# EIGRP LAB

# 

# harsha Bhat

# CCNP 6/7

# Mr. Mason

The purpose of this lab is to configure EIGRP with unequal cost load balancing and adjusted EIGRP metrics on 5 Cisco 4321 routers connected to one another in a linear daisy-chain topology (above).

Background Information\*

Enhanced Interior Gateway Routing Protocol or EIGRP is a Cisco Proprietary Routing protocol. Unlike RIP which uses hop count, and OSPF which uses cost, EIGRP uses bandwidth, delays, load, and reliability\* as metrics to choose the best path. Rather than sending data out a link and forgetting about it later, it saves data from its neighbors’ advertisements to become aware of the network topology, and can forward this information to other routers, creating an efficient, loop-free path. If a router goes down, other routers can always consult the topology to determine a new best path.

This basic setup used to be called IGRP, until it was enhanced to support IPv4 addresses. **<** *The addition of the “E” in the protocol contributed to an abundance of jokes referencing the children’s nursery rhyme “Old MacDonald Had a Farm” in various online tutorials and websites. In my local Cisco lab, this was touched upon with a farm-themed book cover of “Everything I Want to Do is Illegal”; several students agreed the title applied to the lab itself as well as their personal lives. For completely unrelated reasons, Mr. Mason has stated EIGRP is his favorite routing protocol.* ***>***

To calculate the best path, EIGRP uses the cost equation

256\*((K1\*Bw) + (K2\*Bw)/(256 – Load)

+ (K3\*Delay)\*(K5/(Reliability + K4)))

dependent primarily on its metrics but also factoring manual input (K1, K2, etc.). By default, EIGRP considers only 2 of the 4 metrics allowed to (the 5th, MTU, is not used at all) factor into the cost, as the K-values K1 (affecting bandwidth) and K3 (affecting delay) are set to 1 while the others are all set to 0. However, each K-value can manually be changed to a value from 0-255. This is done with the command

**metric weights 0 <K1> <K2> <K3> <K4> <K5> under**

**router eigrp 1.**

A variation on this cost equation and command designed to deal with greater bandwidths and shorter delays, “EIGRP wide metrics”, is discussed later.\*

\*see “Lab Summary”

Another special feature of EIGRP is unequal cost load balancing via variance. If there are multiple routes to a particular address, the

**show ip eigrp topology <address>**

command will display each route and its composite metric. Of these routes, the one with the lowest metric will be a successor while others will be

feasible successors. One can then compute the quotient of the feasible successor and successor, and use the

**show** **variance x command under**

**router eigrp 1**

where **x** is greater than the quotient. This will make the faster link handle more traffic than the slower one, hence the term “unequal cost”. To make the slower link handle more traffic, set x to less than the quotient. The variance command is especially useful to allow a much slower link to pace with an extremely fast link.

## To summarize, EIGRP sends updates only when needed, is loop-free, automatically clears bad routes and is fault-tolerant.

Lab Summary *With more complex background elements discussed*

Initially, I set up EIGRP on 3 routers following Cisco ipv6-specific guidelines and added my own ipv4 configs based on the efficient system demonstrated in the previous lab. The Cisco system ended up being unwieldy as each address was radically different from the others, e.g. they had different subnets and the notation was off for the loopbacks. After getting it working for those 3 routers, I extended the topology to 5 routers (making use of past lab experience to make the router configs standardized, cut and paste rather than typing in commands 1 by 1). I struggled a bit with the network statements, initially thinking I could do interface-configs as I had done with OSPF. After EIGRP was running on all 5 routers, I turned my attention to various other bonus features of EIGRP, namely: metric weights, variance, “wide” metrics, 2 autonomous systems (separate instances of EIGRP) on the same router, and external EIGRP routes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| METRIC: | **342632.6459** | **384.5019608** | **0** | **3223673239** | **76288.0994** |
| K1 | 2 | 2 | 1 | 200 | 200 |
| K2 | 1 | 1 | 0 | 250 | 250 |
| K3 | 1 | 1 | 1 | 100 | 100 |
| K4 | 1 | 1 | 0 | 100 | 100 |
| K5 | 1 | 1 | 0 | 100 | 100 |
| Bandwidth | 1000000 | 1 | 1000000 | 1000000 | 1 |
| Load | 1 | 1 | 1 | 0.003921569 | 1 |
| Delay | 5030 | 1 | 5030 | 5030 | 1 |
| Reliability | 1500 | 1 | 1 | 1500 | 1 |

In an excel table linked to the cost equation, I experimented with the effects of metrics to better understand them.

Bandwidth is the maximum capacity of the route. It’s multiplied by 10 in the cost equation to make scaled bandwidth.

Load is the percentage of the bandwidth being used by incoming traffic.

Delay (also called latency) is the number of microseconds it takes for a packet to reach the end of a link. It’s multiplied by 10 in the cost equation to make scaled delay.

Reliability is the percent chance of a packet being routed correctly.

Of the above metrics, Bandwidth and Delay are constantly updated while Load and Reliability are calculated when the path is first learned.

MTU or maximum transmission unit is the size of the largest piece of data sent on the path.

While online tutorials and other people’s configurations had values like those in row 1, column 2 (rounded up); in my lab and similar experiments I experienced values similar to row 1, column 3. I also looked at the default values for the metrics themselves. For example, the default value for delay is 5000, but it goes up by 10 every link it traverses within the topology. I found the Cisco default values for metrics had little effect on the overall cost, and only with some tweaking could optimal load balancing be achieved. I chose the metric weight configuration 0 2 1 1 1 1 to fulfill the requirement to use all K-values to influence routes. All metrics are used except for MTU, and I gave the bandwidth a greater priority as I would later adjust it.

With this advanced understanding of metric weights, I configured my network to have an interface with bandwidth 200000 (1/5 of the default 1000000 kbit (1 gigabit) value of a gigabitEthernet interface). The results of this can be seen in the figure under “R4: Proof of unequal cost load balancing” in the configuration section. The 192.168.3.0 network is 3 times faster than the 172.16.1.0 network, and thus is taking care of ¾ of the total route. Given that the 192.168.3.0 network has 5 times the bandwidth, I should have adjusted the bandwidth metric a bit more in my initial configuration (e.g. 0 3 1 1 1 1) to facilitate a better ratio.

While I had configured variance several months before I had proof it worked, I somewhat misunderstood the concept and didn’t think I configured variance correctly. I thought I had to use variance solely for load balancing, and wasn’t aware of the capacity for the metrics to do so. While sorting out issues with metrics, I tried to set the variance very high but this prevented the 172.16.1.0 network from operating under EIGRP (see the end of the problems section). When I had Mr. Mason explain to me what was happening, I finally understood this phenomenon: variance is used to “qualify” a route, to decide whether it’s allowed to assist in load balancing. Taking Mr. Mason’s challenge to make the slower link handle more traffic, I set the variance in my topology to 1, though I think with the low bandwidth on the 172.16.1.0 network it had little effect.

While trying to configure variance, I tried to set up multiple autonomous systems and used external routes. While I was aware of the concept of successors and feasible successors, I didn’t think they had to be in the same area or anything, so I tried whatever method possible to get a feasible successor. I set up an instance of router eigrp 2 exclusively on the 172.16.1.0 network and measured its traffic (see configs). Surprisingly, the systems were able to coexist, and pings could go from eirgp 1 through eirgp 2 to eirgp 1 again. A difference with external routes is they have direct metrics, as in the values you enter directly correlate with the values in the cost equation as opposed to being k-values. This also allows you to affect previously untainted factors such as MTU. When setting up initial external routes, I didn’t know the default MTU value (1500) and accidentally set it to 1, however the external routes performed well with no documented issues. They would later come in handy for the BGP lab.

After configuring metric weights, I tried my hand at a new metric cost formula, “wide metrics”. While default EIGRP conventions are 32-bit, wide metrics supports 64-bit ones. It can support bandwidths greater than 1 GB and has delay measured in picoseconds instead of tens of microseconds. Since the original k-values can’t be modified to this format, it multiplies each one by the constant 65536, *DOESN’T* scale them (bandwidth and delay are not multiplied by 10), and uses a new cost equation:

[(K1\*Bandwidth + {K2\*Minimum Throughput} / 256-Load) + (K3\*Total Latency) + (K6\*Extended Attributes)]\* [K5/(K4 + Reliability)]

K6 is yet another K-value influencing “Jitter” (difference in packet delivery time), “Energy” (active power usage of a path), and “Quiescent Energy” (idle power usage of a path), all of which I could find very little information on. I used configuration 0 2 1 1 1 1 0, not using K6 as the “Extended Attributes” features are still in development and can’t really be used (they are by default set to 0 and I haven’t found a way to change them).

I was not nearly as successful with these as with regular metric weights, it’s possible they didn’t work at all on some of the routers I was using as they were introduced in version 16.8. While the bandwidth of gigabitEthernet interfaces is high, regular EIGRP metrics can handle them. EIGRP wide metrics should instead be used to handle more complex tasks such as an EIGRP EtherChannel, which I hope to try out in the future. I also want to experiment with more features such as EIGRP NSF (non-stop forwarding feature).

Lab Commands

General Configuration

enable: done in user exec mode to trigger privileged exec mode

configure terminal: done in privileged exec mode to trigger global config mode

hostname “x”: assigns a name x to the router

ipv6 unicast-routing: enables ipv6 on the router

interface “x”: opens an interface of name x

ip address “address” “mask”: configures an ip address on an interface

ipv6 address “address”: configures an ipv6 address on an interface

ipv6 address “mask” link-local: configures an ipv6 link-local address on an interface

no shutdown: turns an interface on for use

EIGRP Configuration

router eigrp “x”: creates an instance of ipv4 EIGRP with autonomous system number x for you to add ipv4 network statements and other commands

ipv6 eigrp “x”: creates an instance of ipv6 EIGRP with autonomous system number x for you to add other commands

ipv6 router eigrp “x”: configures ipv6 EIGRP on an interface, where x is EIGRP’s autonomous system number.

*Note that while ipv4 EIGRP can only be configured with network statements, ipv6 can only be configured with interface configs.*

network <mask>

metric weights 0 <K1> <K2> <K3> <K4> <K5>: chooses a preferred route by evaluating the parameters you input for K, each 0-255.

metric weights 0 <K1> <K2> <K3> <K4> <K5> <K6>: “wide metrics”, same as regular metrics but accommodates larger bandwidths and faster interfaces.

variance x: prevents EIGRP from not considering slow links in load balancing. X is a value from 1-255 that should (normally) be greater than the quotient of the feasible successor and successor.

redistribute eigrp x metric <Bw> <delay> <reliability> <load> <MTU>: for as x, best practice is 100000 1 255 1 1500 respectively.

Network diagram

10.5.5.5

1040:AB8::5

10.4.4.4

1030:AB8::4

10.3.3.3

1020:AB8::3

10.2.2.2

1010:AB8::2

10.1.1.1

1000:AB8::1

g0/0/1

192.168.2.2 2060:AB8::6

g0/0/1

192.168.2.2 2040:AB8::4

g0/0/1

192.168.2.2 2080:AB8::8

g0/0/0

192.168.1.1 2010:AB8::1

g0/0/1

192.168.1.2 2020:AB8::2

g0/0/0

192.168.2.1 2070:AB8::7

g0/0/0

192.168.3.1 2050:AB8::5

g0/0/0

192.168.2.1 2030:AB8::3

Configurations

Note: even after configuring variance on several routers, it didn’t show up in their running-configs. All ip routes missing route for 172.16.1.0 and associated ipv6 network. Done on purpose to display config without redundant links.

R1 show run

hostname R1  
  
ipv6 unicast-routing

ipv6 router eigrp 1  
metric weights 0 2 1 1 1 1

eigrp router-id 10.1.1.1  
 no shutdown

interface Loopback0  
 ip address 1.1.1.1 255.255.255.0

ipv6 address 1000:AB8::1/64 eui-64  
 ipv6 enable  
 ipv6 eigrp 1  
  
interface g0/0/0  
 ip address 192.168.1.1 255.255.255.0

ipv6 address FE80::1 link-local  
 ipv6 address 2010:AB8::1/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

router eigrp 1  
network 192.168.1.0 0.0.0.255

network 10.1.1.0 0.0.0.255

metric weights 0 2 1 1 1 1

R2 show run

hostname R2  
  
ipv6 unicast-routing

interface Loopback0

ip address 10.2.2.2 255.255.255.0  
 ipv6 address 1010:AB8::2/64 eui-64  
 ipv6 enable  
 ipv6 eigrp 1

interface g0/0/1  
 ip address 192.168.1.2 255.255.255.0  
 ipv6 address FE80::2 link-local  
 ipv6 address 2020:AB8::2/64  
 ipv6 enable  
 ipv6 eigrp 1  
 no shutdown

interface g0/0/0  
 ip address 192.168.2.1 255.255.255.0  
 ipv6 address FE80::3 link-local  
 ipv6 address 2030:AB8::3/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

ipv6 router eigrp 1

metric weights 0 2 1 1 1 1

eigrp router-id 2.2.2.2  
 no shutdown

router eigrp 1  
network 192.168.1.0 0.0.0.255

network 192.168.2.0 0.0.0.255

network 10.2.2.0 0.0.0.255

network 172.16.1.0 0.0.0.255

metric weights 0 2 1 1 1 1

Interface g0/2/0

Bandwidth 20000

ip address 172.16.1.1 255.255.255.0  
 ipv6 address 3010:AB8::1/64 eui-64

ipv6 address FE80::9 link-local

ipv6 enable  
 ipv6 eigrp 1

no shutdown

R3 show run

hostname R3  
  
ipv6 unicast-routing

interface Loopback0

ip address 10.3.3.3 255.255.255.0

ipv6 address 1020:AB8::3/64 eui-64  
 ipv6 enable  
 ipv6 eigrp 1

interface g0/0/1  
 ip address 192.168.2.2 255.255.255.0  
 ipv6 address FE80::4 link-local  
 ipv6 address 2040:AB8::4/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

interface g0/0/0  
 ip address 192.168.3.1 255.255.255.0  
 ipv6 address FE80::5 link-local  
 ipv6 address 2050:AB8::5/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

ipv6 router eigrp 1

metric weights 0 2 1 1 1 1

eigrp router-id 3.3.3.3  
 no shutdown

router eigrp 1

network 192.168.2.0 0.0.0.255

network 192.168.3.0 0.0.0.255

network 10.3.3.0 0.0.0.255

metric weights 0 2 1 1 1 1

R4 show run

hostname R4  
  
ipv6 unicast-routing  
  
interface Loopback0

ip address 10.4.4.4 255.255.255.0

ipv6 address 1030:AB8::4/64 eui-64  
 ipv6 enable  
 ipv6 eigrp 1

interface g0/0/1  
 ip address 192.168.3.2 255.255.255.0  
 ipv6 address FE80::6 link-local  
 ipv6 address 2060:AB8::6/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

interface g0/0/0  
 ip address 192.168.4.1 255.255.255.0  
 ipv6 address FE80::7 link-local  
 ipv6 address 2070:AB8::7/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

ipv6 router eigrp 1

metric weights 0 2 1 1 1 1

eigrp router-id 4.4.4.4  
 no shutdown

router eigrp 1

network 192.168.3.0 0.0.0.255

network 192.168.4.0 0.0.0.255

network 10.4.4.0 0.0.0.255

network 172.16.1.0 0.0.0.255

metric weights 0 2 1 1 1 1

Interface g0/2/0

Bandwidth 20000

ip address 172.16.1.2 255.255.255.0  
 ipv6 address 3020:AB8::2/64 eui-64

ipv6 address FE80::10 link-local

ipv6 enable  
 ipv6 eigrp 1

no shutdown

R5 show run

hostname R5  
  
ipv6 unicast-routing  
  
interface Loopback0

ip address 10.5.5.5 255.255.255.0

ipv6 address 1040:AB8::5/64 eui-64  
 ipv6 enable  
 ipv6 eigrp 1

interface g0/0/1  
 ip address 192.168.4.2 255.255.255.0  
 ipv6 address FE80::8 link-local  
 ipv6 address 2080:AB8::8/64  
 ipv6 enable  
 ipv6 eigrp 1

no shutdown

ipv6 router eigrp 1  
 metric weights 0 2 1 1 1 1

eigrp router-id 5.5.5.5  
 no shutdown

router eigrp 1

network 192.168.4.0 0.0.0.255

network 10.5.5.0 0.0.0.255

metric weights 0 2 1 1 1 1

R1 show ip route

1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 1.1.1.0/24 is directly connected, Loopback0

L 1.1.1.1/32 is directly connected, Loopback0

10.0.0.0/24 is subnetted, 4 subnets

D 10.2.2.0/24 [90/130816] via 192.168.1.2, 00:01:20, GigabitEthernet0/0/0

D 10.3.3.0/24 [90/131072] via 192.168.1.2, 00:01:01, GigabitEthernet0/0/0

D 10.4.4.0/24 [90/131328] via 192.168.1.2, 00:00:41, GigabitEthernet0/0/0

D 10.5.5.0/24 [90/131584] via 192.168.1.2, 00:00:21, GigabitEthernet0/0/0

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.1.0/24 is directly connected, GigabitEthernet0/0/0

L 192.168.1.1/32 is directly connected, GigabitEthernet0/0/0

D 192.168.2.0/24 [90/3072] via 192.168.1.2, 00:01:03, GigabitEthernet0/0/0

D 192.168.3.0/24 [90/3328] via 192.168.1.2, 00:00:42, GigabitEthernet0/0/0

D 192.168.4.0/24 [90/3584] via 192.168.1.2, 00:00:23, GigabitEthernet0/0/0

R2 show ip route

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

C 10.2.2.0/24 is directly connected, Loopback0

L 10.2.2.2/32 is directly connected, Loopback0

D 10.3.3.0/24 [90/130816] via 192.168.2.2, 00:02:16, GigabitEthernet0/0/0

D 10.4.4.0/24 [90/131072] via 192.168.2.2, 00:01:56, GigabitEthernet0/0/0

D 10.5.5.0/24 [90/131328] via 192.168.2.2, 00:01:36, GigabitEthernet0/0/0

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.1.0/24 is directly connected, GigabitEthernet0/0/1

L 192.168.1.2/32 is directly connected, GigabitEthernet0/0/1

192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.2.0/24 is directly connected, GigabitEthernet0/0/0

L 192.168.2.1/32 is directly connected, GigabitEthernet0/0/0

D 192.168.3.0/24 [90/3072] via 192.168.2.2, 00:01:58, GigabitEthernet0/0/0

D 192.168.4.0/24 [90/3328] via 192.168.2.2, 00:01:38, GigabitEthernet0/0/0

R3 show ip route

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

D 10.2.2.0/24 [90/130816] via 192.168.2.1, 00:00:50, GigabitEthernet0/0/1

C 10.3.3.0/24 is directly connected, Loopback0

L 10.3.3.3/32 is directly connected, Loopback0

D 10.4.4.0/24 [90/130816] via 192.168.3.2, 00:00:30, GigabitEthernet0/0/0

D 10.5.5.0/24 [90/131072] via 192.168.3.2, 00:00:10, GigabitEthernet0/0/0

D 192.168.1.0/24 [90/3072] via 192.168.2.1, 00:00:50, GigabitEthernet0/0/1

192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.2.0/24 is directly connected, GigabitEthernet0/0/1

L 192.168.2.2/32 is directly connected, GigabitEthernet0/0/1

192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.3.0/24 is directly connected, GigabitEthernet0/0/0

L 192.168.3.1/32 is directly connected, GigabitEthernet0/0/0

D 192.168.4.0/24 [90/3072] via 192.168.3.2, 00:00:12, GigabitEthernet0/0/0

R4 show ip route

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

D 10.2.2.0/24 [90/131072] via 192.168.3.1, 00:02:02, GigabitEthernet0/0/1

D 10.3.3.0/24 [90/130816] via 192.168.3.1, 00:02:02, GigabitEthernet0/0/1

C 10.4.4.0/24 is directly connected, Loopback0

L 10.4.4.4/32 is directly connected, Loopback0

D 10.5.5.0/24 [90/130816] via 192.168.4.2, 00:01:43, GigabitEthernet0/0/0

D 192.168.1.0/24 [90/3328] via 192.168.3.1, 00:02:02, GigabitEthernet0/0/1

D 192.168.2.0/24 [90/3072] via 192.168.3.1, 00:02:02, GigabitEthernet0/0/1

192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.3.0/24 is directly connected, GigabitEthernet0/0/1

L 192.168.3.2/32 is directly connected, GigabitEthernet0/0/1

192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.4.0/24 is directly connected, GigabitEthernet0/0/0

L 192.168.4.1/32 is directly connected, GigabitEthernet0/0/0

R5 show ip route

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

D 10.2.2.0/24 [90/131328] via 192.168.4.1, 00:01:50, GigabitEthernet0/0/1

D 10.3.3.0/24 [90/131072] via 192.168.4.1, 00:01:50, GigabitEthernet0/0/1

D 10.4.4.0/24 [90/130816] via 192.168.4.1, 00:01:50, GigabitEthernet0/0/1

C 10.5.5.0/24 is directly connected, Loopback0

L 10.5.5.5/32 is directly connected, Loopback0

D 192.168.1.0/24 [90/3584] via 192.168.4.1, 00:01:50, GigabitEthernet0/0/1

D 192.168.2.0/24 [90/3328] via 192.168.4.1, 00:01:50, GigabitEthernet0/0/1

D 192.168.3.0/24 [90/3072] via 192.168.4.1, 00:01:50, GigabitEthernet0/0/1

192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.4.0/24 is directly connected, GigabitEthernet0/0/1

L 192.168.4.2/32 is directly connected, GigabitEthernet0/0/1

R1 show ipv6 route

C 1000:AB8::/64 [0/0]

via Loopback0, directly connected

L 1000:AB8::2E0:A3FF:FEC1:68A6/128 [0/0]

via Loopback0, receive

D 1010:AB8::/64 [90/130816]

via FE80::2, GigabitEthernet0/0/0

D 1020:AB8::/64 [90/131072]

via FE80::2, GigabitEthernet0/0/0

D 1030:AB8::/64 [90/131328]

via FE80::2, GigabitEthernet0/0/0

D 1040:AB8::/64 [90/131584]

via FE80::2, GigabitEthernet0/0/0

C 2010:AB8::/64 [0/0]

via GigabitEthernet0/0/0, directly connected

L 2010:AB8::1/128 [0/0]

via GigabitEthernet0/0/0, receive

D 2020:AB8::/64 [90/3072]

via FE80::2, GigabitEthernet0/0/0

D 2030:AB8::/64 [90/3072]

via FE80::2, GigabitEthernet0/0/0

D 2040:AB8::/64 [90/3328]

via FE80::2, GigabitEthernet0/0/0

D 2050:AB8::/64 [90/3328]

via FE80::2, GigabitEthernet0/0/0

D 2060:AB8::/64 [90/3584]

via FE80::2, GigabitEthernet0/0/0

D 2070:AB8::/64 [90/3584]

via FE80::2, GigabitEthernet0/0/0

D 2080:AB8::/64 [90/3840]

via FE80::2, GigabitEthernet0/0/0

L FF00::/8 [0/0]

via Null0, receive

R2 show ipv6 route

D 1000:AB8::/64 [90/130816]

via FE80::1, GigabitEthernet0/0/1

C 1010:AB8::/64 [0/0]

via Loopback0, directly connected

L 1010:AB8::2E0:F7FF:FE83:1791/128 [0/0]

via Loopback0, receive

D 1020:AB8::/64 [90/130816]

via FE80::4, GigabitEthernet0/0/0

D 1030:AB8::/64 [90/131072]

via FE80::4, GigabitEthernet0/0/0

D 1040:AB8::/64 [90/131328]

via FE80::4, GigabitEthernet0/0/0

D 2010:AB8::/64 [90/3072]

via FE80::1, GigabitEthernet0/0/1

C 2020:AB8::/64 [0/0]

via GigabitEthernet0/0/1, directly connected

L 2020:AB8::2/128 [0/0]

via GigabitEthernet0/0/1, receive

C 2030:AB8::/64 [0/0]

via GigabitEthernet0/0/0, directly connected

L 2030:AB8::3/128 [0/0]

via GigabitEthernet0/0/0, receive

D 2040:AB8::/64 [90/3072]

via FE80::4, GigabitEthernet0/0/0

D 2050:AB8::/64 [90/3072]

via FE80::4, GigabitEthernet0/0/0

D 2060:AB8::/64 [90/3328]

via FE80::4, GigabitEthernet0/0/0

D 2070:AB8::/64 [90/3328]

via FE80::4, GigabitEthernet0/0/0

D 2080:AB8::/64 [90/3584]

via FE80::4, GigabitEthernet0/0/0

L FF00::/8 [0/0]

via Null0, receive

R3 show ipv6 route

D 1000:AB8::/64 [90/131072]

via FE80::3, GigabitEthernet0/0/1

D 1010:AB8::/64 [90/130816]

via FE80::3, GigabitEthernet0/0/1

C 1020:AB8::/64 [0/0]

via Loopback0, directly connected

L 1020:AB8::201:63FF:FECB:C796/128 [0/0]

via Loopback0, receive

D 1030:AB8::/64 [90/130816]

via FE80::6, GigabitEthernet0/0/0

D 1040:AB8::/64 [90/131072]

via FE80::6, GigabitEthernet0/0/0

D 2010:AB8::/64 [90/3328]

via FE80::3, GigabitEthernet0/0/1

D 2020:AB8::/64 [90/3072]

via FE80::3, GigabitEthernet0/0/1

D 2030:AB8::/64 [90/3072]

via FE80::3, GigabitEthernet0/0/1

C 2040:AB8::/64 [0/0]

via GigabitEthernet0/0/1, directly connected

L 2040:AB8::4/128 [0/0]

via GigabitEthernet0/0/1, receive

C 2050:AB8::/64 [0/0]

via GigabitEthernet0/0/0, directly connected

L 2050:AB8::5/128 [0/0]

via GigabitEthernet0/0/0, receive

D 2060:AB8::/64 [90/3072]

via FE80::6, GigabitEthernet0/0/0

D 2070:AB8::/64 [90/3072]

via FE80::6, GigabitEthernet0/0/0

D 2080:AB8::/64 [90/3328]

via FE80::6, GigabitEthernet0/0/0

L FF00::/8 [0/0]

via Null0, receive

R4 show ipv6 route

D 1000:AB8::/64 [90/131328]

via FE80::5, GigabitEthernet0/0/1

D 1010:AB8::/64 [90/131072]

via FE80::5, GigabitEthernet0/0/1

D 1020:AB8::/64 [90/130816]

via FE80::5, GigabitEthernet0/0/1

C 1030:AB8::/64 [0/0]

via Loopback0, directly connected

L 1030:AB8::201:96FF:FED4:E3BC/128 [0/0]

via Loopback0, receive

D 1040:AB8::/64 [90/130816]

via FE80::8, GigabitEthernet0/0/0

D 2010:AB8::/64 [90/3584]

via FE80::5, GigabitEthernet0/0/1

D 2020:AB8::/64 [90/3328]

via FE80::5, GigabitEthernet0/0/1

D 2030:AB8::/64 [90/3328]

via FE80::5, GigabitEthernet0/0/1

D 2040:AB8::/64 [90/3072]

via FE80::5, GigabitEthernet0/0/1

D 2050:AB8::/64 [90/3072]

via FE80::5, GigabitEthernet0/0/1

C 2060:AB8::/64 [0/0]

via GigabitEthernet0/0/1, directly connected

L 2060:AB8::6/128 [0/0]

via GigabitEthernet0/0/1, receive

C 2070:AB8::/64 [0/0]

via GigabitEthernet0/0/0, directly connected

L 2070:AB8::7/128 [0/0]

via GigabitEthernet0/0/0, receive

D 2080:AB8::/64 [90/3072]

via FE80::8, GigabitEthernet0/0/0

L FF00::/8 [0/0]

via Null0, receive

R5 show ipv6 route

D 1000:AB8::/64 [90/131584]

via FE80::7, GigabitEthernet0/0/1

D 1010:AB8::/64 [90/131328]

via FE80::7, GigabitEthernet0/0/1

D 1020:AB8::/64 [90/131072]

via FE80::7, GigabitEthernet0/0/1

D 1030:AB8::/64 [90/130816]

via FE80::7, GigabitEthernet0/0/1

C 1040:AB8::/64 [0/0]

via Loopback0, directly connected

L 1040:AB8::20C:CFFF:FE98:3D97/128 [0/0]

via Loopback0, receive

D 2010:AB8::/64 [90/3840]

via FE80::7, GigabitEthernet0/0/1

D 2020:AB8::/64 [90/3584]

via FE80::7, GigabitEthernet0/0/1

D 2030:AB8::/64 [90/3584]

via FE80::7, GigabitEthernet0/0/1

D 2040:AB8::/64 [90/3328]

via FE80::7, GigabitEthernet0/0/1

D 2050:AB8::/64 [90/3328]

via FE80::7, GigabitEthernet0/0/1

D 2060:AB8::/64 [90/3072]

via FE80::7, GigabitEthernet0/0/1

D 2070:AB8::/64 [90/3072]

via FE80::7, GigabitEthernet0/0/1

C 2080:AB8::/64 [0/0]

via GigabitEthernet0/0/1, directly connected

L 2080:AB8::8/128 [0/0]

via GigabitEthernet0/0/1, receive

L FF00::/8 [0/0]

via Null0, receive

R4: show ip eigrp, implementation of variance

Metric weight K1=2, K2=1, K3=1, K4=1, K5=1

Soft SIA disabled

NSF-aware route hold timer is 240

EIGRP NSF disabled

NSF signal timer is 20s

NSF converge timer is 120s

Router-ID: 2.2.2.2

Topology : 0 (base)

Active Timer: 3 min

Distance: internal 90 external 170

Maximum path: 16

Maximum hopcount 100

Maximum metric variance 1

EIGRP-IPv4 Protocol for AS(1)

Metric weight K1=2, K2=1, K3=1, K4=1, K5=1

Soft SIA disabled

NSF-aware route hold timer is 240

EIGRP NSF disabled

NSF signal timer is 20s

NSF converge timer is 120s

Router-ID: 10.2.2.2

R4: 172.16.1.0 network as a separate autonomous system (removed from original config)

EIGRP-IPv4 Traffic Statistics for AS(1)

Hellos sent/received: 3974/3972

Updates sent/received: 40/41

Queries sent/received: 1/4

Replies sent/received: 4/1

Acks sent/received: 42/41

SIA-Queries sent/received: 0/0

SIA-Replies sent/received: 0/0

Hello Process ID: 515

PDM Process ID: 514

Socket Queue: 0/10000/2/0 (current/max/highest/drops)

Input Queue: 0/10000/2/0 (current/max/highest/drops)

EIGRP-IPv4 Traffic Statistics for AS(2)

Hellos sent/received: 996/989

Updates sent/received: 5/5

The “P” next to the routes in “show ip eigrp topology” stand for “passive” meaning a route is reachable. “A” for “active” will show if the router is looking for more routes.

Queries sent/received: 0/0

Replies sent/received: 0/0

Acks sent/received: 3/3

SIA-Queries sent/received: 0/0

SIA-Replies sent/received: 0/0

Hello Process ID: 518

PDM Process ID: 517

Socket Queue: 0/10000/1/0 (current/max/highest/drops)

Input Queue: 0/10000/1/0 (current/max/highest/drops)

R4: show ip eigrp topology (missing 172.16.1.0 network)

P 10.2.2.0/24, 1 successors, FD is 131072

via 192.168.3.1 (131072/130816), GigabitEthernet0/0/1

P 10.3.3.0/24, 1 successors, FD is 130816

via 192.168.3.1 (130816/128256), GigabitEthernet0/0/1

P 10.4.4.0/24, 1 successors, FD is 128256

via Connected, Loopback0

P 10.5.5.0/24, 1 successors, FD is 130816

via 192.168.4.2 (130816/128256), GigabitEthernet0/0/0

P 192.168.1.0/24, 1 successors, FD is 3328

via 192.168.3.1 (3328/3072), GigabitEthernet0/0/1

P 192.168.2.0/24, 1 successors, FD is 3072

via 192.168.3.1 (3072/2816), GigabitEthernet0/0/1

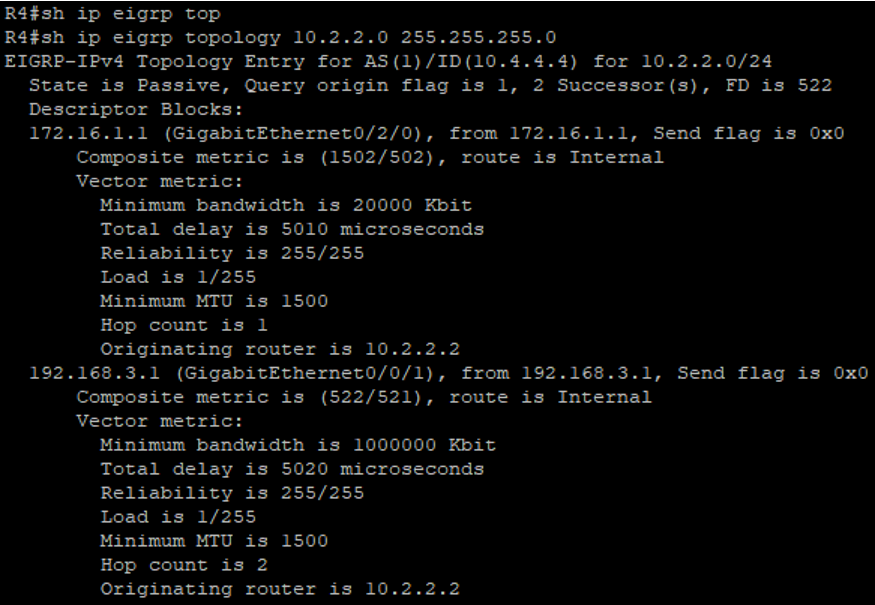
P 192.168.3.0/24, 1 successors, FD is 2816

via Connected, GigabitEthernet0/0/1

P 192.168.4.0/24, 1 successors, FD is 2816

via Connected, GigabitEthernet0/0/0

R4: Proof of unequal cost load balancing



Problems

Following splitting up with my lab partner the previous lab, he did not want to work with me for this lab. I’m not sure why this happened as our split was an amicable one. It’s possible he thought working alone was more productive.

At one point in time, IPv4 EIGRP routes didn’t show up in routing table despite me having entered grammatically correct network statements. The problem was that the statements were worded as network 192.168.0.1 for IP addresses of 192.168.1.1. While this was a minor goof-up, the loopback addresses were totally wrong, e.g., network 10.0.0.1 for 1.1.1.1. Changing the addresses

Moreover, IPv6 EIGRP on R2 displayed a “retry limit” error wherein the interface would successfully connect using EIGRP, then disconnect.

After troubleshooting these issues, I found out that “holding time expired” for both ipv4 and ipv6 on the link between R1 and R2, and an interface goodbye was received, meaning EIGRP went down while the link was still active. At the end of the error messages, the interfaces came back up again, and when I checked the show ip route and show ip eigrp neighbors commands, I saw all the routes and neighbors in place.

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.1.2 (GigabitEthernet0/0/0) is down: holding time expired

%DUAL-5-NBRCHANGE: IPv6-EIGRP 1: Neighbor FE80::2 (GigabitEthernet0/0/0) is down: holding time expired

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.1.2 (GigabitEthernet0/0/0) is up: new adjacency

%DUAL-5-NBRCHANGE: IPv6-EIGRP 1: Neighbor FE80::2 (GigabitEthernet0/0/0) is up: new adjacency

%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 192.168.1.2 (GigabitEthernet0/0/0) is down: Interface Goodbye received

%DUAL-5-NBRCHANGE: IPv6-EIGRP 1: Neighbor FE80::2 (GigabitEthernet0/0/0) is down: Interface Goodbye received

In configuring the IPv4 metric weights, I started from R5, working my way up to R1, the other routers briefly displayed a K-value mismatch (which I expected having read Cisco documentation on K-values). When the messages subsided, I figured this would mean the configuration was working normally again with the new cost equation. However, when I tried to set up variance, a router showed no network available. I checked all the IPv4 routes and EIGRP had disappeared. I realized this was solely a problem with Packet Tracer, as putting the metric weights on the real routers did not display this error.

Additionally, there was this, which happened on the real routers but not on packet tracer:

\*Dec 9 22:14:11.367: %DUAL-6-NBRINFO: EIGRP-IPv4 1: Neighbor 192.168.1.1 (GigabitEthernet0/0/0) is blocked: not on common subnet (192.168.2.1/24)

Finally, I was not able to set up variance initially. First, I tried the Serial ports (which seemed a viable option despite me writing a note otherwise) which would result in a serial port running alone, but then being excluded from the EIGRP topology when the main Ethernet links came online. Mr. Mason suggested I try additional Ethernet ports found on some routers but those didn’t work either. Later, I revisited variance and got it working after adjusting an interface’s bandwidth, and after finding out an EIGRP network statement was wrong. In doing so, I figured out I had variance working all along, just not in the right way (not being used to allow use of slower links). The below command is the result of an interface being unused due to an error in implementing the variance command.

\*Feb 3 21:46:07.984: %COMMON\_FIB-4-UNEQUAL: Ratio of unequal path weightings (1 48) prevents oce IP adj out of GigabitEthernet0/2/0, addr 172.16.1.1 from being used.

### Conclusion

This particular lab is one I should have done on the racks instead of packet tracer, except for the part where I couldn’t find my IPv4 error. While in my last lab, OSPF worked perfectly on packet tracer, packet tracer did not support certain features of EIGRP crucial at least for the bonus, for example metric weights.

I spent too much time going after the bonus requirements and ended up turning in this lab *very* late. I aim to do otherwise next semester.

It may not be ready in time for the BGP lab, but I want to make a Java/Python program to standardize my lab configs. It will take in a few input parameters, for instance the topology, for a particular lab and spit out several full-fledged router configs.

Text, letter, timeline

Description automatically generated