Overview

Monday, October 15, 2018 11:14

EIGRP

packetlife.net

Protocol Header

8	16	24	32
Version	Opcode	Checksum	
Flags			
Sequence Number			
Acknowledgment Number			
Autonomous System Number			
Type		Length	
Value			

Metric Formula

$$256 * \left(K_1 * \mathbf{bw} + \frac{K_2 * \mathbf{bw}}{256 - \mathbf{load}} + K_3 * \mathbf{delay}\right) * \frac{K_5}{\mathbf{rel} + K_4}$$

- **bw** = 10^7 / minimum path bandwidth in kbps
- **delay** = interface delay in μsecs / 10

EIGRP Configuration

Protocol Configuration

```
! Enable EIGRP
router eigrp <ASN>

! Add networks to advertise
network <IP address> <wildcard mask>

! Configure K values to manipulate metric formula
metric weights 0 <k1> <k2> <k3> <k4> <k5>

! Disable automatic route summarization
no auto-summary

! Designate passive interfaces
passive-interface (<interface> | default)

! Enable stub routing
eigrp stub [receive-only | connected | static | summary]

! Statically identify neighboring routers
neighbor <IP address> <interface>
```

Interface Configuration

```
! Set maximum bandwidth EIGRP can consume
ip bandwidth-percent eigrp <AS> <percentage>

! Configure manual summarization of outbound routes
ip summary-address eigrp <AS> <IP address> <mask> [<AD>]

! Enable MD5 authentication
ip authentication mode eigrp <AS> md5
ip authentication key-chain eigrp <AS> <key-chain>

! Configure hello and hold timers
ip hello-interval eigrp <AS> <seconds>
ip hold-time eigrp <AS> <seconds>

! Disable split horizon for EIGRP
no ip split-horizon eigrp <AS>
```

Attributes

Type	Distance Vector
Algorithm	DUAL
Internal AD	90
External AD	170
Summary AD	5
Standard	Cisco proprietary
Protocols	IP, IPX, Appletalk
Transport	IP/88
Authentication	MD5
Multicast IP	224.0.0.10
Hello Timers	5/60
Hold Timers	15/180

K Defaults

K_1	1
K_2	0
K_3	1
K_4	0
K_5	0

Packet Types

1	Update
3	Query
4	Reply
5	Hello
8	Acknowledge

Terminology

Reported Distance
The metric for a route advertised by a neighbor

Feasible Distance
The distance advertised by a neighbor plus the cost to get to that neighbor

Stuck In Active (SIA)
The condition when a route becomes unreachable and not all queries for it are answered; adjacencies with unresponsive neighbors are reset

Passive Interface
An interface which does not participate in EIGRP but whose network is advertised

Stub Router
A router which advertises only a subset of routes, and is omitted from the route query process

Troubleshooting

```
show ip eigrp interfaces
show ip eigrp neighbors
show ip eigrp topology
show ip eigrp traffic
clear ip eigrp neighbors
debug ip eigrp [packet | neighbors]
```

```
ip hold-time eigrp <AS> <seconds>

! Disable split horizon for EIGRP
no ip split-horizon eigrp <AS>
```

```
clear ip eigrp neighbors
debug ip eigrp [packet | neighbors]
```

by Jeremy Stretch v2.1

Fundamentals

Monday, October 15, 2018 10:07

1. ## Significance of AS

A group of networks under a single administrative control which could be an Internet Service Provider (ISP) or a large Enterprise Organization. In a Large organization there can be many AS's and the multiple AS's create a logical boundary

2. ## Network Statement

It tells the router 3 things
Which interfaces to listen on. Accept routing protocol packets on the interfaces described by the network statement.
Which interfaces to advertise out and send routing protocol packets on.
What interface subnets/networks to advertise.

```
Router eigrp 100
network 10.10.10.1 0.0.0.0
```

3. ## Show Commands

Router#show ip eigrp neighbors	Displays the neighbor table.
Router#show ip eigrp neighbors detail	Displays a detailed neighbor table.
	TIP The **show ip eigrp neighbors detail** command verifies whether a neighbor is configured as a stub router.
Router#show ip eigrp interfaces	Shows information for each interface.
Router#show ip eigrp interfaces serial 0/0/0	Shows information for a specific interface.
Router#show ip eigrp interfaces 100	Shows information for interfaces running process 100.
Router#show ip eigrp topology	Displays the topology table.
	TIP The show ip eigrp topology command shows you where your feasible successors are.
Router#show ip eigrp traffic	Shows the number and type of packets sent and received.
Router#show ip route	Shows the complete routing table.
Router#show ip route eigrp	Shows a routing table with only EIGRP entries.
Router#show ip protocols	Shows the parameters and current state of the active routing protocol process.
Router#show key-chain	Shows authentication key information.

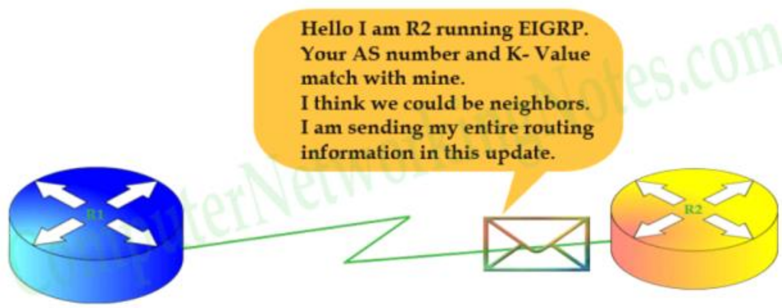
Router#debug eigrp fsm	Displays events/actions related to EIGRP feasible successor metrics (FSM)
Router#debug eigrp packet	Displays events/actions related to EIGRP packets
Router#debug eigrp neighbor	Displays events/actions related to your EIGRP neighbors
Router#debug ip eigrp	Displays events/actions related to EIGRP protocol packets.
Router#debug ip eigrp neighbor	Displays events/actions related to your EIGRP neighbors
Router#debug ip eigrp notifications	Displays EIGRP event notifications

4. ## NET2000

EIGRP Neighbor Discovery process

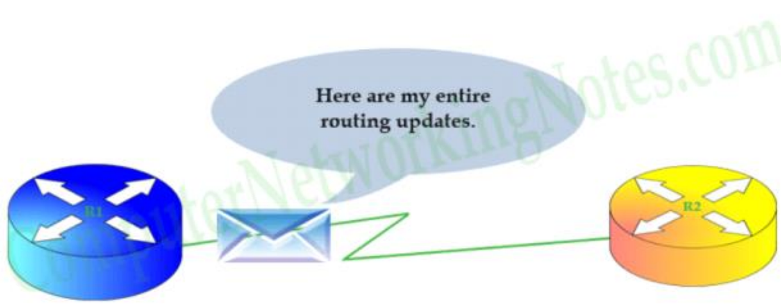
Step 1:- First router R1 sends a hello packet from all active interfaces. This packet contains essential configuration values which are required to be a neighbor.

Step 2:- Receiving router R2 will compare these values with its own configuration values. If both necessary values match (AS number and K-values), it will reply with a routing update. This update includes all routes information from its routing table excluding one route. The route which it learned from the same interface that bring hello packet to it. This mechanism is known as split horizon. It states that if a router receives an update for route on any interface, it will not propagate same route information back to the sender router on same port. Split horizon is used to avoid routing loops.



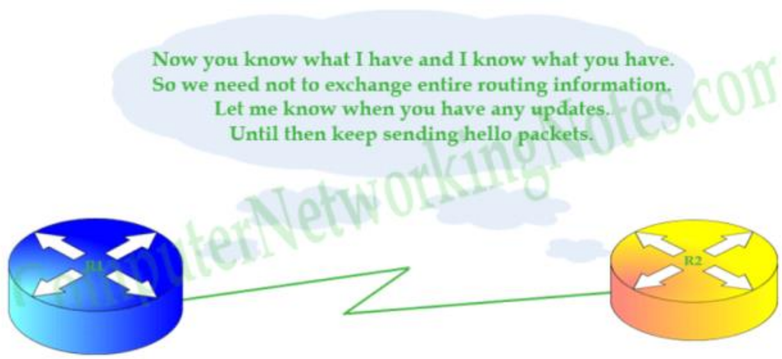
Step 3:- First router will receive R2's routing update and sends an acknowledgement message back to R2.

Step 4:- R1 will sync its EIGRP topology table with routing information that it received in routing update. It will also send a routing update containing all route information from its routing topology to R2.



Step 5:- R2 will respond with an acknowledgement message. It will also sync its EIGRP topology table with routing information that it received in routing update.

At this point, the two routers have becomes neighbor. Now they will maintain this neighborship with ongoing hello packets. If they see any change in network, they will update each other with partial updates.



Partial update contains information only about the recent change.

That's all for this part. In this part we explained how two routers become EIGRP neighbors. In next part we will see how EIGRP routers select the route for routing table.

Hello Packets

EIGRP Hello packets are sent to the link local Multicast group address 224.0.0.10. Hello packets sent by EIGRP do not require an Acknowledgment to be sent confirming that they were received. Because they require no explicit acknowledgment, Hello packets are classified as unreliable EIGRP packets. EIGRP Hello packets have an OPCode of 5. AS Number, K Value and Subnet Address are included in the Hello Packet.

Hello interval is the rate at which EIGRP sends hello packets. The command ip hello-interval eigrp can be used to set the hello interval time manually.

Hello interval defaults to 60 seconds for low-speed, NBMA networks and 5 seconds for other types of networks.

Hold Timer is the amount of time that a router will consider a neighbor alive without receiving hello packets. The hold time is typically three times the hello interval. You can adjust the hold time with the `ip hold-time eigrp` command.

Unlike OSPF changing the hello timer does not automatically adjust the hold timer.

Sometimes hello packets are lost in congested networks and neighbor relationships start to flap; in such cases you may want to increase the hold time.

Increasing the hold timer delays convergence, this is an undesirable effect. If the default hello/hold time value is not suitable for your network try to find the value that best suits your network.

Neighbors

Requirement	EIGRP	OSPF
The routers must be able to send packets to on another.	Yes	Yes
Interface's primary address (not secondary address) must be on same subnet as the network matched by the **network** statement.	Yes	Yes
The interface connecting out to a neighbor must not be passive.	Yes	Yes
Must use the same autonomous system (for EIGRP) or process-ID (for OSPF) in the **router** configuration command.	Yes	No
Hello timer and the Hold timer (for EIGRP) or Dead timer (for OSPF) much match.	No	Yes
Neighbors must authenticate with one another if authentication is configured.	Yes	Yes
Must be in the same area.	N/A	Yes
The IP MTU much match.	No	Yes
K-values must match.	Yes	N/A
Router IDs (RIDs) must be unique.	No (except for EIGRP routers injecting external routes into the EIGRP process)	Yes

For EIGRP to Neighbor

1. The devices must be in the same autonomous system (AS)
2. The devices must have the same authentication configuration
3. The devices must have the same *k-values*

Once neighbor relationships are formed and the routers are talking to one another; the topology table begins to form. This table contains information needed to build a set of distances and vectors to each reachable network including:

1. Lowest bandwidth path to this destination by the upstream neighbors
2. Total delay
3. Path reliability
4. Path loading
5. MTU
6. FD
7. AD
8. Route Source (EX for external routes)
9. Successors
10. Feasible Successors

Each EIGRP router maintains a separate topology table for each network protocol. Routes are classified as external or internal (relative to the AS). The table can track info regarding external routes and, administrator tags

All the external routes have a 32 Bit ID, AS number of the destination, protocol used in that destination, cost or metric received from the external protocol, and a configurable admin route tag.

`show ip eigrp topology`

Output from this command shows the successor routes (that is, the preferred routes) and feasible successor routes (that is, backup routes) known to the EIGRP routing process. **<u>Please keep in mind that a route's appearance in the EIGRP topology table does not guarantee its presence in the router's IP routing table</u>**.

Specifically, the successor routes present in the EIGRP topology table are only candidates to be injected into the router's IP routing table. For example, the router might possess more believable routing information for a network, such as a statically configured route with an administrative distance of 1. If EIGRP is indeed the most believable routing source for a specific network, then that network will be injected into the router's IP routing table.

```
Router_B#show ip eigrp topology
IP-EIGRP Topology Table for AS 10

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 20.0.0.0/8, 1 successors, FD is 2169856
         via Connected, Serial0/0
P 10.0.0.0/8, 1 successors, FD is 2172416
         via 20.0.0.1 (2172416/28160), Serial0/0
Router_B#
```

P: Passive: means the router is not looking for the route actively, thus it means it is in good situation.

A: The status of 'Active' means some instability in network.

FD: Feasible Distance: metric to a destination

2172416 / 28160: In the output 2172416 is the feasible distance and 28160 is the advertised distance.
Advertised distance is the distance from your neighbor to destination.
Feasible distance is the total distance from you to the destination.

Passive Interface

In EIGRP the passive-interface command stops sending outgoing hello packets, hence the router can not form any neighbor relationship via the passive interface**.** This behavior stops both outgoing and incoming routing updates.

Syntax :

R1(config)# *router eigrp 10*
R1(config-router)# *network 10.4.0.0*
R1(config-router)# *network 10.2.0.0*
R1(config-router)# *passive-interface s0*

In OSPF the passive-interface has a similar behavior to EIGRP**.** The command suppresses hello packets and hence neighbor relationships**.**

In OSPF the passive-interface has a similar behavior to EIGRP**.** The command suppresses hello packets and hence neighbor relationships**.**

```
R1(config)# router OSPF 101
R1(config-router)#  network 10.4.0.0
R1(config-router)#  network 10.2.0.0
R1(config-router)#  passive-interface s0
```

Passive interface default command can be used in both EIGRP and OSPF like we used in RIP

Always remember, that the passive-interface command will prevent EIGRP (and OSPF) from forming neighbor relationships out of that interface. No routing updates are passed in either direction.

Important: Passive interface command applying on interfaces wont effect on the sub interfaces created under it.If you want to active "passive interface" command on sub interface,it should be given on that specific sub interface

Router ID Significance

Each EIGRP router has a unique 32-bit router ID (RID) number that is represented the same way as an IP address.
EIGRP automatically selects the highest IP address on any active loopback interface as the router ID. If there is no loopback interface then the highest IP address on any active interface is used. You can also overrule this by manually setting the router ID.

Even though each router should have a unique router ID, it doesn't matter all that much. Two EIGRP routers with the same router ID will still form a neighbor adjacency. When we use OSPF, the router ID is important to know if you are digging in the OSPF LSDB. When we use EIGRP we don't really care about the router ID, you don't need to know it if you are looking at the topology table.

There is one exception to this, when an EIGRP router receives an external route (redistributed route) from a neighbor that has the same router ID then it will **not accept it**. This is an interesting scenario for a troubleshooting lab so let's take a closer look at it.

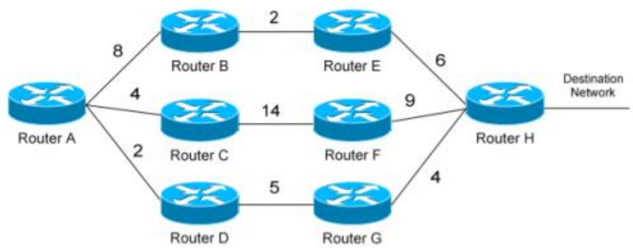
EIGRP METRIC

Monday, October 15, 2018 10:09

Choosing The Best Route

Once EIGRP neighbors form adjacencies, they will begin to share routing information. Each router's update contains a list of all routes known by that router, and the respective metrics for those routes. All such routes are added to an EIGRP router's topology table. The route with the lowest metric to each network will become the Feasible Distance (FD).

FD: The Feasible Distance for each network will be installed into the routing table. The Feasible Distance is derived from the Advertised Distance of the router sending the update, and the local router's metric to the advertising router

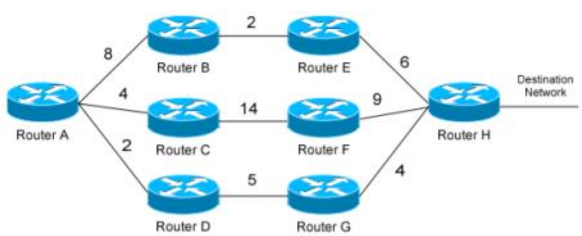


Router A has three separate paths to the Destination Network, either through Router B, C, or D. If we add up the metrics to form a "distance" (the metrics are greatly simplified in this example), we can determine the following:

- Router B's Feasible Distance to the Destination Network is 8.
- Router C's Feasible Distance to the Destination Network is 23.
- Router D's Feasible Distance to the Destination Network is 9.

Router B sends an update to Router A, it will provide an **Advertised Distance** of 8 to the Destination Network. Router C will provide an AD of 23, and D will provide an AD of 9.

Router A calculates the total distance to the Destination network by adding the AD of the advertising router, with its own distance to *reach* that advertising router. For example, Router A's metric to Router B is 8; thus, the *total* distance will be 16 to reach the Destination Network through Router B.



Remember, however, that Router A's Feasible Distance must be the route with the lowest metric. If we add the Advertised Distance with the local metric between each router, we would see that:

- The route through Router B has a distance of **16** to the destination
- The route through Router C has a distance of **27** to the destination
- The route through Router D has a distance of **11** to the destination

Thus, the route through Router D (metric of 11) would become the **Feasible Distance** for Router A, and is added to the routing table as the *best* route. This route is identified as the **Successor**.

To allow convergence to occur quickly if a link fails, EIGRP includes backup routes in the topology table called **Feasible Successors (FS).** A route will only become a Successor if its Advertised Distance is *less* than the current Feasible Distance**.** This is known as a **Feasible Condition (FC)**.

For example, we determined that Router A's Feasible Distance to the destination is 11, through Router D. Router C's Advertised Distance is 23, and thus *would not* become a Feasible Successor, as it has a higher metric than Router A's current Feasible Distance. Routes that are not Feasible Successors become route **Possibilities.**

Router B's Advertised Distance is 8, which is less than Router A's current Feasible Distance. Thus, the route through Router B to the Destination Network *would* become a Feasible Successor.

Feasible Successors provide EIGRP with redundancy, without forcing routers to re-converge (thus stopping the flow of traffic) when a topology change occurs. If no Feasible Successor exists and a link fails, a route will enter an **Active** (converging) state until an alternate route is found.

EIGRP Metrics

EIGRP can utilize 5 separate metrics to determine the best route to a destination:

- **Bandwidth** (K1) – Slowest link in the route path, measured in kilobits
- **Load** (K2) – Cumulative load of all outgoing interfaces in the path, given as a fraction of 255
- **Delay of the Line** (K3) – Cumulative delay of all outgoing interfaces in the path in tens of microseconds
- **Reliability** (K4) – Average reliability of all outgoing interfaces in the path, given as a fraction of 255
- **MTU** (K5) – The smallest Maximum Transmission Unit in the path. The MTU value is actually *never* used to calculate the metric

By default, only **Bandwidth** and **Delay of the Line** are used. This is identical to IGRP, except that EIGRP provides a more granular metric by multiplying the bandwidth and delay by 256. Bandwidth and delay are determined by the interfaces that lead towards the destination network.

By default, the full formula for determining the EIGRP metric is:

[10000000/bandwidth + delay] * 256

The bandwidth value represents the link with the *lowest* bandwidth in the path, in kilobits. The delay is the total delay of all outgoing interfaces in the path.

As indicated above, each metric is symbolized with a "K" and then a number. When configuring EIGRP metrics, we actually identify which metrics we want EIGRP to consider. Again, by default, only Bandwidth and Delay are considered. Thus, using on/off logic:

K1 = 1, K2 = 0, K3 = 1, K4 = 0, K5 = 0

If all metrics were set to "on," the full formula for determining the EIGRP metric would be:

[K1 * bandwidth * 256 + (K2 * bandwidth) / (256 - load)
+ K3 * delay * 256] * [K5 / (reliability + K4)]

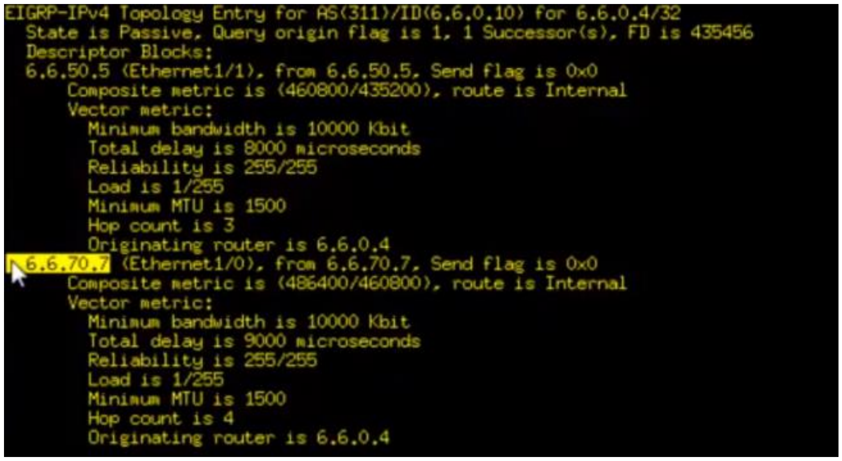
Remember, the "K" value is either set to on ("1") or off ("0").

To See Metric Values for a specific network enter show ip eigrp topology N.N.N.N

```
RTX#sh ip eigrp top 204.100.50.0
IP-EIGRP topology entry for 204.100.50.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
  2297856
  Routing Descriptor Blocks:          FD/RD
  10.1.0.1 (Serial0), from 10.1.0.1, Send flag is 0x0
      Composite metric is (2297856/128256), Route is External
      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
      External data:
        Originating router is 192.168.1.1
        AS number of route is 0
        External protocol is Connected, external metric is 0
        Administrator tag is 0 (0x00000000)
```

Do Metric Calculation

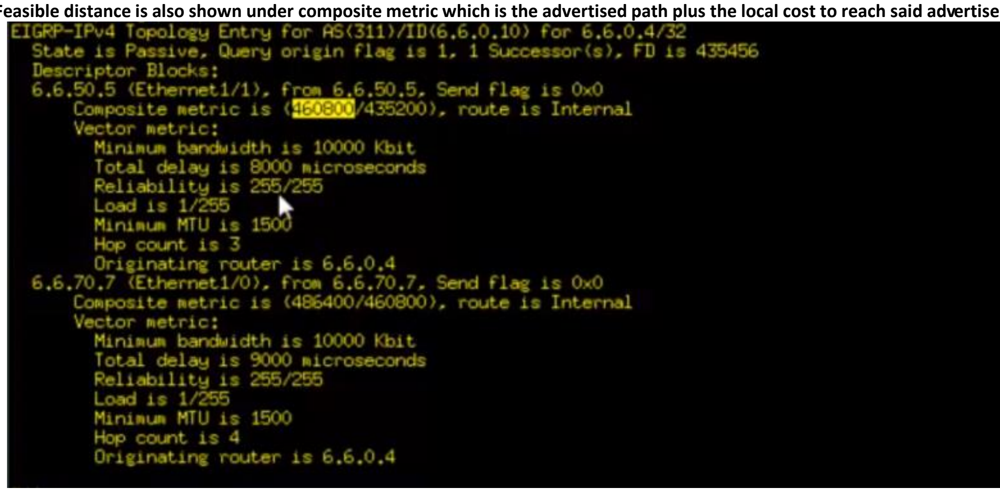
- Enter #show ip eigrp top 6.6.0.4/32


```
EIGRP-IPv4 Topology Entry for AS(311)/ID(6.6.0.10) for 6.6.0.4/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 435456
  Descriptor Blocks:
  6.6.50.5 (Ethernet1/1), from 6.6.50.5, Send flag is 0x0
      Composite metric is (460800/435200), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 8000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 3
        Originating router is 6.6.0.4
  6.6.70.7 (Ethernet1/0), from 6.6.70.7, Send flag is 0x0
      Composite metric is (486400/460800), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 9000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 4
        Originating router is 6.6.0.4
```
The Links in that subnet are showed as 6.6.50.5 and 6.6.70.7

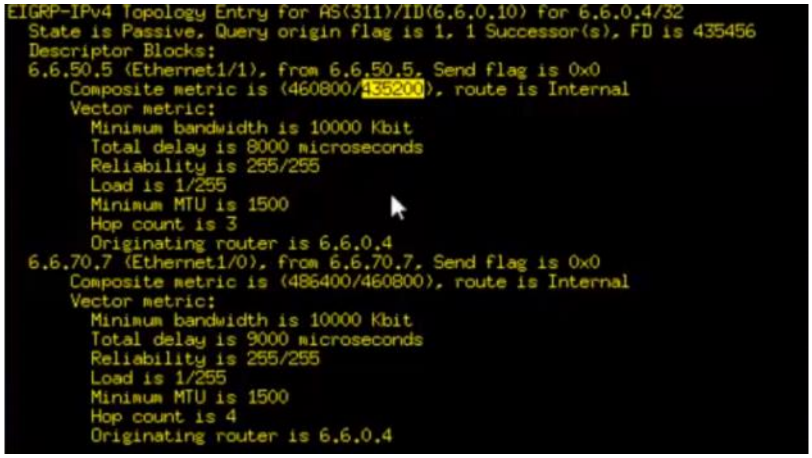
Bandwidth and Delay are shown


```
EIGRP-IPv4 Topology Entry for AS(311)/ID(6.6.0.10) for 6.6.0.4/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 435456
  Descriptor Blocks:
  6.6.50.5 (Ethernet1/1), from 6.6.50.5, Send flag is 0x0
      Composite metric is (460800/435200), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 8000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 3
        Originating router is 6.6.0.4
  6.6.70.7 (Ethernet1/0), from 6.6.70.7, Send flag is 0x0
      Composite metric is (486400/460800), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 9000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 4
        Originating router is 6.6.0.4
```

The Feasible distance is also shown under composite metric which is the advertised path plus the local cost to reach said advertise paths.


```
EIGRP-IPv4 Topology Entry for AS(311)/ID(6.6.0.10) for 6.6.0.4/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 435456
  Descriptor Blocks:
  6.6.50.5 (Ethernet1/1), from 6.6.50.5, Send flag is 0x0
      Composite metric is (460800/435200), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 8000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 3
        Originating router is 6.6.0.4
  6.6.70.7 (Ethernet1/0), from 6.6.70.7, Send flag is 0x0
      Composite metric is (486400/460800), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 9000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 4
        Originating router is 6.6.0.4
```

The Advertised Distance is also shown



```
EIGRP-IPv4 Topology Entry for AS(311)/ID(6.6.0.10) for 6.6.0.4/32
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 435456
  Descriptor Blocks:
  6.6.50.5 (Ethernet1/1), from 6.6.50.5, Send flag is 0x0
      Composite metric is (460800/435200), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 8000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 3
        Originating router is 6.6.0.4
  6.6.70.7 (Ethernet1/0), from 6.6.70.7, Send flag is 0x0
      Composite metric is (486400/460800), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 9000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 4
        Originating router is 6.6.0.4
```

**When doing calculations delay must be divided by ten*

$$EIGRP\ Metric\ = 256\ \times \left[\left(\frac{10^7}{Min.Bandwidth}\right) + \left(\frac{Delay}{10}\right)\right]$$

So Plugging in the information for above we know that the metric is 460800, but too see the math of how the router came up with it we can plug in the delay and bandwidth.

$$460800 = 256\ \times \left[\left(\frac{10^7}{10000}\right) + \left(\frac{800}{10}\right)\right]$$

Wide Metrics

-The EIGRP composite metric doesnt scale for high bandwidth interfaces. The lowest delay is 10microseconds, and interfaces like 10 Gigibit ones or ether channels will appear to EIGRP as a gigait interface.

- To Fix This EIGRP NAMED MODE must be used.

- *EIGRP Tracks but doesnt use bandwidth, delay, and reliability*

Bandwidth

Expressed in kilobits, is a static value for metric calculations only, doesnt reflect actual bandwidth of the link unless an admin configured it so, and the actual bandwidth of the interface cannot be altered by modifying the bandwidth value in EIGRP.

Delay

Microseconds, is a static value, cannot actually modify it by using static values as you would just be modifying the metric. EIGRP uses the sum of all outbound interfaces on the path to the destination network.

Reliability

Measured dynamically based on the observed error rate, and reflects the total outgoing error rates of the interfaces along the route. 255 is a 100% reliable link while 1 is a trash link.

Load

Dynamically measured using channel occupancy and reflects total outgoing load of all interface alonjg the route. 255 is a 100% lopaded link while 1 is a minimally loaded link.

Even if Load and Reliability are dynamic, EIGRP will not recalculate the route metric as these values change.

Variance

Monday, October 15, 2018 10:22

When a router learns multiple routes to a specific network via multiple routing processes, it installs the route with the lowest administrative distance in the routing table.

Sometimes the router must select a route from among many learned via the same routing process with the same administrative distance. In this case, the router chooses the path with the lowest cost (or metric) to the destination.

If the router receives and installs multiple paths with the same administrative distance and cost to a destination, load-balancing can occur. 4 Paths can be used by default and up to 6 paths configured is the maximum.

EIGRP can unequal load balance using the variance command

Basically Memorize This:

$$FD_2 \leq FD_1 \times V$$

Where

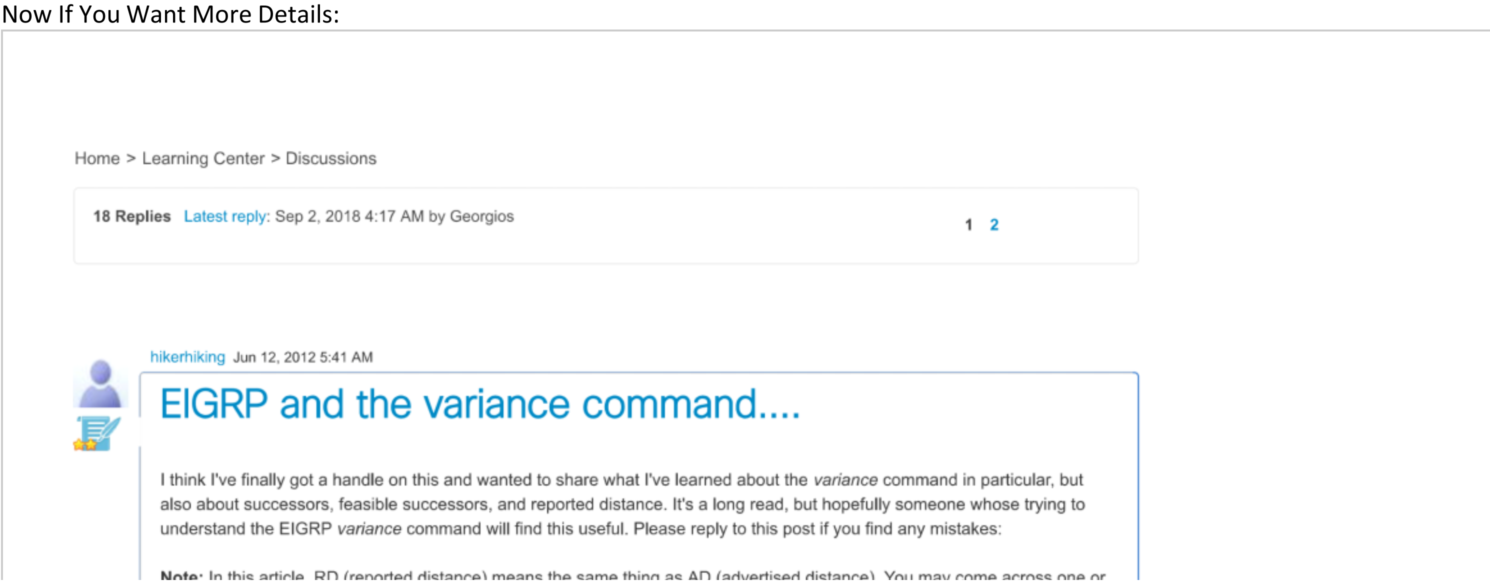
$FD_2 = Feasible\ Distance\ of\ Feasible\ Sucessor$

$FD_1 = Feasible\ Distance\ of\ Sucessor$

$V = Variance$

*IN ORDER TO USE THIS FORMULA THE ROUTE MUST BE A FEASIBLE SUCESSOR, IF THE ROUTE ISNT A FEASIBLE SUCESSOR YOU CAN NOT LOAD BALANCE

Now If You Want More Details:



Home > Learning Center > Discussions

18 Replies Latest reply: Sep 2, 2018 4:17 AM by Georgios 1 2

hikerhiking Jun 12, 2012 5:41 AM

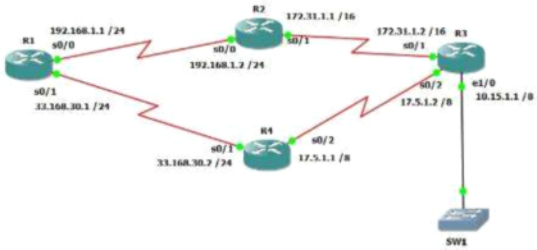
EIGRP and the variance command....

I think I've finally got a handle on this and wanted to share what I've learned about the *variance* command in particular, but also about successors, feasible successors, and reported distance. It's a long read, but hopefully someone whose trying to understand the EIGRP *variance* command will find this useful. Please reply to this post if you find any mistakes:

Note: In this article, RD (reported distance) means the same thing as AD (advertised distance). You may come across one or

I think I've finally got a handle on this and wanted to share what I've learned about the *variance* command in particular, but also about successors, feasible successors, and reported distance. It's a long read, but hopefully someone whose trying to understand the EIGRP *variance* command will find this useful. Please reply to this post if you find any mistakes:

Note: In this article, RD (reported distance) means the same thing as AD (advertised distance). You may come across one or the other in books or on web sites.



By default, in order to load-balance over different paths, the metrics of those paths must match **exactly**.In the network shown above, all bandwidth and delay settings were identical across both paths. I then went into R4 and changed the delay on s0/2 to 2001 (the default is 2000, or 20000 usec). When I did that, R1 would no longer used the path through R4, as the metrics were no longer equal. This proves that EIGRP wants to see metrics that are **exactly** the same across multiple paths before it will load balance (when **not** using the *variance* command).

Use of Variance

When using the variance command, eigrp will add a feasible successor to the route table **if** the feasible successor has a feasible distance that is less than or equal to the product of the feasible distance of the successor times the variance setting **and** the feasibility condition is met. In math terms:

FD (of the FS) **<=** FD (of the S) * *variance*

- FD – feasible distance
- FS – feasible successor
- S – successor

For variance to work, the route must **already** be a feasible successor. A route becomes a feasible successor by meeting the feasibility condition. Then, if variance is configured and the variance logic is met, the route is added to the routing table (this is explained in more detail later). When we say "the variance logic is met", we mean that the FD of the FS is less than or equal to the product of the successor's FD**variance-value* *(the value configured with the variance command):*

*The feasibility condition states that the reported distance (RD) of a route must be **less than** the feasible distance of the current successor route (the feasible successor). In other words, in order to become a feasible successor (FS), the reported distance of the feasible successor must be less than the feasible distance of the successor.*

RD< FD (of the FS)

To illustrate this, we'll perform all testing from R1's perspective, making changes in the delay settings in R4 only. With all routers set to their defaults (bandwidth and delay), R1 load balances across both paths to the 10.0.0.0 /8 network:

```
R1#ship route
D   10.0.0.0/8 [90/2707456] via 192.168.1.2, 00:04:00, Serial0/0
        [90/2707456]via 33.168.30.2, 00:04:00, Serial0/1
```



```
R1#showip eigrp top
P10.0.0.0/8, 2 successors, FD is 2707456
    via33.168.30.2 (2707456/2195456), Serial0/1
    via192.168.1.2 (2707456/2195456), Serial0/0
```

After changing R4 s0/2 to a delay of 3000 (or 30000 usec):

Notice that since the two paths to reach the 10.0.0. /8 network are no longer equal, R1 no longer load balances across both paths (it only has one route to the 10.0.0.0 /8 network, via R2). Also notice that the topology table still has two entries for the 10.0.0.0/8 network, because the reported distance of 2451456 is still less than the feasible distance of the successor route (2707456):

```
R1#showip route
D   10.0.0.0/8 [90/2707456] via192.168.1.2, 04:43:17, Serial0/0
```

(2707456):

```
R1#showip route
D   10.0.0.0/8 [90/2707456] via192.168.1.2, 04:43:17, Serial0/0

R1#showip eigrp topology
P10.0.0.0/8, 1 successors, FD is 2707456
    via192.168.1.2 (2707456/2195456),Serial0/0
    via33.168.30.2 (2963456/2451456),Serial0/1
```

Now, we work with the *variance* command to force eigrp to load-balance across two unequal-cost paths.

To determine what value to use for the *variance* command, we first divide the FD of the backup route (FS) by the FD of the successor route:

2963456/2707456= 1.095

Our multiplier is 1.095 (it's greater than 1, so we need to round up to the next whole number, or **2**.). So, we now go into the eigrp routing process and configure the *variance* command:

r1(config)#router eigrpx
r1(config-router)#variance 2

This tells EIGRP to take the feasible distance of every successor in the topology table and multiply that value by 2. Now, all feasible successors which are less than or equal to the FD*2 (that's the FD of the successor * 2) will be placed into the route table. Sticking with R1 and its path to the 10.0.0.0 /8 network, we take 2*2707456 (the FD of our successor), which give us 5414912. Now, any backup route (feasible successor) whose FD is less than 5414912 will be placed into the route table (as long as its reported distance is less than the feasible distance of the successor route, which will be explained further in a minute).

So, since the FD of the backup route (2963456) is less than 2*FD of the successor route (2707456*2 or 5414912), **AND**, the RD of its backup route (2451456) is less than the FD of the successor (2707456), that route is now placed back into the routing table:

```
R1#
D   10.0.0.0/8 [90/2707456] via 192.168.1.2, 00:00:02, Serial0/0
       [90/2963456]via 33.168.30.2, 00:00:02, Serial0/1

R1#ship eigrp topo
P10.0.0.0/8, 1 successors, FD is 2707456
    via192.168.1.2 (2707456/2195456), Serial0/0
    via33.168.30.2 (2963456/2451456), Serial0/1   → backup route
                   ( FD  /   RD  )
```

Next, I modified the delay in R4 s0/2 to 4000. Notice what happens (the variance is still set at 2). R1 is no longer load-balancing across R2 and R4, and the topology table no longer has a feasible successor (backup route) listed. Why? First, in order to view ALL the routes stored inthe topology table (successor, feasible successor , and non-successor routes), use the keyword **all-links** after the **show ip eigrp topology** command.

Now, we can see that R1 **does** know about the path through R4 (via 33.168.30.2). In addition, thebackup route's FD of 3219456 is definitely less than 5414912 that we calculated earlier, so it "should" load balance across the two paths, right? **Wrong!** The reason R1 is NOT sending traffic across R4 is because the reported distance of the backup route is equal to the FD of the successor route (see highlight in red below). Therefore, it does **not** meet the feasibility condition (i.e. the RD must be **less** than the FD of the successor) and will not be considered as a backup route, no matter how high you set the variance value. You could enter the command *variance 128* and it would have no effect, because there is no feasible successor in the topology table.

```
R1#show ip route
D   10.0.0.0/8 [90/2707456] via 192.168.1.2, 00:00:58, Serial0/0

R1#showip eigrp top
```



```
P10.0.0.0/8, 1 successors, FD is 2707456
    via192.168.1.2 (2707456/2195456), Serial0/0

R1#show ip eigrp top all-links
IP-EIGRPTopology Table for AS(5)/ID(192.168.1.1)

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

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

P10.0.0.0/8, 1 successors, FD is 2707456, serno 192
    via192.168.1.2 (2707456/2195456),Serial0/0
    via33.168.30.2 (3219456/2707456),Serial0/1
```

39425 Views Tags:

Sey Jun 11, 2012 12:48 PM (in response to hikerhiking)

1. Re: What I've learned about eigrp and the variance command....

Cool post, you did a great job! Way to go.

Actions Like (0)

tdistlists Dec 13, 2012 2:11 PM (in response to hikerhiking)

2. Re: EIGRP and the variance command....

Great post hikerhiking, well put.

I do have a question, however, regarding the *feasibility condition* that I never got a clear answer for; WHY must FD be **less than** RD? Why can't they be equal? In reading Dr. j. garcia-luna-aceves' literature, it seemed at one point it was FD <= RD. I can come up with loops where if RD > FD, and it's clear why no loops will be made if RD < FD, but I don't see why RD = FD is also loop-free.

Perhaps you can shed some light on this!
Thanks!

Actions Like (0)

Chandan Singh Takuli Jan 3, 2013 9:42 AM (in response to tdistlists)

3. Re: EIGRP and the variance command....

Hi Tdistlists,

RD/AD can never be equal to FD bcoz

FD = Distance between source roouter to advertising or reporting router + RD/AD.

WHY must FD be **less than** RD BCOZ, it may be psosible that the router who is providing RD/AD can be using the FD route.
let me Know if you have any query?

Actions Like (0)

Marko Milivojevic Jan 3, 2013 9:55 AM (in response to Chandan Singh Takuli)

4. Re: EIGRP and the variance command....

Yes it can be equal and that's the reason why FC requires it to be **less than** and not **less or equal to**.

As you know, EIGRP uses something called the vector metric. These are in fact, six separate metrics: bandwidth, delay, reliability, load, hop-count, and MTU. Of these six, four (bandwidth, delay, reliability, and load) can be used to calculate the composite metric. By default, only bandiwdth and delay are used. However, you can change this behavior by modifying the K-constants.

If you modify your K-constants to take only bandwidth into account when calculating the metric and in your network all links are of equal cost, calculated composite nmetric will be the same on all routers. The reason for this is because only the lowest bandwidth on the path is taken into account.

Converging By Going Active

Monday, October 15, 2018 10:49

When EIGRP removes a successor route and no FS route exists, the router begins a process by which the router discovers whether any loop-free alternative routes exist to reach that prefix.

This process is called going active on a route. Routes for which the router has a successor route, and no failure has yet occurred, remain in a passive state. Routes for which the successor route fails, and no feasible successor routes exist, move to an active state.

If a route goes Active, a “Query” is sent to all EIGRP neighbors (Except Stubs).
Once this Query sending is triggered, the “Active Timer” starts (3 minutes by default) in which the router waits to get a “Reply” from a neighbor.

After half the time of the Active Timer (90 seconds by default), a SIA-Query is sent to the upstream router to confirm it is Alive, if so it will send back an SIA-Reply. Usually if it does this, the upstream neighbor itself is waiting for a reply. Why? When he first got the request, he checked his routing table. If he had a feasible successor he would of responded immediately. Since he didn’t this means he didn’t have the route in the table so he sent a query out to his upstream neighbors. This process repeats and on large AS's this can lead to long wait times. There will be a trail of neighbors throughout the AS. If another 90 seconds goes by the route is officially Stuck in Active.

In essence, SIA is the state that occurs when a feasible successor cannot be found for a route due to lack of response from a neighbor. The reason for lack of response from a neighbor is typically the result of unrelated causes, but the effect is that the adjacency to that neighbor is dropped after a timeout occurs.

Troubleshoot SIA

The typical procedure for determining the cause of a stuck-in-active route is a manual process that also requires some fortunate timing. Assuming that the network engineer becomes aware of an active route in progress, they can attempt to trace the source of the stuck route as follows:

1. Log into the active router and run the show ip eigrp topology active show ip eigrp topology active show ip eigrp topology active command

2. Sort through the routes that are active and aging to see which neighbor routers have not responded

3. Log into each neighbor router that was queried and repeat the above process until you find the router or routers that were queried, but are not active and are not responding to the query – this is the source of the SIA error

4. Determine why the stuck router(s) are not responding to the active query

In most cases, it is uncommon for an administrator to notice an SIA event and be able to react in time to find the source of the problem before adjacencies are reset.

STUBs

Monday, October 15, 2018 12:22

Overview

The stub router concept came into existence to avoid EIGRP stuck in Active (SIA). The router on the very edge of the network will tell its neighbor "I AM THE LAST ROUTER ON THE ENTWORK, I HAVE NO NEIGHBORS BESIDES YOU and NO CONNECTIONS TO ANOTHER AS". Stubs also improves network stability, reduces resource utilization, and simplifies stub router configuration.

Stub routers also can effect queries. Instead of “deflecting” queries like summary routes, it simply signals the neighbors NOT to query the given router. This means the stub router could not be used as a **transit** path for any network. So when a path goes down the stub router neighbors will not query it and thus stub feature prevents queries from even being generated. It is especially effective in “start” network designs, such as hub-and-spoke networks.

There are different stub types in EIGRP.

1. Stub: default behavior is to share connected and summary routes.
2. Receive Only: Does not send any routes
3. Connected: Advertises all the connected routes to the single neighbor
4. Static: Advertises all static routes to the neighbor but still need to be redistributed into EIGRP to be advertised
5. Summary: Advertises all summary routes

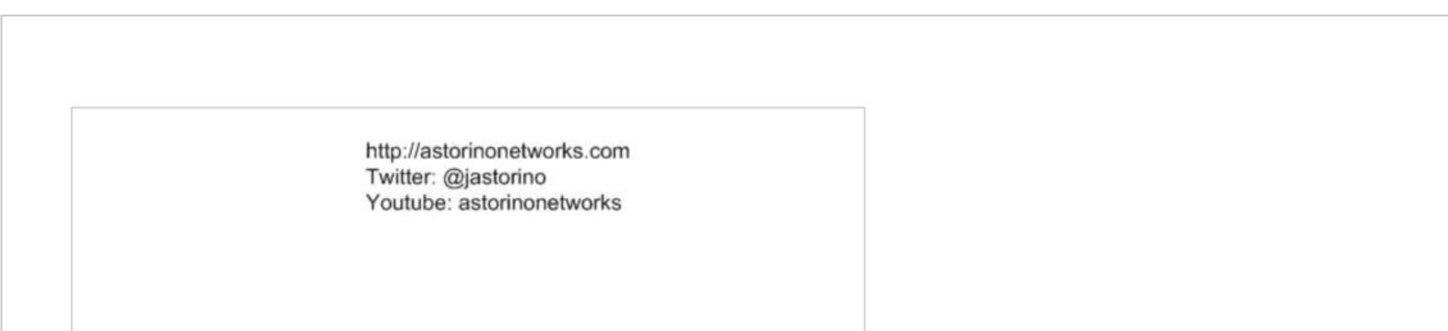
If you configure the **receive-only** option, you can’t include any of the other options on the above list.
If you configure the **static** option, you still must allow EIGRP to share the static routes by issuing the **redistribute static** command in the config-router context, or the router won’t share the routes. The same goes for the **connected** option.

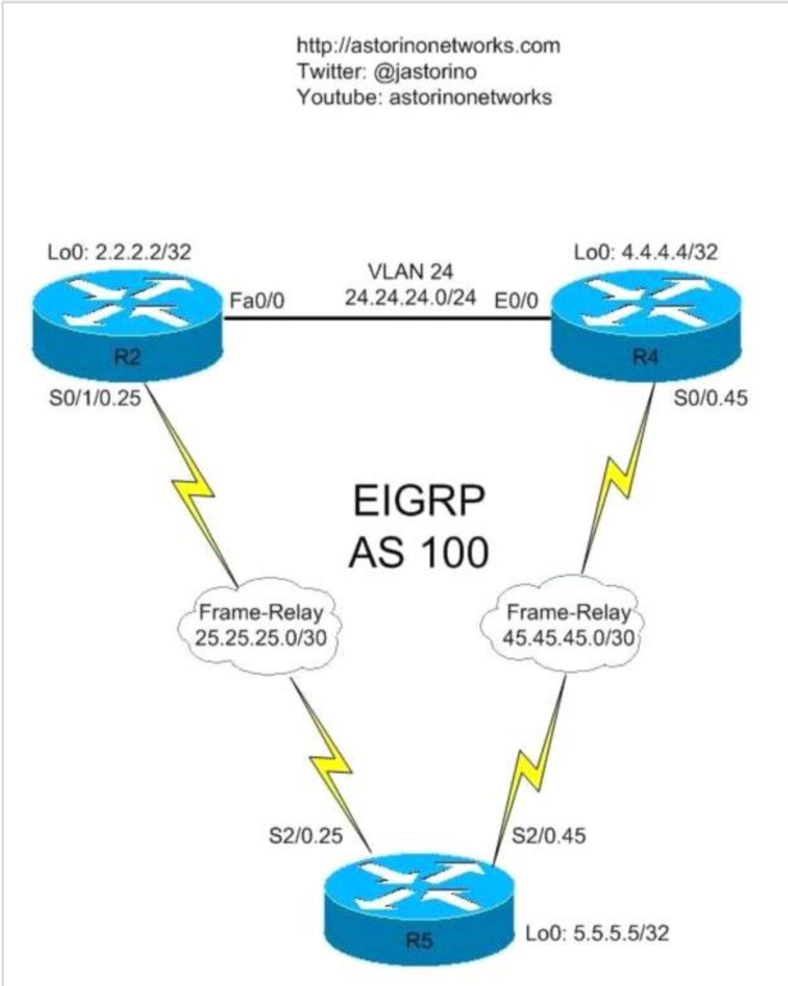
If a **network** statement does not include the connected routes you want to share, then you must issue the **redistribute connected** command.

One last aspect which may seem counterintuitive is that if you use the **redistribute** option, you are permitting the router to share redistributed routes, but you still must actually redistribute the routes for them to be shared.

If you choose the **summary option**, don’t forget to either manually create summary routes or enable **auto-summary**.

*AUTO SUMMARY IS OFF BY DEFAULT





Here we have a pretty common hub and spoke frame-relay setup. R2 and R4 are located at a corporate headquarters. R5 is dual homed back to corporate headquarters via two frame-relay links (nevermind that we are using point-to-point subinterfaces on the same physical link. The important thing logically here is that we have two connections and that was one way to accomplish that). The hub routers R2 and R4 send down only the default-route to R5.

IMPROVING NETWORK STABILITY

To understand how EIGRP stub routing will improve network stability, we need to understand some things about how EIGRP works normally.

As far as network stability, stub routing accomplishes two main things. Recall that when an EIGRP router loses a route and does not have a feasible successor for that route, it will go active and query all of it's neighbors. If the router doesn't hear back in 3 minutes, the route will be "stuck in active" and could cause routing issues in the rest of the network. Without stub routing turned on in our example, if R2 or R4 loses some route that sits up in the corporate network behind the hub routers, they may query R5 for those routes. This consumes resources on R5 and consumes precious bandwidth on the WAN links. It also opens the possibility of the route getting stuck in active if there are communication problems on the WAN. Furthermore, in this situation it really makes no sense because there are no other routers sitting behind R5 — It is a stub router by definition. When we configure R5 as an EIGRP stub, R5 will notify R2 and R4 that it is a stub router. In doing so, we prevent R2 and R4 from ever sending queries to R5 for active routes, so we eliminate that problem. If R5 should ever receive a query, it will simply reply "inaccessible".

The second thing in regards to network stability that EIGRP stub routing gives us is automatically preventing suboptimal transit routing. Imagine for a second that the LAN link over VLAN 24 between R2 and R4 was a GigabitEthernet link in a high speed corporate LAN. The main frame-relay link between R5 and R2 is a T1 line and the backup frame-relay link between R5 and R4 is 128k. Now imagine that there is some sort of problem on VLAN 24 and R2 loses EIGRP adjacency with R4. What would happen? R4 would continue sending EIGRP routes for the corporate LAN down to R5. R5 would in turn send them up to R2. R2 could then route traffic destined for somewhere else on the corporate network over the T1 link to R5. That could be a LOT of traffic for a T1 but then R5 is going to turn around and try to cram all of that down a 128k frame-relay link back up to R4!!! What you have is inherent disaster. In these cases you really don't want to allow intra-HQ traffic to route through your site WAN links. R5 really only needs to advertise in it's directly connected routes and nothing else. EIGRP stub routing also fixes this problem. When you enable R5 as an EIGRP stub, R5 by default will ONLY advertise to it's neighbors directly connected and summary routes by default. The feature can be tweeked to also include redistributed routes from other protocols or redistributed static routes. What this means is that when R5 learns routes from say R4 it will NOT relay those routes back up to R2. R5 will also not relay routes learned from R2 up to R4. This prevents R5 becoming a "transit" router for HQ routes.

REDUCING RESOURCE UTLIZATION

First of all, in a setup like this, R5 has no reason whatsoever to have any EIGRP routes other than the default route because no matter where packets need to go outside the local network, those packets need to go through R2 or R5. Anything more than the default route is a waste of R5's resources. That particular piece of the puzzle doesn't have to do with stub routing, but is generally something you will see in these topologies. The resource reduction part of this feature we have already talked about. Mainly, it means R5's resources do not become consumed by EIGRP queries. Since queries won't be sent over the WAN, we also consume WAN bandwidth. Also, R5 would send end up sending way less routes as well. Imagine that we did not filter and sent every HQ route down to R5 and that R5 learned 1000 routes from R4. Without stub routing, R5 would send those 1000 routes back up to R2.

SIMPLIFYING CONFIGURATION.

Simplifying configuration is a bit of as stretch in my opinion, but the argument is that because R5 automatically does not send anything except connected and summary routes, we no longer need to configure outbound distribute-lists to prevent R5 from becoming a transit router.

CONFIGURATION

EIGRP stub configuration is very straight forward. It requires no changes on the upstream routers. We simply configure the *eigrp stub* command on the remote stub router. When this happens, the stub router will actually notify it's neighbors that it is a stub router. This let's the neighbor routers know to not query the stub router for active routes. Let's look at this on R5

```
router eigrp 100
 network 25.25.25.2 0.0.0.0
 network 45.45.45.2 0.0.0.0
 no auto-summary
 eigrp stub
```

We enabled EIGRP on both frame-relay links. We enabled stub routing, which we said by default advertised directly connected and summary routes. However, if I look at R2, I don't see my 5.5.5.5 route

```
R2#sh ip route e
     4.0.0.0/32 is subnetted, 1 subnets
D       4.4.4.4 [90/156160] via 24.24.24.4, 01:03:16, FastEthernet0/0
     45.0.0.0/30 is subnetted, 1 subnets
D       45.45.45.0 [90/2172416] via 24.24.24.4, 00:01:45, FastEthernet0/0
D*   0.0.0.0/0 is a summary, 00:22:07, Null0
```

Although the stub feature does advertise connected routes by default, you still need network statements that cover those interfaces.

```
R5(config)#router eigrp 100
R5(config-router)#network 5.5.5.5 0.0.0.0
```

```
R2#sh ip route e
     4.0.0.0/32 is subnetted, 1 subnets
D       4.4.4.4 [90/156160] via 24.24.24.4, 01:04:23, FastEthernet0/0
     5.0.0.0/32 is subnetted, 1 subnets
D       5.5.5.5 [90/2297856] via 25.25.25.2, 00:00:17, Serial0/1/0.25
     45.0.0.0/30 is subnetted, 1 subnets
D       45.45.45.0 [90/2172416] via 24.24.24.4, 00:02:52, FastEthernet0/0
D*   0.0.0.0/0 is a summary, 00:23:14, Null0
```

Let's add a static route now, and modify the stub configuration on R5.

```
R5(config)#ip route 100.100.100.0 255.255.255.0 45.45.45.1
R5(config)#router eigrp 100
R5(config-router)#eigrp stub connected static
```

Now look at R2 again.

```
R2#sh ip route e
     4.0.0.0/32 is subnetted, 1 subnets
D       4.4.4.4 [90/156160] via 24.24.24.4, 01:06:23, FastEthernet0/0
     5.0.0.0/32 is subnetted, 1 subnets
D       5.5.5.5 [90/2297856] via 25.25.25.2, 00:00:24, Serial0/1/0.25
     45.0.0.0/30 is subnetted, 1 subnets
D       45.45.45.0 [90/2172416] via 24.24.24.4, 00:00:24, FastEthernet0/0
D*   0.0.0.0/0 is a summary, 00:25:14, Null0
```

I don't have the route to 100.100.100.0/24. Why? Even though I added the static keyword to the eigrp stub command, I still need to redistribute static to get the route to be advertised from R5

```
R5(config)#router eigrp 100
R5(config-router)#redistribute static metric 1 1 1 1 1
```

Now R2 should see our 100.100.100.0/24 route

```
R2#sh ip route e
     100.0.0.0/24 is subnetted, 1 subnets
D EX    100.100.100.0
           [170/2560512256] via 25.25.25.2, 00:00:52, Serial0/1/0.25
     4.0.0.0/32 is subnetted, 1 subnets
D       4.4.4.4 [90/156160] via 24.24.24.4, 01:08:22, FastEthernet0/0
     5.0.0.0/32 is subnetted, 1 subnets
```

```
D       5.5.5.5 [90/2297856] via 25.25.25.2, 00:02:23, Serial0/1/0.25
     45.0.0.0/30 is subnetted, 1 subnets
D       45.45.45.0 [90/2172416] via 24.24.24.4, 00:02:23, FastEthernet0/0
D*   0.0.0.0/0 is a summary, 00:27:13, Null0
```

VERIFICATION

We can verify a stub router configuration by look at the upstream neighbors. If a neighbor is configured as a stub router, it will report that information in the output of *show ip eigrp neighbor detail*

```
R2#show ip eigrp neighbor detail
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   25.25.25.2              Se0/1/0.25        13 00:04:59   42   252  0  105
   Version 12.4/1.2, Retrans: 1, Retries: 0, Prefixes: 3
   Stub Peer Advertising ( CONNECTED STATIC ) Routes
   Suppressing queries
0   24.24.24.4              Fa0/0             11 01:10:57    1   200  0  106
   Version 12.4/1.2, Retrans: 1, Retries: 0, Prefixes: 4
```

```
R4#sh ip eigrp neighbor detail
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   45.45.45.2              Se0/0.45          11 00:05:36  836  5000  0  106
   Version 12.4/1.2, Retrans: 1, Retries: 0, Prefixes: 3
   Stub Peer Advertising ( CONNECTED STATIC ) Routes
   Suppressing queries
0   24.24.24.2              Et0/0             14 01:11:34    1   200  0  109
   Version 12.4/1.2, Retrans: 2, Retries: 0, Prefixes: 4
```

That about does it for EIGRP stub routing.

SHARE THIS:

Tags: **ccie**, **CCNP**, **Cisco**, **eigrp**, **eigrp stub**, **stub**, **stub router**, **stub routing**

10 Comments

Anupam Nautiyal *says:*
May 4, 2012 at 7:47 am

SUMMARY ROUTES

Monday, October 15, 2018 11:03

Configuration Methods

These methods are available in order to advertise default route in EIGRP

1. Use default route and redistribution

2. Use Summary address: Manual summarization is where you summarize routes being advertised to a neighbor through an interface. So say you have a network with the following subnets:

```
10.10.1.0/24
10.10.2.0/24
10.10.3.0/24
10.10.4.0/24
```

You would make a classful network summary address of 10.10.1.0/22 and advertise it to the neighbor. This would let the neighbor and essentially the entire domain know that all traffic destined for 10.10.1.0/22 should be sent to you. This is not for the whole router but for that interface. AD of a summary route is 5

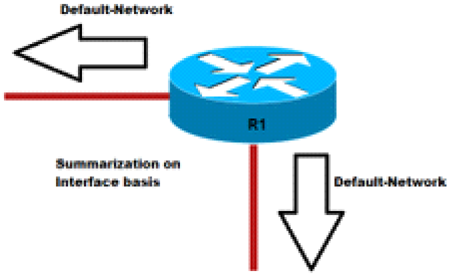
Overall Benefits

With redistribute static, it would show up as EIGRP external routes. With the IP summary-address command, it shows up as an local EIGRP route and you can further manipulate the AD with the ip summary address. But it will summarize only out that particular interface. Manual summarization reduces overhead of routing protocols.

Method 1

```
ip route 0.0.0.0 0.0.0.0 X.X.X.X
router eigrp 1
redistribute static
```

Method 2



```
R1(config)#interface gigabitEthernet 0/1
ip summary-address eigrp 1 0.0.0.0 0.0.0.0
```

Effects on Queries

In EIGRP route summarization can be used as a tool to limit the query scope. An EIGRP router will only forward queries about a network **only if it has an <u>exact match</u>** for that network in the routing table.

For Example

[A]---[B]---[C]---[D]

Say we have 4 routers connected, routers A-D.

A has the following Networks:

```
10.10.1.0/24
10.10.2.0/24
10.10.3.0/24
10.10.4.0/24
```

When One of A's interfaces goes down (say the 10.10.1.1) , B would receive a query asking about the route and he wouldn't know, so he'd send a query to C, and C would to D.

In this scenario we have 4 routers, so the query trail would end at D. But say it was a larger network, this can cause issues.

When summarization is done so 10.10.1.0/22 and one of the interfaces goes down, since B only has a summary it would automatically send a cost of infinity for that route. Thus stopping the query trail.

EIGRP for IPV6

Monday, October 15, 2018 11:13

Difference Between EIGRP for IPv4

Hellos are sourced from link local addresses and destined to FF02::A (All EIGRP Routers). This means neighbors do not have to share the same global prefix.

Auto summary is disabled by default for EIGRP for IPv6 as the concept of network classes don't exist

No Split Horizon because IPv6 supports multiple prefixes (addresses) per interface

EIGRP for IPV6 advertises prefixes and lengths rather than subnet mask IPv4 information

Neighbors Link Local is the next hop IP Address. Because of this in the neighbor table the interface the updates are learned from must be included.

Messages are sent in IPv6 Packets

Neighbors do not have to be on the same subnet

It is Possible to Run Both IPv4 and IPv6 EIGRP Processes simultaneously

Set-UP IPv6 First

```
Router(config) # ipv6 unicast-routing
Router (config-if)# ipv6 address X:X..X:X/prefix
```

Note that by configuring an IPV6 address you will have a global or unique-local IPv6 address and a link-local IPv6 address which is `FE80::interface-id`

The local-link IPv6 address is constructed automatically by concatenating FE80 with Interface ID as soon as IPv6 is enabled on the interface either by assigning an IPv6 address or simply by entering the command

```
Router(config-if)# ipv6 enable
```

```
R1E# conf t
R1E(config)# ipv6 unicast-routing
R1E(config)# ^Z

R1E#sh ipv6 interface
Ethernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::A8BB:CCFF:FE00:1E00
  No global unicast address is configured
  Joined group address(es):
```

```
R1E# conf t
R1E(config)# ipv6 unicast-routing
R1E(config)# ^Z

R1E#sh ipv6 interface
Ethernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::A8BB:CCFF:FE00:1E00
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1E00
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled

 <snip>
```

Setting Static Routes

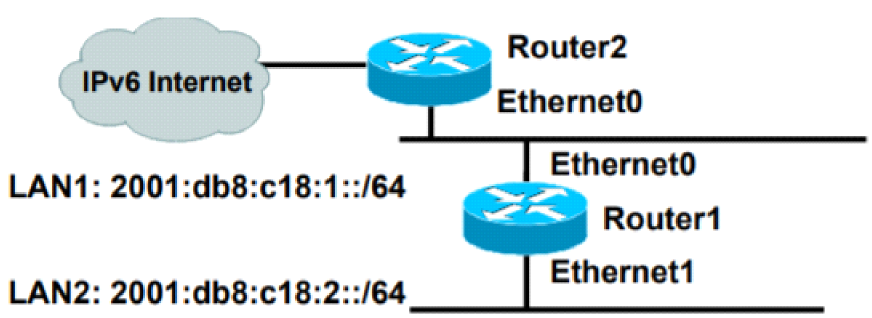
- **Syntax is:**

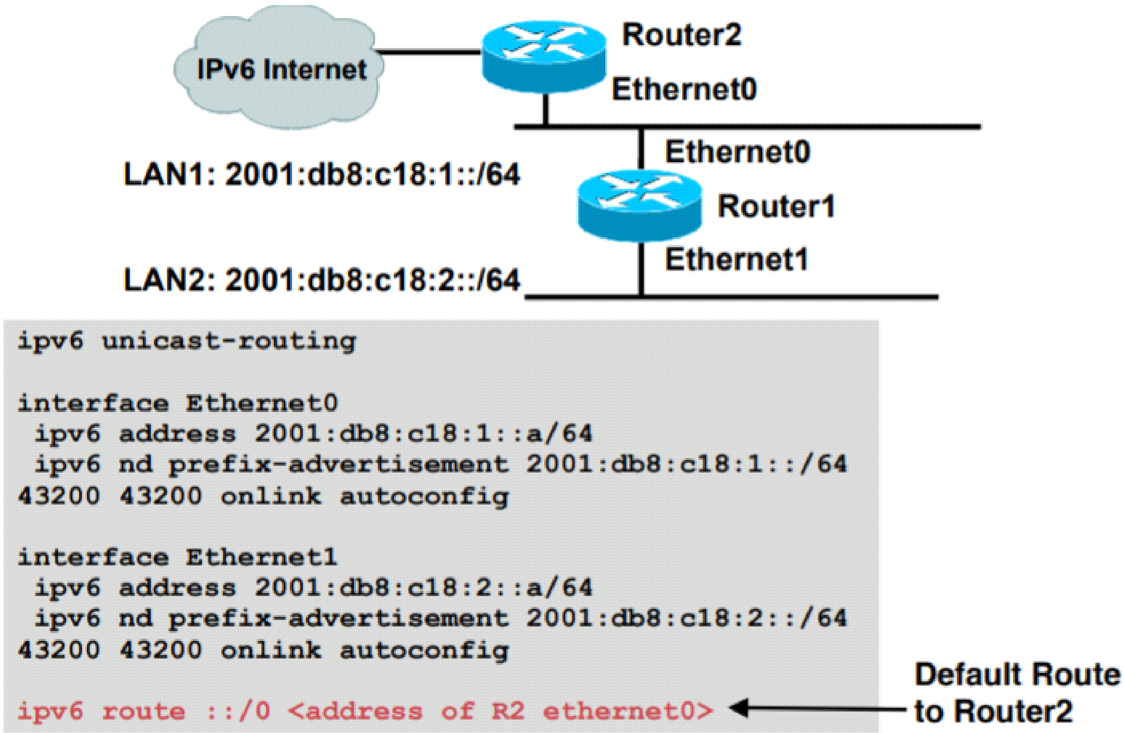
  ipv6 route ***ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number} [administrative-distance]***

- **Static Route**

  Router(config)# ipv6 route 2001:db8::/64 2001:db8:0:CC00::1 110

  Routes packets for network 2001:db8::/64 to a networking device at 2001:db8:0:CC00::1 with an administrative distance of 110





Setup EIGRP for IPv6

Router ID

***Criteria for deriving the router ID*:**

1. **Configured router ID:**

   Configured with router-id *router-id* command

2. **Highest Loopback IPv4 address:**
3. **Highest active interface IPv4 address:**

EIGRP (IPv4 amd IPv6) requires a 32-bit router ID.

If there is no 32-bit IPv4 address configured on the router as its running soley IPv6, then a router-id command is required since there is no 32-Bit loopback or interface. This is used to uniquely identify the router in EIGRP messages.

EIGRP Process

Use the command:

```
R2(config)# ipv6 router eigrp 2
R2(config-rtr)# eigrp router-id 2.2.2.2
R2(config-rtr)# exit
```

Now you have to go to the interface you want to enable EIGRP on and add the ipv6 eigrp command.

```
R2(config)# interface serial 0/0/0
R2(config-if)# ipv6 eigrp 2
```

```
R2(config-if)# exit
```

Static Default Route

To Add a static route use the ipv6 route command mentioned above.

```
R3(config)# ipv6 route ::/0 2001:db8:77::2
```

Then Go the EIGRP process to redistribute it.

```
R3(config)# ipv6 router eigrp 2
R3(config-rtr)# redistribute static
R3(config-rtr)# exit
```

Both EIGRP for IPv4 and IPv6

```
ipv6 unicast-routing
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ipv6 address FE80::1 link-local
 ipv6 address 2001:DB8:CAFE:1::1/64
 ipv6 eigrp 2
!
interface Serial0/0/0
 ip address 192.168.2.1 255.255.255.0
 ipv6 address FE80::1 link-local
 ipv6 address 2001:DB8:CAFE:2::1/64
 ipv6 eigrp 2
```

```
router eigrp 1
 network 192.168.1.0
 network 192.168.2.0
 eigrp router-id 1.1.1.1
!
ipv6 router eigrp 2
 eigrp router-id 1.1.1.1
```

NAMED EIGRP

Monday, October 15, 2018 11:49

OVERVIEW

Classic EIGRP requires separate EIGRP configuration modes and commands.

Named EIGRP Configuration is more recent method of configuring EIGRP for IPv4 and IPv6.

It is a single mode to gather all EIGRP commands in one place.

Uses address families unify EIGRP IPv4 and IPv6 configuration.

DETAILS

The traditional way to configure EIGRP requires various parameters to be configured under the interface and EIGRP configuration mode. In order to configure EIGRP IPV4 and IPv6, it is required to configure separate EIGRP instances. Traditional EIGRP does not support Virtual Routing and Forwarding (VRF) in IPv6 EIGRP implementations. With Named mode EIGRP, everything is configured at a single place under the EIGRP configuration and there are no restrictions as mentioned previously.

Address Family: where you configure IPv4 and IPv6 specific properties for a routing protocol.

Address-Family Interface: AF-Interface. This lets you configure interface level EIGRP configuration within EIGRP named mode. Cisco summarizes its usage on one of their command reference

Configuration for a DUAL STACK

```
R1(config)# ipv6 unicast-routing
R1(config)# router eigrp DUAL-STACK
R1(config-router)# address-family ?
  ipv4  Address family IPv4
  ipv6  Address family IPv6
R1(config-router)# address-family ipv4 unicast autonomous-system 1
R1(config-router-af)# eigrp router-id 1.1.1.1
R1(config-router-af)# network 192.168.1.0
R1(config-router-af)# network 192.168.2.0
R1(config-router-af)# exit-address-family
R1(config-router)#
```

The "DUAL-STACK" can be any string, it is just the name of a virtual instance and is local to the router only. It doesn’t have to match other routers

AS Numbers MUST match for neighbors in the same AS

Ipv6 unicast-routing must be configured prior to configuring an ipv6 address family

```
R2(config)# ipv6 unicast-routing
R2(config)# router eigrp DUAL-STACK
R2(config-router)# address-family ipv4 unicast autonomous-system 1
R2(config-router-af)# eigrp router-id 2.2.2.2
R2(config-router-af)# network 192.168.2.0
*Sep  7 04:47:00.832: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.2.1
   (Serial0/0/0) is up: new adjacency
R2(config-router-af)# network 192.168.3.0
R2(config-router-af)# exit-address-family
R2(config-router)# address-family ipv6 unicast autonomous-system 2
*Sep  7 04:47:46.536: %DUAL-5-NBRCHANGE: EIGRP-IPv6 2: Neighbor FE80::1
   (Serial0/0/0) is up: new adjacency
R2(config-router-af)# eigrp router-id 2.2.2.2
```

IPv6 adjacency with R1 comes up immediately because all IPv6 interfaces are enabled automatically however IPv4 ones come up with the network commands.

TOPOLOGY CONFIGURATION (DEFAULT ROUTES)

To redistribute a default route in named mode you have to do it in the topology configuration mode, which is a subset of the address family. Commands like variance, traffic share, metric, auto-summary are done here.

```
R3(config)# ip route 0.0.0.0 0.0.0.0 192.168.77.2
R3(config)# ipv6 route ::/0 2001:db8:feed:77::2 serial 0/1/0
R3(config)# router eigrp DUAL-STACK
R3(config-router)# address-family ipv4 unicast autonomous-system 1
R3(config-router-af)# topology base
R3(config-router-af-topology)# redistribute static
R3(config-router-af-topology)# exit-af-topology
R3(config-router-af)# exit-address-family
R3(config-router)# address-family ipv6 unicast autonomous-system 2
R3(config-router-af)# topology base
R3(config-router-af-topology)# redistribute static
R3(config-router-af-topology)#
```

DISABLE AN INTERFACE

```
R3(config)# router eigrp DUAL-STACK
R3(config-router)# address-family ipv6 unicast autonomous-system 2
R3(config-router-af)# af-interface serial 0/1/0
R3(config-router-af-interface)# shutdown
R3(config-router-af-interface)# exit-af-interface
R3(config-router-af)# exit-address-family
R3(config-router)# end
R3# show ipv6 interface brief | begin Serial0/1/0
Serial0/1/0            [up/up]
    FE80::3
    2001:DB8:77::1
```

PASSIVE MODE

```
R3(config)# router eigrp DUAL-STACK
R3(config-router)# address-family ipv4 unicast autonomous-system 1
R3(config-router-af)# af-interface gigabitethernet 0/0
R3(config-router-af-interface)# passive-interface
R3(config-router-af-interface)# exit-af-interface
R3(config-router-af)# exit-address-family
R3(config-router)# address-family ipv6 unicast autonomous-system 2
R3(config-router-af)# af-interface gigabitethernet 0/0
R3(config-router-af-interface)# passive-interface
R3(config-router-af-interface)# exit-af-interface
R3(config-router-af)#
```

SUMMARY ADDRESS

```
R3(config)# router eigrp DUAL-STACK
R3(config-router)# address-family ipv4 unicast autonomous-system 1
R3(config-router-af)# af-interface serial 0/0/1
R3(config-router-af-interface)# summary-address 10.0.0.0/8
R3(config-router-af-interface)# exit-af-interface
R3(config-router-af)# exit-address-family
R3(config-router)# address-family ipv6 unicast autonomous-system 2
R3(config-router-af)# af-interface serial 0/0/1
R3(config-router-af-interface)# summary-address 2001:db8:10::/48
R3(config-router-af-interface)#
```

SCRIPT

```
router eigrp DUAL-STACK
```

```
 !
 address-family ipv4 unicast
     autonomous-system 1
  network 10.0.0.0
  network 192.168.3.0
  network 192.168.4.0
  eigrp router-id 3.3.3.3
 !
 af-interface GigabitEthernet0/0
   passive-interface
 !
 af-interface Serial0/0/1
   summary-address 10.0.0.0 255.0.0.0
 !
 topology base
   redistribute static
!
!

address-family ipv6 unicast
      autonomous-system 2
   eigrp router-id 3.3.3.3
  !
  af-interface GigabitEthernet0/0
   passive-interface
  !
  af-interface Serial0/0/1
   summary-address 2001:DB8:10::/48
  !
  af-interface Serial0/1/0
   shutdown
  exit-af-interface
  !
  topology base
   redistribute static
```

Query Scoping

Monday, October 15, 2018 13:00

EIGRP is based on the concept of diffusing computations. When something changes in network topology, the routers that detect a loss of network prefix will send out EIGRP QUERY messages that propagate in circular waves similar to the ripples on water surface.

Every queried router will in turn query its neighbors and so on, until all routers that knew about the prefix affected. After this, the expanding circle will start collapsing back with EIGRP REPLY messages.

The maximum radius of that circle may be viewed as the **query scope**. From scalability standpoint, it is very important to know what conditions will limit the average query scope, as this directly impact the network stability.

You may compare the "query scope" with the concept of flooding domain in OSPF or ISIS. However, in contrast with the link-state protocols, you are very flexible with choosing the query scope boundaries, which is a powerful feature of EIGRP.

There are four conditions that affect query propagation. Almost all of them are based on the fact that query stops once the queried router cannot find the **exact** match for the requested subnet in its topology table. After this the router responds back that the network is unknown. Based on this behavior, the following will stop query from propagation

1) **Network summarization**. This could be considered as one of the most effective methods of query scoping. If router A sends a summary prefix to router B, then any query from A to B with regards to the subnets encompassed by the summary route will not find the exact match in B’s topology table. Thus queries are stopped on the routers that are one-hop from point of summarization. Given the fact that EIGRP allows for introducing summarization virtually everywhere you may easily partition your network into query domains. However, one important thing here – **this requires well-planned hierarchical addressing**, based on the Core-Edge model. Sending a default route to isolated stub domain could be considered an extreme case of summarization and is very effective.

2) **Stub routers.** This feature is different. Instead of “deflecting” queries, it simply signals the neighbors NOT to query the given router. This means the stub router could not be used as a **transit** path for any network. The stub router neighbors will not query it and thus stub feature prevents queries from even being generated. It is especially effective in “start” network designs, such as hub-and-spoke networks.

3) **Different EIGRP AS numbers**. EIGRP processes run independently from each other, and queries from one system don’t leak into another. However, if redistribution is configured between two processes a behavior similar to query leaking is observed. Consider that router R runs EIGRP processes for AS1 and AS2 with mutual

redistribution configured between the two processes. Assuming that a query from AS1 reaches R and bounces back, R is supposed to remove the route from the routing table. This in effect will trigger route removal from AS2 topology table, as redistribution is configured. Immediately after this R will originate a query into AS2, as the prefix becomes active. The query will travel across AS2 per the normal query propagation rules and eventually R will learn all replies and become passive for the prefixes. It's even possible for R to learn the path to the lost prefix via AS2 and re-inject it back into AS1 if the network topology permits that. As per the regular query propagation rules, you may use prefix summarization when redistributing routes to limit the query scope.

!!!!!!!SHE DIDN'T TEACH US THIS	**DO NOT READ**	**ONLY READ IF INTRESTED!!!!!!**

4) ***Route filtering****. You may filter EIGRP routes at any point using distribute-lists, route-maps etc. As soon as a route is filtered, the next-hop router will not learn it. When a query propagates to the next-hop, it will stop and return back. The use of route filtering is probably not as popular as route summarization, but could be used in some situations when you want to reduce the overall amount of queries but want to retain some routing information. In general, using proper route summarization should be enough.*

The above described are four main things that may limit query scoping. If you are using different AS numbers for this purpose, make sure you configure route summarization when doing redistribution, as otherwise you may get the "leaked query" effect.