04:54:00.052: IP-EIGRP(Default-IP-Routing-Table:123): Int 10.175.113.32/30
metric 4294967295 - 0 4294967295

On R2 we can see the QUERY packet and a few milliseconds later it receives a REPLY packet from R3 confirming there is no other route.

*Mar 2 04:53:59.744: IP-EIGRP(Default-IP-Routing-Table:123): Processing incoming QUERY packet

*Mar 2 04:53:59.920: IP-EIGRP(Default-IP-Routing-Table:123): Processing incoming REPLY packet

Since we are blocking any traffic on R1's Fa0/0 interface it will not receive R2's REPLY so it will remain in active, we can see this by checking show ip eigrp topology.

R1#**show ip eigrp topology**
IP-EIGRP Topology Table for AS(123)/ID(150.123.11.11)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - reply Status, s - sia Status
P 192.168.33.0/24, 1 successors, FD is 33280
via 192.168.12.2 (33280/30720), FastEthernet0/0
P 192.168.11.0/24, 1 successors, FD is 28160
via Connected, FastEthernet0/1
P 192.168.12.0/24, 1 successors, FD is 28160
via Connected, FastEthernet0/0
P 192.168.23.0/24, 1 successors, FD is 30720
via 192.168.12.2 (30720/28160), FastEthernet0/0
P 150.123.22.22/32, 1 successors, FD is 156160
via 192.168.12.2 (156160/128256), FastEthernet0/0
P 150.123.33.33/32, 1 successors, FD is 158720
via 192.168.12.2 (158720/156160), FastEthernet0/0
P 150.123.11.11/32, 1 successors, FD is 128256
via Connected, Loopback0
A 10.175.113.32/30, 1 successors, FD is Inaccessible, Q

1 replies, active 00:00:17, query-origin: Local origin
Remaining replies:
via 192.168.12.2, r, FastEthernet0/0

Eventually the hold-time expires and the route is removed.

R1#
*Mar 2 04:55:46.068: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 123: Neighbor 192.168.12.2
 (FastEthernet0/0) is down: holding time expired

Now shut off the debug and review the topology table to see that the route has been removed.

R1#**debug ip eigrp**

Finally EIGRP can be a bit of a chatty protocol with all its hellos and various other packets it sends frequently it's possible on some slow WAN links that EIGRP can drown out actual data traffic if EIGRP is busy enough. By default EIGRP may take up to 80% of a link for its own communications. You can adjust this value with the **ip bandwidth-percent eigrp <AS> <percent>** command under an interface.

Note: The bandwidth-percent command replies on the configured bandwidth on the interface, if this value is wrong the command won't work as expected.

R1(config)#**int s0/0**
R1(config-if)#**ip bandwidth-percent eigrp 123 20**

Examine the routing tables of all three routers starting with Router1 and then going to Router2 and Router3.

EIGRP Review Questions

1) What is the administrative distance of EIGRP?_______________________________

2) Does EIGRP support load balancing by default? ______________________________

3) From the Router1 router, what would be the command to display the EIGRP topology table?____

4) What command would save the current configuration of all the routers?__________

Answers

1) 90 is for internal EIGRP and 170 for external EIGRP.
2) Yes, only equal cost load balancing. Unequal cost load balancing can be enabled using the variance command.
3) show ip eigrp topology
4) copy run start