Introduction

The **Configuring EIGRP** module provides you with the instructions and Cisco hardware to develop your hands on skills in EIGRP, including authentication and receiving routes over a wide area network. This module includes the following exercises:

1) Basic EIGRP configuration

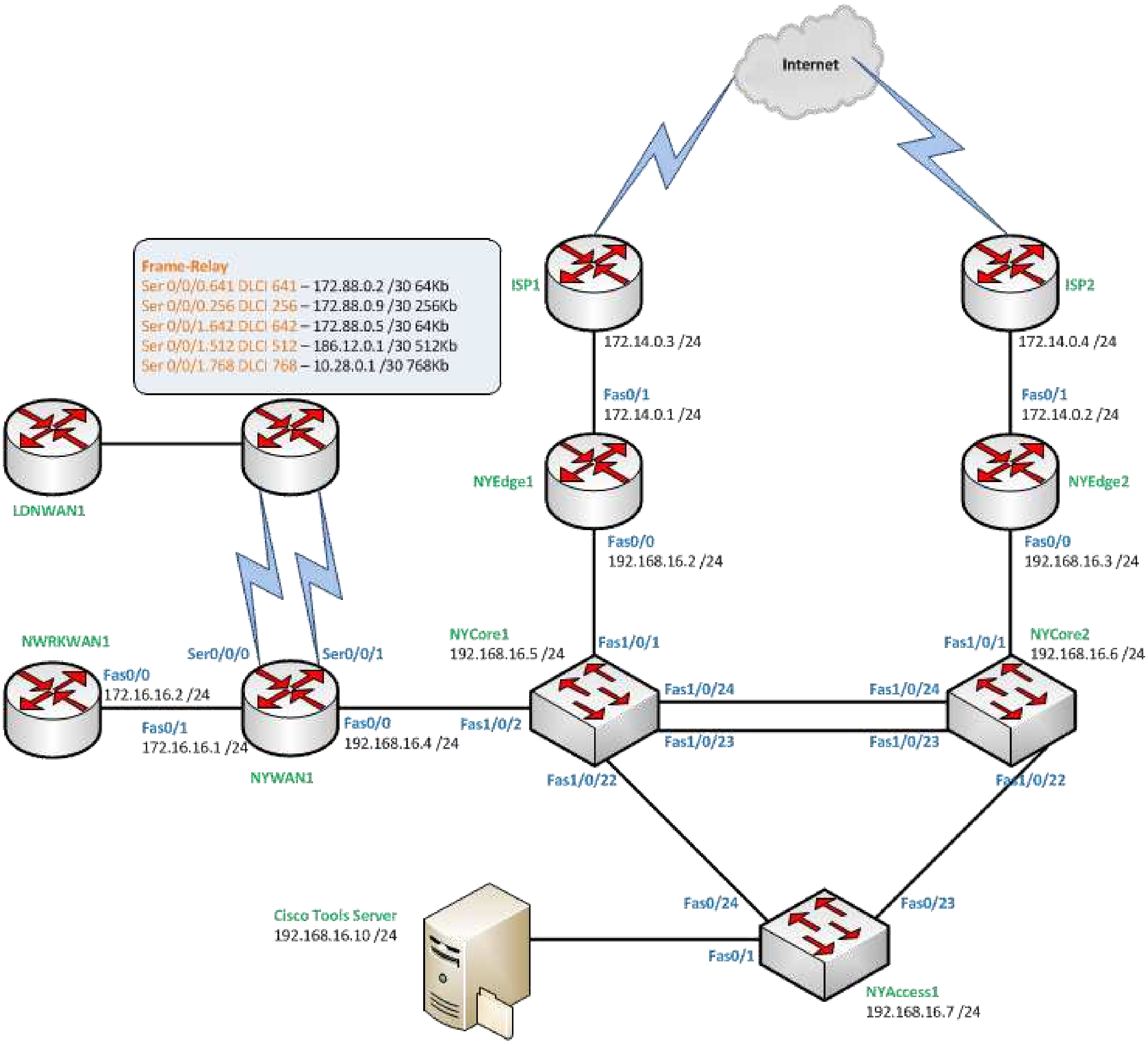
2) EIGRP Authentication

3) EIGRP over WAN links

4) Tuning EIGRP

Lab Diagram

During your session you will have access to the following lab configuration. Depending on the exercises you may or may not use all of the devices, but they are shown here in the layout to get an overall understanding of the topology of the lab.



Connecting to your lab

In this module you will be working on the following equipment to carry out the steps defined in each exercise.

- NYEDGE1
- NYEDGE2
- NYWAN1
- NYCORE1
- NYCORE2
- NYACCESS1
- PLABCSCO01

Each exercise will detail which terminal you are required to work on to carry out the steps.

During the boot up process an activity indicator will be displayed in the device name tab:

- Black - Powered Off
- Orange - Working on your request
- Green - Ready to access

If the remote terminal is not displayed automatically in the main window (or popup) click the **Connect** icon located in the tools bar to start your session.

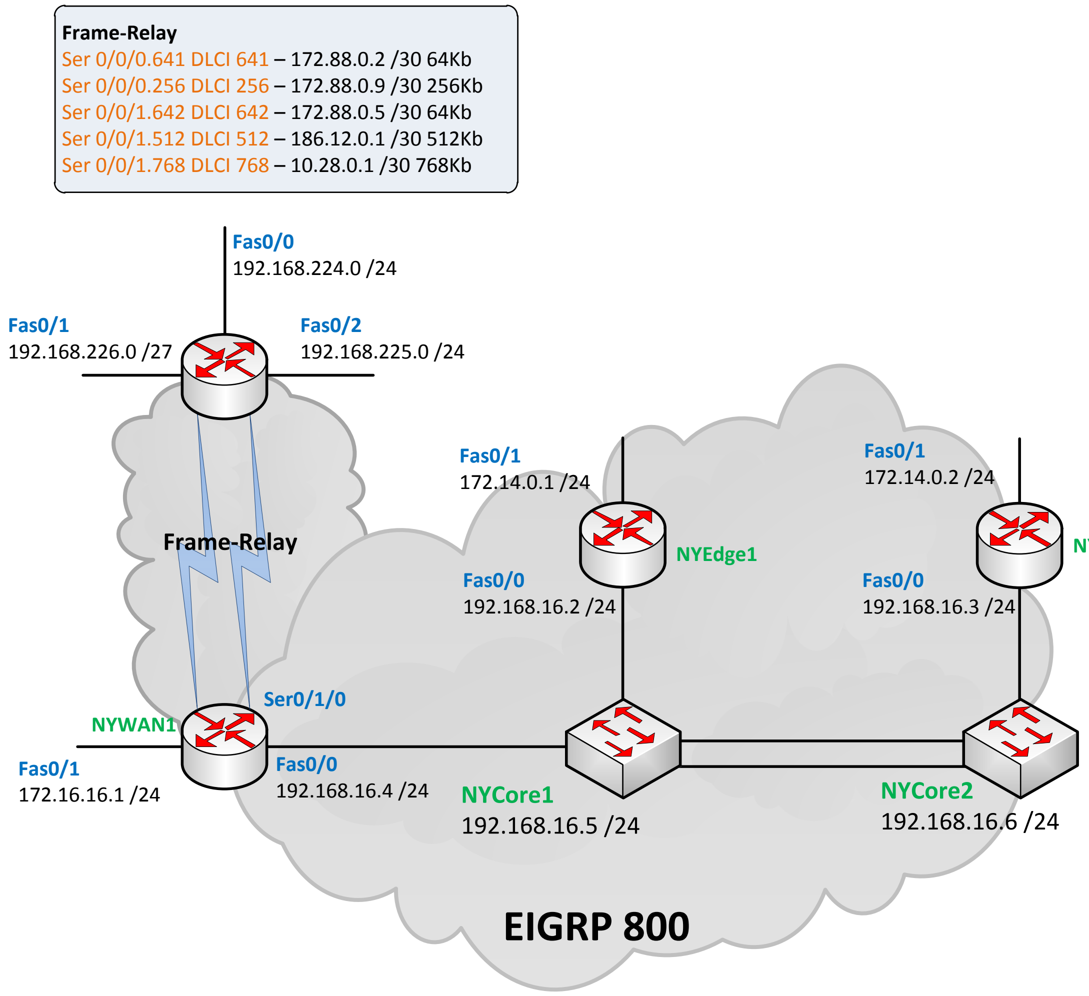
Copyright Notice

Exercise 1 – Basic EIGRP configuration

In this exercise you will learn how to configure the basics of Cisco's EIGRP routing protocol. Please refer to your course material or use your preferred search engine to gain an understanding of these tasks.

Lab Diagram

This diagram focuses on the devices and topology in this exercise.



Configuring EIGRP

In this example you will configure EIGRP AS 800 between the three routers within the lab. Within the configurations we'll use a different approach to advertising networks and forming neighbor relationships.

Use the following steps to create the base EIGRP configuration.

Step 1

Configure the NYEDGE1 router with EIGRP AS 800 and attempt to build relationships on the 192.168.16.0/24 subnet:

```
NYEDGE1(config)#router eigrp 800

NYEDGE1(config-router)#network 192.168.16.0 0.0.0.255
```

Step 2

Configure NYEDGE2:

```
NYEDGE2(config)#router eigrp 800

NYEDGE2(config-router)#network 192.168.16.0

NYEDGE2(config-router)#no auto-summary
```

Step 3

Finally we'll configure the NYWAN router, this time advertising both its 192.168.16.0 networks and its 172.16.16.0 network. However, we don't want to form any relationships on the 172.16.16.0 subnet therefore we use the passive interface command:

```
NYWAN1#configure terminal

NYWAN1(config)#router eigrp 800

NYWAN1(config-router)#passive-interface fastEthernet 0/1

NYWAN1(config-router)#network 172.16.16.0 0.0.0.255

NYWAN1(config-router)#network 192.168.16.0

NYWAN1(config-router)#no auto-summary
```

Note: Notice how for the 172.16.16.0 network a mask was used and for the 192.168.16.0 network the mask was dropped. Why can we do this?

Step 4

Observe the routing tables of NYEDGE1 and NYEDGE2:

```
NYEDGE1#show ip route

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

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

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

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

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

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

       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
     172.14.0.0/24 is subnetted, 1 subnets
C       172.14.0.0 is directly connected, FastEthernet0/1
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.16.0 [90/30720] via 192.168.16.4, 00:00:23, FastEthernet0/0
C    192.168.16.0/24 is directly connected, FastEthernet0/0
```

Observe the routing table of NYWAN1:

```
NYWAN1#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
     186.12.0.0/30 is subnetted, 1 subnets
C       186.12.0.0 is directly connected, Serial0/0/1.512
     172.88.0.0/30 is subnetted, 3 subnets
C       172.88.0.4 is directly connected, Serial0/0/1.642
C       172.88.0.0 is directly connected, Serial0/0/0.641
C       172.88.0.8 is directly connected, Serial0/0/0.256
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.16.0 is directly connected, FastEthernet0/1
     10.0.0.0/30 is subnetted, 1 subnets
C       10.28.0.0 is directly connected, Serial0/0/1.768
C    192.168.16.0/24 is directly connected, FastEthernet0/0
```

Notice how not all the subnets on each of the routers are being advertised, for example NYWAN1 is not advertising its serial interface subnets and similarly NYEDGE1 & 2 are not advertising their Fas0/1 subnets. This is because we have not told EIGRP to do this.

Verifying your configuration

Now that you have configured EIGRP, whilst you have viewed the routing table to confirm you are seeing routes advertised by EIGRP from different neighbors, let's use some additional commands to help us verify EIGRP.

Use the following commands on NYWAN1:

- show ip eigrp interfaces
- show ip protocols
- show ip eigrp topology
- show ip eigrp topology all-links

Output from the **show ip eigrp interfaces** command shows that we are not multicasting on the FastEtethernet 0/1 interface:

```
NYWAN1#show ip eigrp interfaces

IP-EIGRP interfaces for process 800

                        Xmit Queue   Mean   Pacing Time   Multicast    Pending

Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes

Fa0/0              2        0/0      1273       0/1           50          0
```

Output from the **show ip protocols** command verifies that EIGRP is running on the router (and any other routing protocols). This command is quite useful as it gives us a summary of neighbors that are communicating with us and networks that we are sending out:

```
NYWAN1#show ip protocols

Routing Protocol is "eigrp 800"

  Outgoing update filter list for all interfaces is not set

  Incoming update filter list for all interfaces is not set

  Default networks flagged in outgoing updates

  Default networks accepted from incoming updates

  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0

  EIGRP maximum hopcount 100

  EIGRP maximum metric variance 1

  Redistributing: eigrp 800

  EIGRP NSF-aware route hold timer is 240s

  Automatic network summarization is not in effect

  Maximum path: 4

  Routing for Networks:
```

```
    172.16.16.0/24
    192.168.16.0
  Passive Interface(s):
    FastEthernet0/1
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      00:02:30
    192.168.16.2          90      00:02:28
    192.168.16.3          90      00:02:28
  Distance: internal 90 external 170
```

Output from the **show ip eigrp topology** command shows us the topology table:

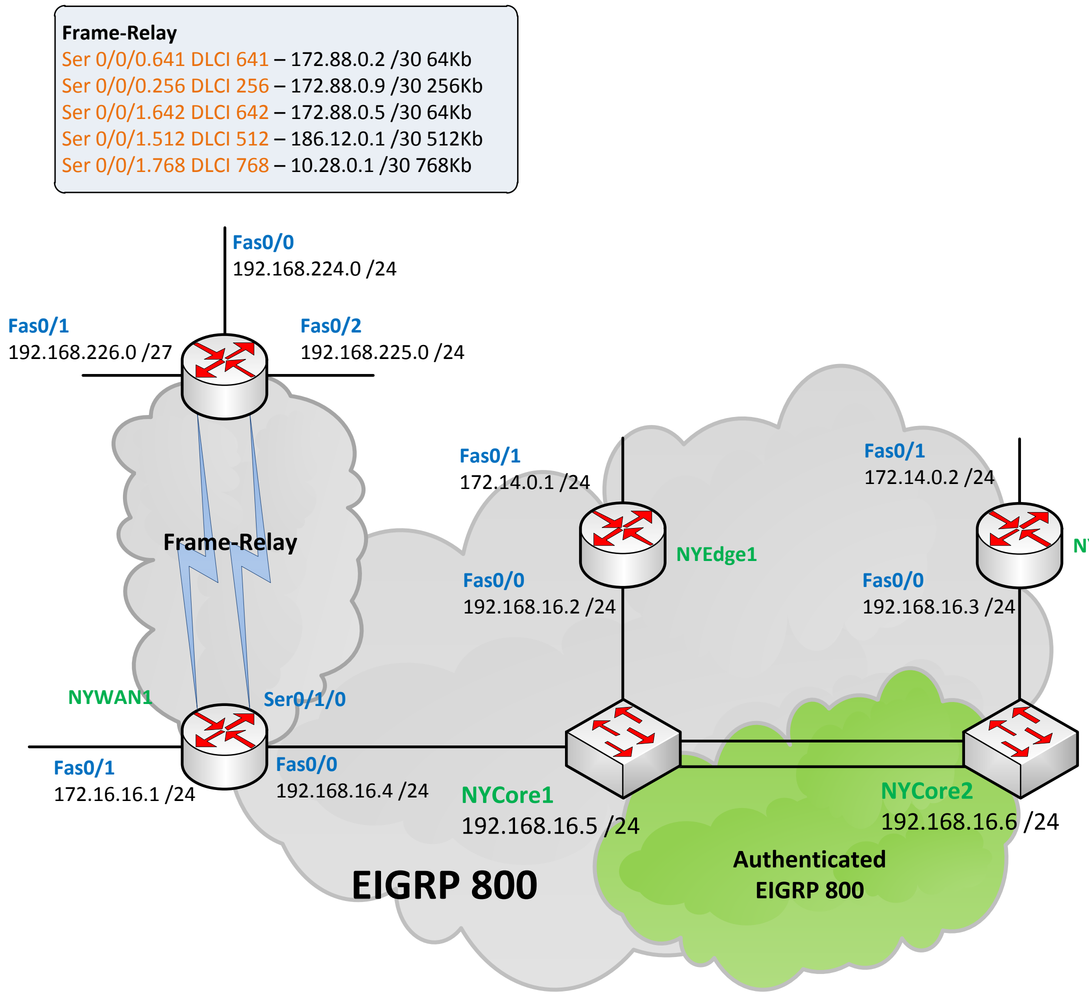
```
NYWAN1#show ip eigrp topology
IP-EIGRP Topology Table for AS(800)/ID(192.168.16.4)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 192.168.16.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
P 172.16.16.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/1
```

Leave your devices on and with their current configurations and continue to the next exercise.

Exercise 2 – EIGRP Authentication

In this exercise you will learn how to configure EIGRP authentication between EIGRP neighbors. Please refer to your course material or use your preferred search engine to gain an understanding of these tasks.

Lab Diagram



EIGRP Authentication

Some organisations require that their EIGRP neighbor relationships are authenticated, or some parts of their network require authentication. This is becoming more common as network engineers and organisations are becoming ever increasingly more security aware and conscious.

Another reason you may authenticate part of your network is because you have third party devices being present in a subset of your network, you wouldn't want to start advertising your entire network topology to a third party, would you?

In this exercise you will configure EIGRP authentication between NYCORE1 and NYCORE2 on VLAN 100.

Step 1

First we need to enable the EIGRP process on NYCORE1:

```
NYCORE1(config)#router eigrp 800

NYCORE1(config-router)#network 192.168.100.0

NYCORE1(config-router)#no auto-summary
```

Step 2

Next we configure the authentication keys:

```
NYCORE1(config)#key chain KEYS

NYCORE1(config-keychain)#key 1

NYCORE1(config-keychain-key)#key-string practice

NYCORE1(config-keychain-key)#accept-lifetime 00:00:00 1 January 1993 infinite

NYCORE1(config-keychain-key)#send-lifetime 00:00:00 1 January 1993 infinite
```

Step 3

Lastly we configure EIGRP to authenticate its peers on VLAN 100:

```
NYCORE1(config)#interface vlan 100

NYCORE1(config-if)#ip authentication mode eigrp 800 md5

NYCORE1(config-if)#ip authentication key-chain eigrp 800 KEYS
```

Step 4

Confirm our authentication is working by configuring NYCORE2:

```
NYCORE2(config)#router eigrp 800

NYCORE2(config-router)#network 192.168.100.0

NYCORE2(config-router)#no auto-summary
```

Enable EIGRP packet debugging on NYCORE1 using the **debug eigrp packets** command:

```
NYCORE1#debug eigrp packets

    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUERY,
SIAREPLY)

EIGRP Packet debugging is on

NYCORE1#
```

```
*Mar  1 00:28:01.303: EIGRP: Vlan100: ignored packet from 192.168.100.3, opcode = 5 (missing authentication)
```

Note the invalid authentication error.

To turn off EIGRP packet debugging type: **no debug eigrp packets**

Step 5

Configure NYCORE2 to use the same authentication process:

```
NYCORE2(config)#key chain KEYS

NYCORE2(config-keychain)#key 1

NYCORE2(config-keychain-key)#key-string practice

NYCORE2(config-keychain-key)#accept-lifetime 00:00:00 1 January 1993 infinite

NYCORE2(config-keychain-key)#send-lifetime 00:00:00 1 January 1993 infinite

NYCORE2(config)#interface vlan 100

NYCORE2(config-if)#ip authentication mode eigrp 800 md5

NYCORE2(config-if)#ip authentication key-chain eigrp 800 KEYS
```

You should now have an EIGRP relationship between the NYCORE1 and NYCORE2 switches.

```
*Mar  1 00:29:50.875: %DUAL-5-NBRCHANGE: EIGRP-IPv4 800: Neighbor 192.168.100.2 (Vlan100) is up: new adjacency
```

Step 6

Finally we will configure the two switches to advertise on the 192.168.16.0 subnet:

NYCORE1

```
NYCORE1(config)#router eigrp 800

NYCORE1(config-router)#network 192.168.16.0
```

NYCORE2

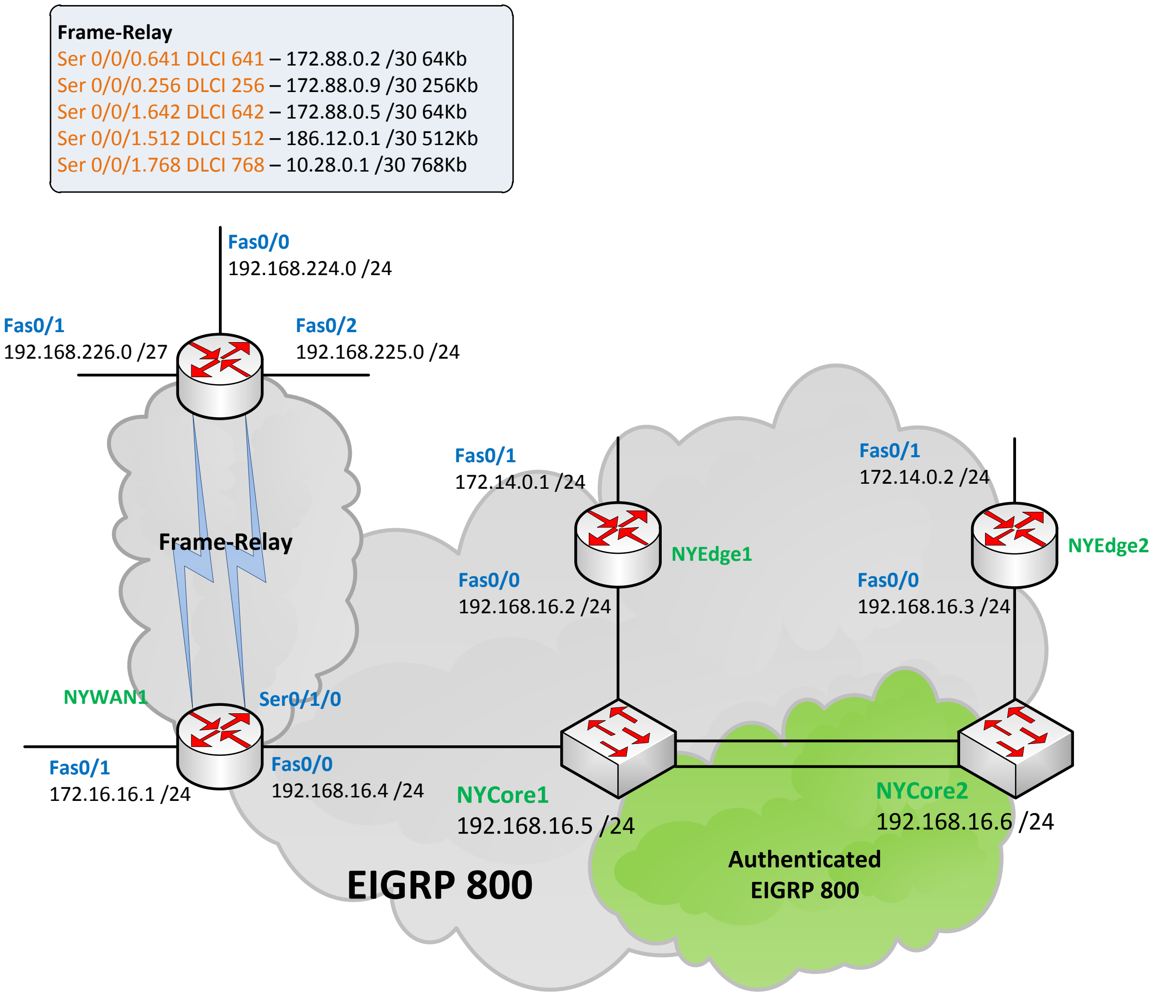
```
NYCORE2(config)#router eigrp 800

NYCORE2(config-router)#network 192.168.16.0
```

Before continuing to the next exercise, have a look at the routing table now present on the other routers. Do you notice anything different, are there routes that look slightly different (hint equal cost paths)?

Exercise 3 – EIGRP over WAN links

In this exercise you will learn how to configure EIGRP to run over a WAN links, including configuring EIGRP to operate in low bandwidth environments. Please refer to your course material or use your preferred search engine to gain an understanding of these tasks.

Lab Diagram



Configuring EIGRP on WAN links

EIGRP does not normally need any special requirements when configuring on WAN links like some other routing protocols. But there may be some design considerations that need to be taken in to account when dealing with the WAN.

In this exercise you will configure EIGRP to run on a frame-relay network with low bandwidth links.

Step 1

First configure EIGRP to run on the additional networks for the serial interface links that connect to the frame relay network:

```
NYWAN1#sho ip interface brief
```

```
Interface                  IP-Address      OK? Method Status
Protocol

FastEthernet0/0            192.168.16.4    YES manual up                    up

FastEthernet0/1            172.16.16.1     YES manual up                    up

Serial0/0/0                unassigned      YES unset  up                    up

Serial0/0/0.256            172.88.0.9      YES manual up                    up

Serial0/0/0.641            172.88.0.1      YES manual up                    up

Serial0/0/1                unassigned      YES unset  up                    up

Serial0/0/1.512            186.12.0.1      YES manual up                    up

Serial0/0/1.642            172.88.0.5      YES manual up                    up

Serial0/0/1.768            10.28.0.2       YES manual up                    up

Serial0/1/0                unassigned      YES unset  administratively down down
```

NYWAN1 configuration:

```
NYWAN1(config)#router eigrp 800

NYWAN1(config-router)#network 172.88.0.0 0.0.0.15

*May 22 13:07:28: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.10
(Serial0/0/0.256) is up: new adjacency

*May 22 13:07:28: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.6
(Serial0/0/1.642) is up: new adjacency

*May 22 13:07:28: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.2
(Serial0/0/0.641) is up: new adjacency
```

You should see a new EIGRP neighbour across each link:

```
NYWAN1#show ip eigrp neighbors

IP-EIGRP neighbors for process 800

H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq

                                            (sec)         (ms)       Cnt Num

6   172.88.0.2              Se0/0/0.641       13 00:00:29  797  4782  0  260

5   172.88.0.6              Se0/0/1.642       12 00:00:29  797  4782  0  259

4   172.88.0.10             Se0/0/0.256       13 00:00:29  795  4770  0  257

3   192.168.16.6            Fa0/0             13 00:03:30   18   200  0  19

2   192.168.16.5            Fa0/0             10 00:03:44   12   200  0  17
```

```
1   192.168.16.2          Fa0/0          12 00:18:12  520  3120  0  13
0   192.168.16.3          Fa0/0          11 00:18:12  521  3126  0  14
```

Notice what mask we used when configuring the network address under EIGRP?

Step 2

Next, view the routing table on the router. You will notice that there is a problem:

```
NYWAN1#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
     192.168.226.0/27 is subnetted, 2 subnets
D       192.168.226.0 [90/2297856] via 172.88.0.10, 00:01:38, Serial0/0/0.256
                      [90/2297856] via 172.88.0.6, 00:01:38, Serial0/0/1.642
                      [90/2297856] via 172.88.0.2, 00:01:38, Serial0/0/0.641
D       192.168.226.32 [90/2297856] via 172.88.0.10, 00:01:38, Serial0/0/0.256
                       [90/2297856] via 172.88.0.6, 00:01:38, Serial0/0/1.642
                       [90/2297856] via 172.88.0.2, 00:01:38, Serial0/0/0.641
     186.12.0.0/30 is subnetted, 1 subnets
C       186.12.0.0 is directly connected, Serial0/0/1.512
D    192.168.224.0/24 [90/2297856] via 172.88.0.10, 00:01:39, Serial0/0/0.256
                      [90/2297856] via 172.88.0.6, 00:01:39, Serial0/0/1.642
                      [90/2297856] via 172.88.0.2, 00:01:39, Serial0/0/0.641
D    192.168.225.0/24 [90/2297856] via 172.88.0.10, 00:01:39, Serial0/0/0.256
                      [90/2297856] via 172.88.0.6, 00:01:39, Serial0/0/1.642
                      [90/2297856] via 172.88.0.2, 00:01:41, Serial0/0/0.641
     172.88.0.0/30 is subnetted, 3 subnets
C       172.88.0.4 is directly connected, Serial0/0/1.642
```

```
C       172.88.0.0 is directly connected, Serial0/0/0.641
C       172.88.0.8 is directly connected, Serial0/0/0.256
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.16.0 is directly connected, FastEthernet0/1
     10.0.0.0/30 is subnetted, 1 subnets
C       10.28.0.0 is directly connected, Serial0/0/1.768
C    192.168.16.0/24 is directly connected, FastEthernet0/0
D    192.168.100.0/24 [90/28416] via 192.168.16.6, 00:01:41, FastEthernet0/0
                      [90/28416] via 192.168.16.5, 00:01:41, FastEthernet0/0
```

Each of the networks learnt via the frame-relay network have the same distance. This is not good due to the fact that each of the links has a different bandwidth!

Step 3

Configure the interfaces correctly to reflect the real bandwidth of the links (note the sub interface ID is the bandwidth of the link, this is also shown in the diagram above):

```
NYWAN1(config)#interface serial 0/0/0.641
NYWAN1(config-if)#bandwidth 64
NYWAN1(config-if)#interface serial 0/0/0.256
NYWAN1(config-if)#bandwidth 256
NYWAN1(config-if)#interface serial 0/0/1.642
NYWAN1(config-if)#bandwidth 64
```

Now your routing table looks like this:

```
NYWAN1#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
```

```
     192.168.226.0/27 is subnetted, 2 subnets
D       192.168.226.0 [90/10639872] via 172.88.0.10, 00:00:06, Serial0/0/0.256
D       192.168.226.32
           [90/10639872] via 172.88.0.10, 00:00:06, Serial0/0/0.256
     186.12.0.0/30 is subnetted, 1 subnets
C       186.12.0.0 is directly connected, Serial0/0/1.512
D    192.168.224.0/24 [90/10639872] via 172.88.0.10, 00:00:06, Serial0/0/0.256
D    192.168.225.0/24 [90/10639872] via 172.88.0.10, 00:00:06, Serial0/0/0.256
     172.88.0.0/30 is subnetted, 3 subnets
C       172.88.0.4 is directly connected, Serial0/0/1.642
C       172.88.0.0 is directly connected, Serial0/0/0.641
C       172.88.0.8 is directly connected, Serial0/0/0.256
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.16.0 is directly connected, FastEthernet0/1
     10.0.0.0/30 is subnetted, 1 subnets
C       10.28.0.0 is directly connected, Serial0/0/1.768
C    192.168.16.0/24 is directly connected, FastEthernet0/0
D    192.168.100.0/24 [90/28416] via 192.168.16.6, 00:00:09, FastEthernet0/0
                      [90/28416] via 192.168.16.5, 00:00:09, FastEthernet0/0
```

Step 4

Finally tune EIGRP so that on the 64Kb links, only 50% of the bandwidth can be used for the EIGRP process:

```
NYWAN1(config)#interface serial 0/0/0.641
NYWAN1(config-subif)#ip bandwidth-percent eigrp 800 50
NYWAN1(config-subif)#interface serial 0/0/1.642
NYWAN1(config-subif)#ip bandwidth-percent eigrp 800 50
```

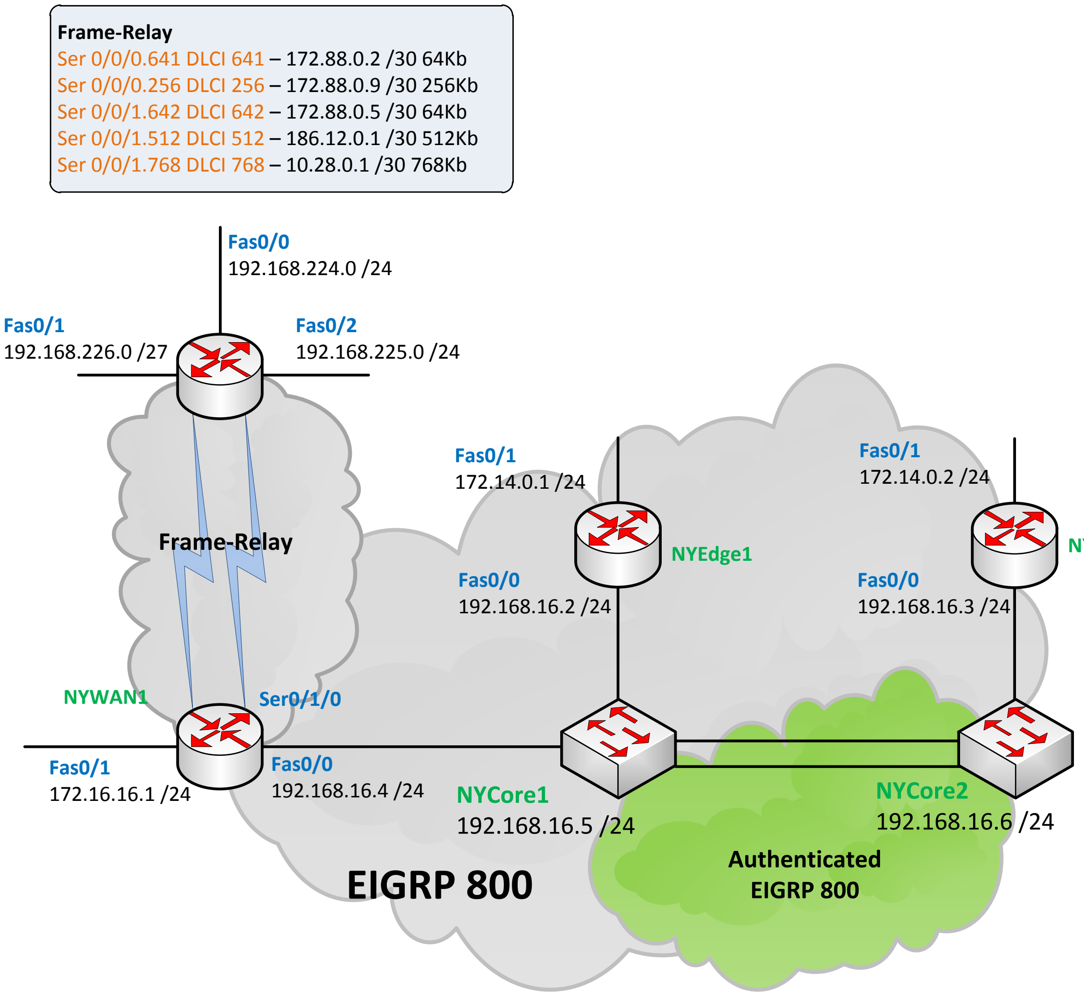
The 64kb links are serial 0/0/0.641 and 0/0/1.642.

Leave your devices in their current states and continue to the next exercise.

Exercise 4 – Tuning EIGRP

In this exercise you will learn how to tune some parameters of the EIGRP routing protocol. Please refer to your course material or use your preferred search engine to gain an understanding of these tasks.

Lab Diagram



Offset lists

Offset lists enable the modification of a routes distance either inbound or outbound. For example say you had two routes with equal cost paths and you wanted to prefer one route over another you could use offset lists to modify the outgoing distance on your non-preferred route.

If you look at the route on NYWAN for 192.168.100.0/24 you will see there are multiple paths via NYCORE1 and NYCORE2. Even though NYWAN1 and NYCORE1 are directly connected to each other, because NYCORE2 is advertising on the same Layer2 (VLAN) network it does not go through any additional router to get to NYWAN1 meaning its distance doesn't change.

```
NYEDGE1#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
     192.168.226.0/27 is subnetted, 2 subnets
D       192.168.226.0
           [90/10642432] via 192.168.16.4, 00:14:06, FastEthernet0/0
D       192.168.226.32
           [90/10642432] via 192.168.16.4, 00:14:06, FastEthernet0/0
D    192.168.224.0/24
           [90/10642432] via 192.168.16.4, 00:14:06, FastEthernet0/0
D    192.168.225.0/24
           [90/10642432] via 192.168.16.4, 00:14:06, FastEthernet0/0
     172.88.0.0/30 is subnetted, 3 subnets
D       172.88.0.4 [90/40514560] via 192.168.16.4, 00:14:16, FastEthernet0/0
D       172.88.0.0 [90/40514560] via 192.168.16.4, 00:14:18, FastEthernet0/0
D       172.88.0.8 [90/10514432] via 192.168.16.4, 00:14:18, FastEthernet0/0
     172.14.0.0/24 is subnetted, 1 subnets
C       172.14.0.0 is directly connected, FastEthernet0/1
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.16.0 [90/30720] via 192.168.16.4, 00:21:12, FastEthernet0/0
C    192.168.16.0/24 is directly connected, FastEthernet0/0
D    192.168.100.0/24 [90/28416] via 192.168.16.6, 00:21:12, FastEthernet0/0
                      [90/28416] via 192.168.16.5, 00:21:12, FastEthernet0/0
```

Note: Distance calculated is **28416**

You could choose to modify the bandwidth or delay statement of NYCORE2 to change this but this may affect other routes from this device.

Another approach is to use an offset list.

Follow these steps to modify the distance for the 192.168.100.0/24 subnet on NYCORE2.

Step 1

Use the following commands to create an access-list to match the 192.168.100.0/24 subnet and apply an outgoing offset list:

```
NYCORE2(config)#access-list 10 permit 192.168.100.0 0.0.0.255

NYCORE2(config)#router eigrp 800

NYCORE2(config-router)#offset-list 10 out 1000
```

Notice the log message generated on the other peers when you apply the offset-list!

```
*May 22 13:27:06: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 192.168.16.6
(FastEthernet0/0) is resync: peer graceful-restart
```

Viewing the routing table on NYWAN1 post configuration:

```
NYWAN1#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

     192.168.226.0/27 is subnetted, 2 subnets

D       192.168.226.0 [90/10639872] via 172.88.0.10, 00:16:38, Serial0/0/0.256

D       192.168.226.32

           [90/10639872] via 172.88.0.10, 00:16:38, Serial0/0/0.256

     186.12.0.0/30 is subnetted, 1 subnets

C       186.12.0.0 is directly connected, Serial0/0/1.512

D    192.168.224.0/24 [90/10639872] via 172.88.0.10, 00:16:38, Serial0/0/0.256

D    192.168.225.0/24 [90/10639872] via 172.88.0.10, 00:16:38, Serial0/0/0.256

     172.88.0.0/30 is subnetted, 3 subnets

C       172.88.0.4 is directly connected, Serial0/0/1.642

C       172.88.0.0 is directly connected, Serial0/0/0.641

C       172.88.0.8 is directly connected, Serial0/0/0.256

     172.16.0.0/24 is subnetted, 1 subnets
```

```
C       172.16.16.0 is directly connected, FastEthernet0/1
     10.0.0.0/30 is subnetted, 1 subnets
C       10.28.0.0 is directly connected, Serial0/0/1.768
C    192.168.16.0/24 is directly connected, FastEthernet0/0
D    192.168.100.0/24 [90/28416] via 192.168.16.5, 00:00:30, FastEthernet0/0
```

The route via NYCORE2 has dropped out of the routing table!

Step 2

Viewing the eigrp topology entries on NYWAN1 for the 192.168.100.0 /24 subnet reveals that the route is still viable via NYCORE2 and its metric has changed:

```
NYWAN1#sho ip eigrp topology 192.168.100.0
IP-EIGRP (AS 800): Topology entry for 192.168.100.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 28416
  Routing Descriptor Blocks:
  192.168.16.5 (FastEthernet0/0), from 192.168.16.5, Send flag is 0x0
      Composite metric is (28416/2816), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 110 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  192.168.16.6 (FastEthernet0/0), from 192.168.16.6, Send flag is 0x0
      Composite metric is (29416/3816), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 149 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

Variance

Variance is used to allow unequal cost load balancing. For example in the lab there is a frame-relay link that has 4 paths to the same subnets. In the previous exercise you changed the bandwidth configuration to reflect the correct speed of the links which meant that the lower bandwidth links dropped out of the routing table.

You may however wish to use all or a number of these links to load balance across them, you're paying for them so why waste them right!

One word of caution here however, in the real world I have seen load balancing of links cause some issues when the latency across the links vary. For example I have seen WAN circuits that go between countries in two different directions and the latencies vary on each of the links by over 80ms. Using EIGRP load balancing (in a per-packet fashion) cause's big problems for applications as packets arrive out of sequence (imagine a voice conversation!) or if on a per session basis some applications can end up being slower than others.

In the lab scenario however the links do not have any latency differences and CEF will load balance the packets accordingly.

To configure variance, use the following steps:

Step 1

View the topology of a route coming through the serial interface:

```
NYWAN1#show ip eigrp topology 192.168.224.0

IP-EIGRP (AS 800): Topology entry for 192.168.224.0/24

  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 10639872

  Routing Descriptor Blocks:

  172.88.0.10 (Serial0/0/0.256), from 172.88.0.10, Send flag is 0x0

      Composite metric is (10639872/128256), Route is Internal

      Vector metric:

        Minimum bandwidth is 256 Kbit

        Total delay is 25000 microseconds

        Reliability is 255/255

        Load is 1/255

        Minimum MTU is 1500

        Hop count is 1

  172.88.0.2 (Serial0/0/0.641), from 172.88.0.2, 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
  172.88.0.6 (Serial0/0/1.642), from 172.88.0.6, 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
```

We can see that the route for 192.168.224.0 has 3 paths. However, 2 of those paths have the same metric, so we need to artificially modify this for the purpose of getting variance working.

Change the bandwidth on serial 0/0/0.641 to be 192Kbs:

```
NYWAN1(config)#interface serial 0/0/0.641
NYWAN1(config-if)#bandwidth 192
```

Next, let's say that we want to install the route in to the routing table for both the 256Kb link and our new 192Kb link. The metrics for both routes respectively are:

256Kb = 10639872

192Kb = 13973248

Step 2

To install both of these routes we can use the variance command as follows:

```
NYWAN1(config)#router eigrp 800
NYWAN1(config-router)#variance 2
```

Step 3

Viewing the routing table now shows (note that in the command in the screenshot, the ser is case sensitive, it should be **Ser**):

```
NYWAN1#show ip route | inc Ser

D       192.168.226.0 [90/10639872] via 172.88.0.10, 00:00:18, Serial0/0/0.256

                      [90/13973248] via 172.88.0.2, 00:00:18, Serial0/0/0.641

           [90/10639872] via 172.88.0.10, 00:00:18, Serial0/0/0.256

           [90/13973248] via 172.88.0.2, 00:00:18, Serial0/0/0.641

C       186.12.0.0 is directly connected, Serial0/0/1.512

D    192.168.224.0/24 [90/10639872] via 172.88.0.10, 00:00:18, Serial0/0/0.256

                      [90/13973248] via 172.88.0.2, 00:00:18, Serial0/0/0.641

D    192.168.225.0/24 [90/10639872] via 172.88.0.10, 00:00:18, Serial0/0/0.256

                      [90/13973248] via 172.88.0.2, 00:00:18, Serial0/0/0.641

C       172.88.0.4 is directly connected, Serial0/0/1.642

C       172.88.0.0 is directly connected, Serial0/0/0.641

C       172.88.0.8 is directly connected, Serial0/0/0.256

C       10.28.0.0 is directly connected, Serial0/0/1.768
```

Notice the routes that are available via both Serial 0/0/0.256 and Serial 0/0/0.641

Step 4

If CEF was not enabled and the processor was switching each packet, we would find that for every two packets sent down Serial 0/0/0.256 1 packet would be sent down 0/0/0.641

Note: Use your preferred search engine to find out how does load balancing work.

However, CEF is enabled so to view which link a session would go down, you can use the **show ip cef exact-route** command, for example:

```
NYWAN1#show ip cef exact-route 192.168.100.2 192.168.224.5

192.168.100.2   -> 192.168.224.5  : Serial0/0/0.256 (next hop 172.88.0.10)

NYWAN1#show ip cef exact-route 192.168.100.19 192.168.224.23

192.168.100.19  -> 192.168.224.23 : Serial0/0/0.641 (next hop 172.88.0.2)
```

Note: You may have to try different combinations of addresses in the two subnets to receive find output for both links.

Hello and hold times

Hello and hold timers are used to verify that a nieghbor still exists. A router usually knows if an interface fails as it goes in to an up/down or down/down state, however in some instances this does may happen (some WAN links are a good example of this), therefore EIGRP needs another method to confirm a neighbors reachability.

The Hello timer forces EIGRP to send a periodic hello message out of its interfaces (the default is 5 seconds on links over 1.544Mb and 60 seconds otherwise). The hold timer then sets a specific amount of time before the nighbor is considered down.

For example, the default LAN hello time is 5 seconds with a hold time of 15, which means if EIGRP does not hear any response from a peer after 3 hellos it is considered down, and for slow WAN links it's 60 * 3 =180 seconds.

Adjusting these timers tell remote neighbors how long to consider the sender valid.

Use the following commands to configure these timers:

Note: I would not recommend you adjust these in a live environment without consulting Cisco.

Step 1

On the NYWAN1 router change the timers for hello's to be 2 seconds and the hold timer to be 6 seconds on the LAN interface:

```
NYWAN1(config)#interface fastethernet 0/0

NYWAN1(config-if)#ip hello-interval eigrp 800 2

NYWAN1(config-if)#ip hold-time eigrp 800 6
```

Step 2

To confirm your settings use the **show ip eigrp interface detail** command:

```
NYWAN1#show ip eigrp interface detail fastethernet 0/0

IP-EIGRP interfaces for process 800

                        Xmit Queue   Mean   Pacing Time   Multicast    Pending

Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes

Fa0/0              4        0/0        29       0/1          144           0

  Hello interval is 2 sec

  Next xmit serial

  Un/reliable mcasts: 0/12  Un/reliable ucasts: 50/32

  Mcast exceptions: 7  CR packets: 4  ACKs suppressed: 0

  Retransmissions sent: 6  Out-of-sequence rcvd: 0

  Authentication mode is not set
```

```
  Use multicast
```

Step 3

To view the timers in action, make sure you have a connection to one of the other neighbors (for example NYEDGE1) and then shut down the interface fas 0/0 on NYWAN1 and be ready to count before seeing the neighbor relationship drop:

```
NYWAN1(config)#interface fastethernet 0/0

NYWAN1(config-if)#shutdown
```

You should notice that the neighbor relationship drops within 6 seconds:

```
*May 22 13:24:06: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 192.168.16.4
(FastEthernet0/0) is down: holding time expired
```

Bring up the interface again.

K Values

As with the hello and hold timers, the K values shouldn't really be adjusted without consulting Cisco, however in this next exercise you will configure these values.

Step 1

On NYWAN1 modify the K values to these arbitrary values:

0 100 125 150 175 200

```
NYWAN1(config)#router eigrp 800

NYWAN1(config-router)#metric weights 0 100 125 150 175 200
```

You will notice that all your neighbor relationships will drop because of the mismatch in k values. You would need to configure this on each router to re-enable the relationships.

```
NYWAN1(config-if)#router eigrp 800

NYWAN1(config-router)#metric weights 0 100 125 150 175 200

NYWAN1(config-router)#

*May 22 13:38:59: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.10
(Serial0/0/0.256) is down: metric changed

*May 22 13:38:59: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.2
(Serial0/0/0.641) is down: metric changed

*May 22 13:38:59: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.6
(Serial0/0/1.642) is down: metric changed

*May 22 13:39:01: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 800: Neighbor 172.88.0.6
(Serial0/0/1.642) is down: K-value mismatch
```

Remove this statement:

```
NYWAN1(config-router)#no metric weights 0 100 125 150 175 200
```

Your relationships will reform.

Summary

In this module you achieved the following activities:

- You configured basic EIGRP.
- You configured EIGRP authentication.
- You tuned paths and links using offset lists and bandwidth statements.
- You modified the K values and saw how this needs to be reflected on all your devices.

Also Try

Using your lab infrastructure you can attempt the following topics at your own pace, these are additional tasks that can be done building on what you have learnt in this module:

- Add a second key chain for authentication on NYCORE1 and NYCORE2.
- Modify the K values on the LAN based EIGRP neighbors.