

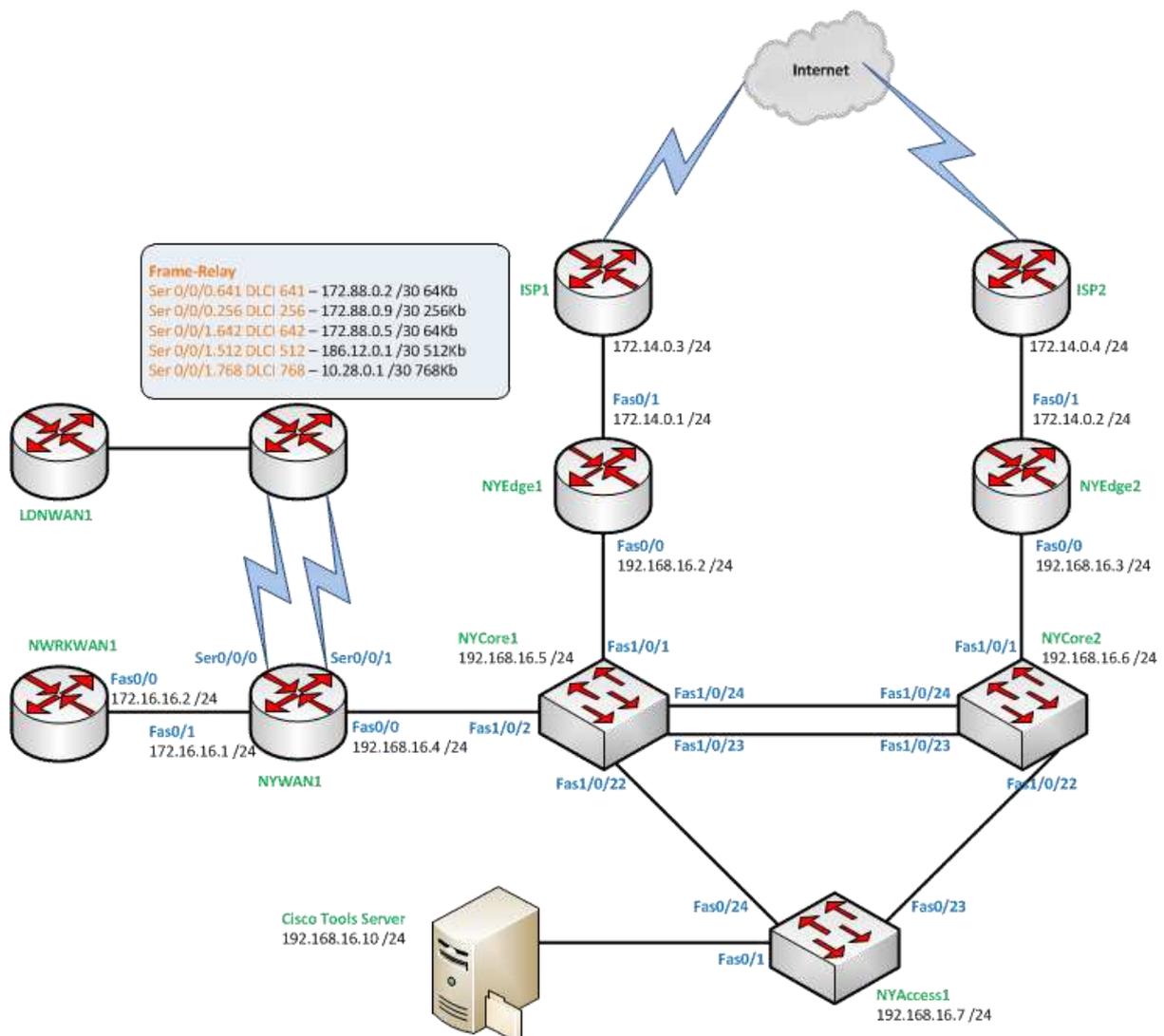
Introduction

The **Troubleshooting routing protocols** module provides you with the instructions and Cisco hardware to develop your hands on skills in troubleshooting routing protocols, specifically EIGRP. This module includes the following exercises:

- 1) Troubleshooting 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.

- NYWAN1

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

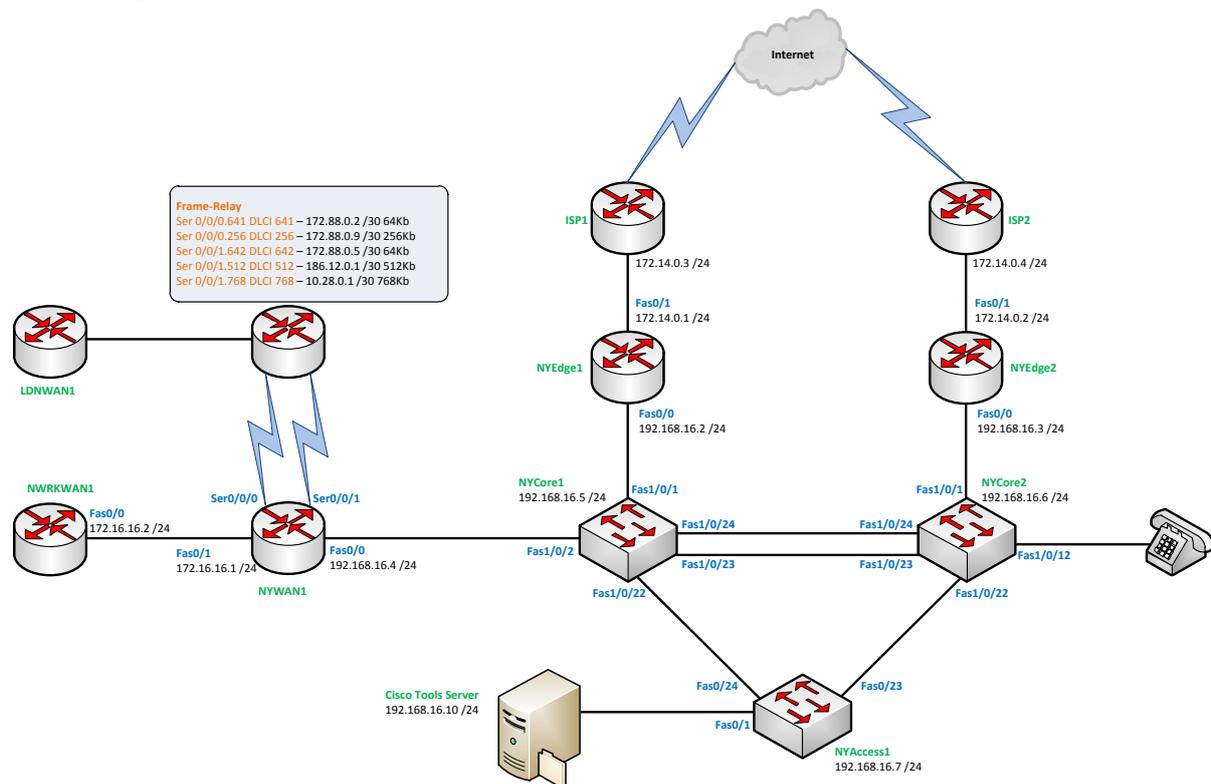
This document and its content is copyright of Practice-IT - © Practice-IT 2013. All rights reserved. Any redistribution or reproduction of part or all of the contents in any form is prohibited other than the following:

- 1) You may print or download to a local hard disk extracts for your personal and non-commercial use only.
- 2) You may copy the content to individual third parties for their personal use, but only if you acknowledge the website as the source of the material. You may not, except with our express written permission, distribute or commercially exploit the content. Nor may you transmit it or store it in any other website or other form of electronic retrieval system.

Exercise 1 – Troubleshooting EIGRP

In this exercise you will learn some troubleshooting techniques for 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



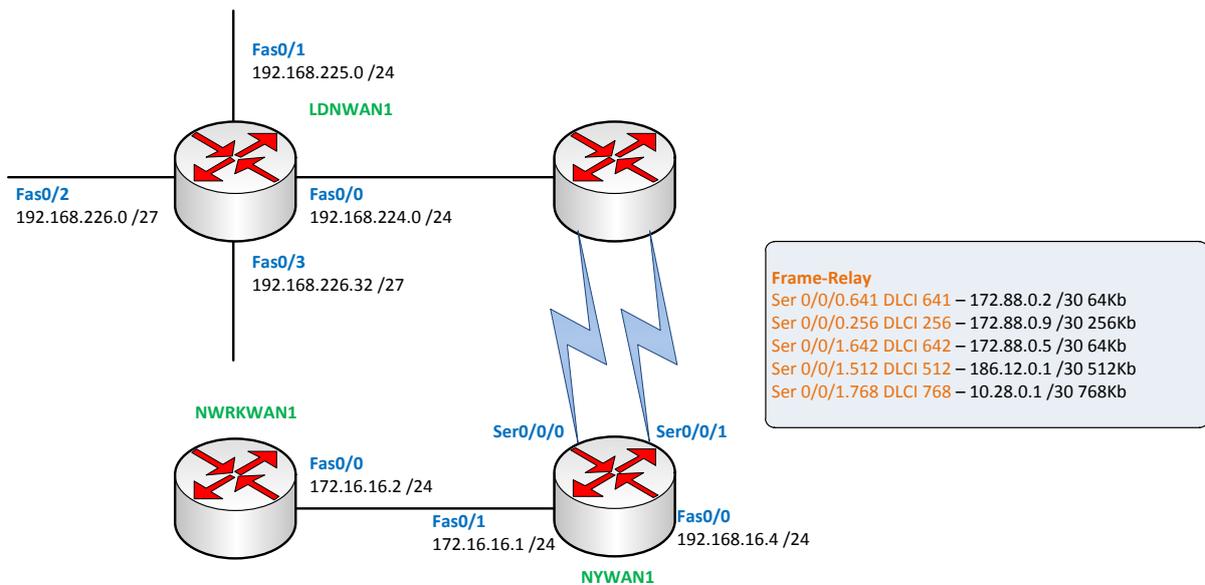
Problem definition

You have a frame-relay network connecting your wide area network, with Internet access and a small Newark office that is separated from this environment. There are a number of subnets in the London (LDNWAN1) office that you need to get access to as they host some critical services that your office staff relies on.

Recently you wanted to upgrade the bandwidth to the office, but your service provider recommended getting a separate frame-relay PVC to the site as they can divert this through some secondary infrastructure, whilst some single points of failure still exist they recommended this approach as it does remove some key failure points.

Your service provider informed you that you could aggregate the bandwidth of the two links, but for some reason you don't see any performance increase, even though you have configured the link the same as the previous connection.

The network topology is as follows:



Troubleshooting commands

In this task you will learn how to use the following commands to troubleshoot a network related problem:

- show ip route
- show ip cef
- show adjacency detail
- show ip arp
- show ip cef exact-route
- show frame-relay map
- show ip eigrp interfaces
- show ip eigrp topology

Step 1

The first step you take is to view the routing table. We'll first view the full routing table using the **show ip route** command on 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
```

```
192.168.226.0/27 is subnetted, 2 subnets
```

```

D      192.168.226.0 [90/10639872] via 172.88.0.10, 00:03:02, Serial0/0/0.256
D      192.168.226.32
      [90/10639872] via 172.88.0.10, 00:03:02, 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, 00:03:02, Serial0/0/0.256
D      192.168.225.0/24 [90/10639872] via 172.88.0.10, 00:03:02, 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

```

Step 2

As the routing table has a number of routes, you weren't able to clearly see which routes to focus on to see where the problem may be so you decide to use a more specific command to focus on a particular route, in this instance you choose to view the 192.168.224.0 network address which is one of the subnets in London:

```

NYWAN1#show ip route 192.168.224.0
Routing entry for 192.168.224.0/24
  Known via "eigrp 800", distance 90, metric 10639872, type internal
  Redistributing via eigrp 800
  Last update from 172.88.0.10 on Serial0/0/0.256, 00:04:31 ago
  Routing Descriptor Blocks:
  * 172.88.0.10, from 172.88.0.10, 00:04:31 ago, via Serial0/0/0.256
    Route metric is 10639872, traffic share count is 1
    Total delay is 25000 microseconds, minimum bandwidth is 256 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1

```

From the output of this command you can see that the route is being learnt from the peer on the interface Serial 0/0/0.256

Step 3

Using the frame-relay map command you notice that Serial 0/0/0.256 is **dlci 256** which your service provider told you was the higher bandwidth of the two links, this has a 256kb link whereas the alternate link is just 64kb.

```
NYWAN1#show frame-relay map
Serial0/0/0.256 (up): point-to-point dlci, dlci 256(0x100,0x4000), broadcast
                        status defined, active
```

Step 4

Your colleague has read somewhere about the CEF and tells you that you should make sure the router has the route in CEF as maybe the CEF is corrupt somehow. He's slightly more senior than you so you amuse him by checking to see if the route exists in CEF:

```
NYWAN1#show ip cef 192.168.224.0
192.168.224.0/24, version 22, epoch 0, cached adjacency to Serial0/0/0.256
0 packets, 0 bytes
  via 172.88.0.10, Serial0/0/0.256, 0 dependencies
    next hop 172.88.0.10, Serial0/0/0.256
    valid cached adjacency
```

Step 5

You show your colleague the output from the **show ip cef** command and he informs you that instead you should use the show ip cef exact-route command:

```
NYWAN1#show ip cef exact-route 192.168.16.0 192.168.224.0
192.168.16.0    -> 192.168.224.0  : Serial0/0/0.256 (next hop 172.88.0.10)
```

You can see from the output that the router is using the path via **Serial 0/0/0.256**, these were interesting commands to note from your colleague although you are not sure he was really telling the truth about the corruption thing.

Step 6

You decide to confirm whether the router is even hearing advertisements from EIGRP on the old link so you use the following commands to make sure you have EIGRP configured on the correct interfaces, a neighbor relationship exists for the remote device on DLCI 256 and that the route is known by EIGRP across this link. You use the following commands:

```
NYWAN1#show ip eigrp interfaces
NYWAN1#show ip eigrp neighbors
```

```
NYWAN1#show ip eigrp topology 10.88.16.0 255.255.255.224
```

Command output from **show ip eigrp interfaces** on NYWAN1:

```
NYWAN1#show ip eigrp interfaces
```

```
IP-EIGRP interfaces for process 800
```

Interface	Peers	Xmit Queue Un/Reliable	Mean SRTT	Pacing Time Un/Reliable	Multicast Flow Timer	Pending Routes
Fa0/0	4	0/0	1101	0/1	3384	0
Se0/0/0.256	1	0/0	385	2/95	2015	0
Se0/0/0.641	1	0/0	324	10/380	1996	0
Se0/0/1.642	1	0/0	990	10/380	5180	0

The **show ip eigrp interfaces** command shows you that in fact EIGRP is running on **Serial 0/0/0.641** and **Serial 0/0/0.256**. This is a good sign.

The **show ip eigrp neighbors** command confirms that you definitely have a peer relationship across the Serial 0/0/0.641 link, this is an even more positive sign!

Command output from **show ip eigrp neighbors** on NYWAN1:

```
NYWAN1#show ip eigrp neighbors
```

```
IP-EIGRP neighbors for process 800
```

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q Cnt	Seq Num
6	192.168.16.3	Fa0/0	11 00:08:54	1560	5000	0	13
5	192.168.16.6	Fa0/0	11 00:08:54	604	3624	0	9
4	192.168.16.5	Fa0/0	11 00:09:18	1563	5000	0	5
3	172.88.0.6	Se0/0/1.642	10 00:09:24	990	5000	0	90
2	192.168.16.2	Fa0/0	11 00:09:47	677	4062	0	7
1	172.88.0.2	Se0/0/0.641	12 00:10:00	324	2280	0	89
0	172.88.0.10	Se0/0/0.256	13 00:10:00	385	2310	0	88

The **show ip eigrp topology 192.168.224.0 255.255.255.0** command actually proves that you are receiving the route on all the relevant interfaces, this is great news!

```
NYWAN1#show ip eigrp topology 192.168.224.0 255.255.255.0
```

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

172.88.0.6 (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

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

Step 7

You then use your favourite course provider to research more about **EIGRP load balancing**. You find some interesting content about the **variance** command in EIGRP.

You realise that the metric of 10639872 (the first topology entry) is just over a quarter 40640000 (the last entry) so you decide to put in a change control request to implement the command **variance 4** under the EIGRP process.

Step 8

After your change control has been approved, you implement the command:

```
NYWAN1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
NYWAN1(config)#router eigrp 800
NYWAN1(config-router)#variance 4
```

Confirming the fix

After having implemented the command your change control states that you must prove the command has indeed fixed the problem.

You use the following commands to confirm that the network will indeed be load balanced across the links:

```
show ip route
```

Command output for 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

192.168.226.0/27 is subnetted, 2 subnets
D       192.168.226.0 [90/10639872] via 172.88.0.10, 00:00:47, Serial0/0/0.256
        [90/40640000] via 172.88.0.6, 00:00:47, Serial0/0/1.642
```

```

[90/40640000] via 172.88.0.2, 00:00:47, Serial0/0/0.641
D    192.168.226.32
      [90/10639872] via 172.88.0.10, 00:00:47, Serial0/0/0.256
      [90/40640000] via 172.88.0.6, 00:00:47, Serial0/0/1.642
      [90/40640000] via 172.88.0.2, 00:00:47, 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 [90/10639872] via 172.88.0.10, 00:00:48, Serial0/0/0.256
      [90/40640000] via 172.88.0.6, 00:00:48, Serial0/0/1.642
      [90/40640000] via 172.88.0.2, 00:00:48, Serial0/0/0.641
D    192.168.225.0/24 [90/10639872] via 172.88.0.10, 00:00:49, Serial0/0/0.256
      [90/40640000] via 172.88.0.6, 00:00:49, Serial0/0/1.642
      [90/40640000] via 172.88.0.2, 00:00:49, 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

```

Notice how there is now multiple entries in the routing table, compare this output with the output at the beginning of the exercise.

The next commands may not reproduce the exact results, if not, try different combinations of the source subnet and destination subnet. For example, you don't need to stick with 192.168.16.10, you can change this to any address in the subnet, the same with 192.168.224.1,2,3.

```

NYWAN1#show ip cef exact-route 192.168.16.10 192.168.224.1
192.168.16.10  -> 192.168.224.1  : Serial0/0/1.642 (next hop 172.88.0.6)
NYWAN1#show ip cef exact-route 192.168.16.10 192.168.224.2
192.168.16.10  -> 192.168.224.2  : Serial0/0/0.256 (next hop 172.88.0.10)
NYWAN1#show ip cef exact-route 192.168.16.10 192.168.224.3

```

```
192.168.16.10 -> 192.168.224.3 : Serial10/0/0.256 (next hop 172.88.0.10)
```

Note: Notice the output is different for the first iteration of the **show ip cef exact-route** command, why do you think this is?

Summary

In this module you achieved the following activities:

- You used a number of commands to verify that your underlying layer 3 network is operational.
- You used various EIGRP related commands to verify the network topology.
- You learnt how to view the EIGRP topology table.
- You learnt how to modify EIGRP so that you can use unequal load balancing.

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:

View the output on NYWAN1 of the following commands:

- `show adjacency detail`
- `debug eigrp packets hello`
- `debug ip eigrp neighbor 800`