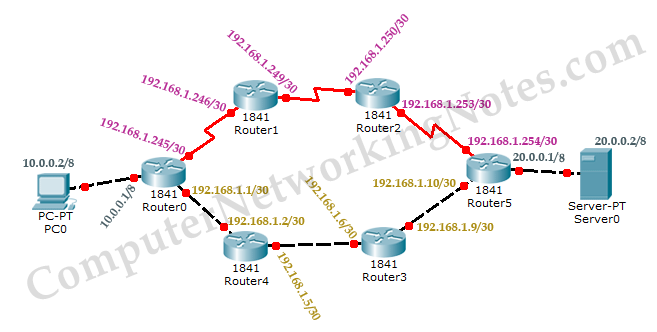
**Configuration for EIGRP Routing Protocol**

In this assignment we will see basic concepts of EIGRP such as Features and characteristics of EIGRP, Neighbor Table, Topology Table, Routing Table, Protocol Dependent Modules, Metric, RTP, DUAL, Autonomous System and Administrative Distance.

Also we we will see how two routers become EIGRP neighbor and maintain this neighborship. In order to become an EIGRP neighbor, three essential configuration values must be matched.

EIGRP uses composite metric calculation formula to calculate the best path. Bandwidth, reliability, delay, load and MTU are the components of formula. In this we explained these components with formula in easy language with examples.

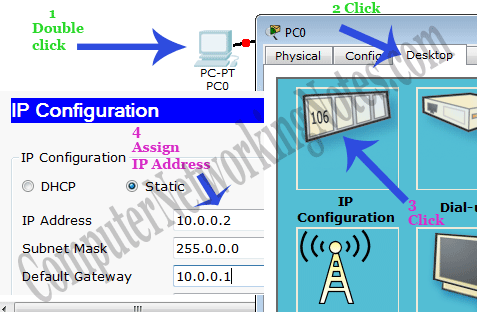
Create a topology as illustrate in following figure or download this pre-created topology.



|  |  |  |  |
| --- | --- | --- | --- |
| Device | Interface | IP Configuration | Connected with |
| PC0 | Fa0/0 | 10.0.0.2/8 | Router0’s Fa0/0 | |
| Router0 | Fa0/0 | 10.0.0.1/8 | PC0’s Fa0/0 | |
| Router0 | Fa0/1 | 192.168.1.1/30 | Router4’s Fa0/1 | |
| Router4 | Fa0/1 | 192.168.1.2/30 | Router0’s Fa0/1 | |
| Router4 | Fa0/0 | 192.168.1.5/30 | Router3’s F0/0 | |
| Router3 | Fa0/0 | 192.168.1.6/30 | Router4’s Fa0/0 | |
| Router3 | Fa0/1 | 192.168.1.9/30 | Router5’s Fa0/1 | |
| Router5 | Fa0/1 | 192.168.1.10/30 | Router3’s Fa0/1 | |
| Router5 | Fa0/0 | 20.0.0.1/8 | Serve0’s Fa0/0 | |
| Server | Fa0/0 | 20.0.0.2/8 | Router5’s Fa0/0 | |
| Router5 | Se0/0/0 | 192.168.1.254/30 | Router2’s Se0/0/0 | |
| Router2 | Se0/0/0 | 192.168.1.253/30 | Router5’s Se0/0/0 | |
| Router2 | Se0/0/1 | 192.168.1.250/30 | Router1’s Se0/0/1 | |
| Router1 | Se0/0/1 | 192.168.1.249/30 | Router2’s Se0/0/1 | |
| Router1 | Se0/0/0 | 192.168.1.246/30 | Router0’s Se0/0/0 | |
| Router0 | Se0/0/0 | 192.168.1.245/30 | Router1’s Se0/0/0 | |

**Assign IP address to PCs**

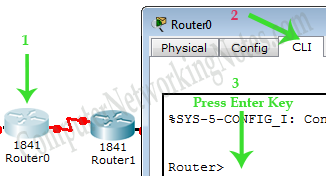
Double click **PC0** and click **Desktop** menu item and click **IP Configuration**. Assign IP address **10.0.0.2/8** to **PC0**.



Repeat same process for Server0 and assign IP address 20.0.0.2/8.

Assign IP address to interfaces of routers

Double click **Router0** and click **CLI** and press Enter key to access the command prompt of **Router0**.



Three interfaces **FastEthernet0/0, FastEthernet0/1** and **Serial0/0/0** of **Router0** are used in this topology. By default interfaces on router are remain administratively down during the start up.

We need to configure IP address and other parameters on interfaces before we could actually use them for routing. Interface mode is used to assign the IP address and other parameters. Interface mode can be accessed from global configuration mode. Following commands are used to access the global configuration mode.

Router>enable

Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#

From global configuration mode we can enter in interface mode. From there we can configure the interface. Following commands will assign IP address on FastEthernet0/0 and FastEthernet0/1.

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 10.0.0.1 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface fastEthernet 0/1

Router(config-if)#ip address 192.168.1.1 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#

**interface fastEthernet 0/0** command is used to enter in interface mode.

**ip address 10.0.0.1 255.0.0.0** command will assign IP address to interface.

**no shutdown** command will bring the interface up.

**exit** command is used to return in global configuration mode.

Serial interface needs two additional parameters **clock rate** and **bandwidth**. Every serial cable has two ends DTE and DCE. These parameters are always configured at DCE end.

We can use **show controllers interface** command from privilege mode to check the cable’s end.

Router#show controllers serial 0/0/0

Interface Serial0/0/0

Hardware is PowerQUICC MPC860

DCE V.35, clock rate 2000000

[Output omitted]

Fourth line of output confirms that DCE end of serial cable is attached. If you see DTE here instead of DCE skip these parameters.

Now we have necessary information let’s assign IP address to serial interface.

Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface serial 0/0/0

Router(config-if)#ip address 192.168.1.245 255.255.255.252

Router(config-if)#clock rate 64000

Router(config-if)#bandwidth 64

Router(config-if)#no shutdown

Router(config-if)#exit

**Router#configure terminal** Command is used to enter in global configuration mode.

**Router(config)#interface serial 0/0/0** Command is used to enter in interface mode.

**Router(config-if)#ip address 192.168.1.245 255.255.255.252** Command assigns IP address to interface. For serial link we usually use IP address from /30 subnet.

**Router(config-if)#clock rate 64000**

In real life environment this parameter controls the data flow between serial links and need to be set at service provider’s end. In lab environment we need not to worry about this value. We can use any valid rate here.

**Router(config-if)#bandwidth 64**

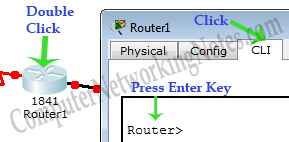
Bandwidth works as an influencer. It is used to influence the metric calculation of EIGRP or any other routing protocol which uses bandwidth parameter in route selection process.

**Router(config-if)#no shutdown** Command brings interface up.

**Router(config-if)#exit** Command is used to return in global configuration mode.

We will use same commands to assign IP addresses on interfaces of remaining routers. We need to provided clock rate and bandwidth only on DCE side of serial interface. Following command will assign IP addresses on interface of Router1.

**Router1**



Router>enable

Router#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface serial 0/0/0

Router(config-if)#ip address 192.168.1.246 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface serial 0/0/1

Router(config-if)#ip address 192.168.1.249 255.255.255.252

Router(config-if)#clock rate 64000

Router(config-if)#bandwidth 64

Router(config-if)#no shutdown

Router(config-if)#exit

We will use same commands to assign IP addresses on interfaces of remaining routers.

**Router2**

Router>enable

Router#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface serial 0/0/1

Router(config-if)#ip address 192.168.1.250 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface serial 0/0/0

Router(config-if)#ip address 192.168.1.253 255.255.255.252

Router(config-if)#clock rate 64000

Router(config-if)#bandwidth 64

Router(config-if)#no shutdown

Router(config-if)#exit

**Router5**

Router>enable

Router#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 20.0.0.1 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface fastEthernet 0/1

Router(config-if)#ip address 192.168.1.10 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#

Router(config)#interface serial 0/0/0

Router(config-if)#ip address 192.168.1.254 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

**Router3**

Router>enable

Router#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 192.168.1.6 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface fastEthernet 0/1

Router(config-if)# ip address 192.168.1.9 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#

**Router4**

Router>enable

Router#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip address 192.168.1.5 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface fastEthernet 0/1

Router(config-if)# ip address 192.168.1.2 255.255.255.252

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#

Great job we have finished our half journey. Now routers have information about the networks that they have on their own interfaces. Routers will not exchange this information between them on their own. We need to implement EIGRP routing protocol that will insist them to share this information.

To be on same track I have uploaded my practice topology on our server. Use this if you want to skip the IP configuration part.

**Configure EIGRP routing protocol**

Enabling EIGRP is a two steps process:-

1. Enable EIGRP routing protocol from global configuration mode.
2. Tell EIGRP which interfaces we want to include.

For these steps following commands are used respectively.

Router(config)# router eigrp autonomous\_system\_#

Router(config-router)# network IP\_network\_# [subnet\_mask]

**Router(config)# router eigrp autonomous\_system\_#**

This command will enable EIGRP routing protocol in router. We can use any ASN (Autonomous System Number) from 1 to 65,535. In order to become EIGRP neighbors this number must be same on all participates.

**Router(config-router)# network IP\_network\_# [subnet\_mask]**

This command allows us to specify the local interfaces which we want to include in EIGRP. Basically we define a range of addresses and router search for these addresses in local interfaces. If match found EIGRP will be enabled on that interface. Once enabled, EIGRP will starts advertising about the connected subnets with that interface.

We have two options while defining the range of addresses with **network** command

1. Without wildcard mask
2. With wildcard

**Without wildcard**

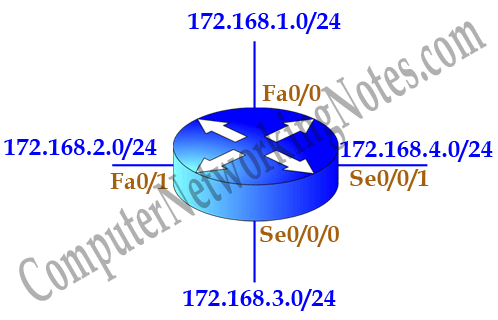
Choosing this option allows us to configure the classful network. This option is very straightforward. All we need to do is, type the network ID with network command. For example network 172.168.0.0 command will enable EIGRP on all interfaces which belong to network 172.168.0.0.

What if I type network number instead of network ID?

Well in this situation EIGRP will automatically convert it back to network ID in which this network number is resides. For example 172.168.1.1 will be converted back in 172.168.0.0.

This creates another query. Why it will be converted in 172.168.0.0 instead of 172.168.1.0?

Answer of this question is hidden in classful configuration. In classful configuration EIGRP will match network addresses with in default boundary. Consider following figure



We have four networks 172.168.1.0/24, 172.168.2.0/24, 172.168.3.0/24 and 172.168.4.0/24 Subnetted from single class B network 172.168.0.0/16. Classful configuration does not understand the concept of Subnetting. In classful configuration all these networks belong to a single network. Classful configuration works only with in default boundary of mask. Default boundary of this address is 16 bits. So it will match only first 16 bits (172.168.x.y) of network address.

If we want excludes serial interfaces from EIGRP, we need to configure network command with more specific information.

**With wildcard**

In this option we provide wildcard mask along with network ID. Wildcard mask allows us to match exact networks. With wildcard we are no longer limited with default boundaries. We can match Subnetted networks as well as default networks.

For example we were tasked to exclude serial interfaces in above configuration. We can use a wildcard mask of 0.0.0.255 to match the subnet mask of /24.

Router(config-router)# network 172.168.1.0 0.0.0.255

Router(config-router)# network 172.168.2.0 0.0.0.255

Above commands will ask router to match /24 bits of address instead of default /16 bits. Now router will look for 172.168.1.x and 172.168.2.x network. Our serial interfaces have 172.168.3.0/24 and 172.168.4.0/24 networks which do not fall in these search criteria.

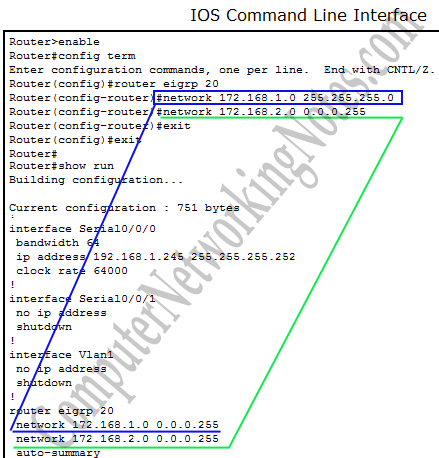
If you are unfamiliar with wildcard mask, I suggest you to read our tutorials on ACL where we explained wildcard mask in detail with examples.

Until you learn wildcard mask, use subnet mask in the place of wildcard mask. Following commands are also valid and do the same job by matching /24 bits of address.

Router(config-router)# network 172.168.1.0 255.255.255.0

Router(config-router)# network 172.168.2.0 255.255.255.0

Subnet mask is a substitute, not a replacement of wildcard mask. When we use Subnet mask, router converts them in wildcard mask before searching for associated interfaces. We can look in running configuration to know what exactly being used by router.



**EIGRP configuration**

Now we know the essential commands for configuration. Let’s implement them in our network.

**Router0**

Router(config)#router eigrp 20

Router(config-router)#network 10.0.0.0 0.0.0.255

Router(config-router)#network 192.168.1.244 0.0.0.3

Router(config-router)#network 192.168.1.0 0.0.0.3

Router(config-router)#

**Router1**

Router(config)#router eigrp 20

Router(config-router)#network 192.168.1.244 0.0.0.3

Router(config-router)#

%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.245 (Serial0/0/0) is up: new adjacency

Router(config-router)#network 192.168.1.248 0.0.0.3

Router(config-router)#

**Router2**

Router(config)#router eigrp 20

Router(config-router)#network 192.168.1.248 0.0.0.3

Router(config-router)#

%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.249 (Serial0/0/1) is up: new adjacency

Router(config-router)#network 192.168.1.252 0.0.0.3

Router(config-router)#

As I mentioned earlier, we can use both wildcard mask and subnet mask with network command. We have used wildcard mask for above routers. In remaining routers we will use subnet mask.

**Router5**

Router(config)#router eigrp 20

Router(config-router)#network 20.0.0.0 255.0.0.0

Router(config-router)#network 192.168.1.252 255.255.255.252

Router(config-router)#

%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.253 (Serial0/0/0) is up: new adjacency

Router(config-router)#network 192.168.1.8 255.255.255.252

Router(config-router)#

**Router3**

Router(config)#router eigrp 20

Router(config-router)#network 192.168.1.8 255.255.255.252

Router(config-router)#

%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.10 (FastEthernet0/1) is up: new adjacency

Router(config-router)#network 192.168.1.4 255.255.255.252

Router(config-router)#

**Router4**

Router(config)#router eigrp 20

Router(config-router)#network 192.168.1.4 255.255.255.252

Router(config-router)#

%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.6 (FastEthernet0/0) is up: new adjacency

Router(config-router)#network 192.168.1.0 255.255.255.252

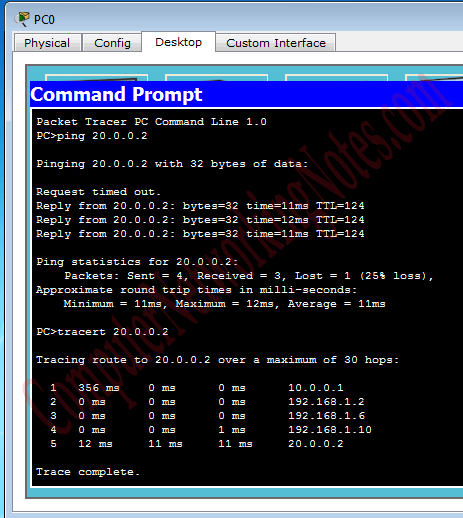
Router(config-router)#

%DUAL-5-NBRCHANGE: IP-EIGRP 20: Neighbor 192.168.1.1 (FastEthernet0/1) is up: new adjacency

Router(config-router)#

That’s it. Our network is ready to take the advantage of EIGRP routing. To verify the setup we will use ping command. **ping** command is used to test the connectivity between two devices. We have two routes between source and destination. **tracert** command is used to know the route which is used to get the destination.

Access the command prompt of PC1 and use ping command to test the connectivity from Server0. After that use **tracert** command to print the taken path.



Good going we have successfully implemented EIGRP routing protocol in our network. For cross check we have uploaded a configured topology on our server. You can use this if not getting same output.

EIGRP protocol automatically manages all routes for us. If one route goes down, it will automatically switch to another available route. To explain this process more clearly we have added one additional route in our network.

Currently there are two routes between PC0 and Server.

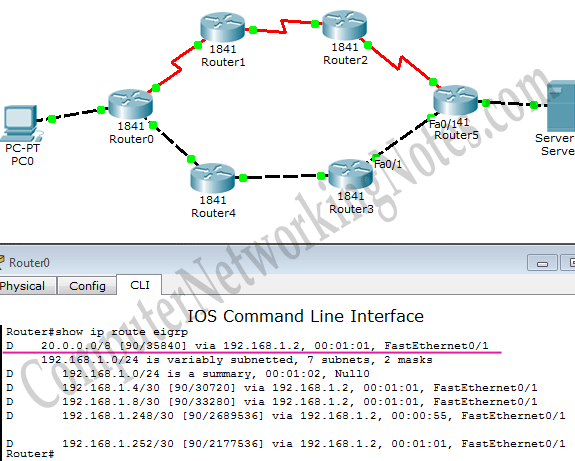
**Route 1**

PC0 <==> Router0 <==> Router4 <==> Router3 <==> Router5 <==> Server0

**Route 2**

PC0 <==> Router0 <==> Router1 <==> Router2 <==> Router5 <==> Server0

By default EIGRP uses the route that has low metric value. Our path separates from Router0, so let’s see which route it takes to deliver the packet of 20.0.0.0 network. **show ip route eigrp** command will list all available routes.



**Output of show ip route eigrp Explained**

**D: -** It indicates that route is learned by EIGRP. Cisco chose letter D for EIGRP, because letter E was already taken by Exterior Gateway Protocol (EGP).

**20.0.0.0/8: -** It is our destination network.

**90: -** Administrative distance of EIGRP.

**35840: -** Is the metric value of this route calculated by EIGRP

**Via 192.168.1.2: -** IP address of the next hop.

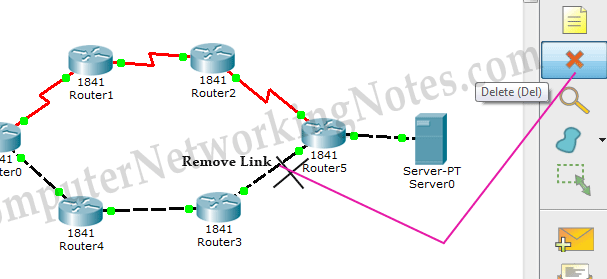
**00:01:01: -** How long this route was learned (Age of route)

**FastEthernet1: -** Exit interface of this router to get the next hop.

You may wonder where Route2 is in this output. Well EIGRP puts only the best route in routing table. Route2’s metric value is higher than Route1. Till route1 is available, it will not insert route2 in routing table. When route1 is down, it will look for next possible route. If other routes are available, it will replace current route with new route which has the lowest metric value. We can watch this process live with **debug eigrp fsm** command. On debug process on Router0.

Router# debug eigrp fsm

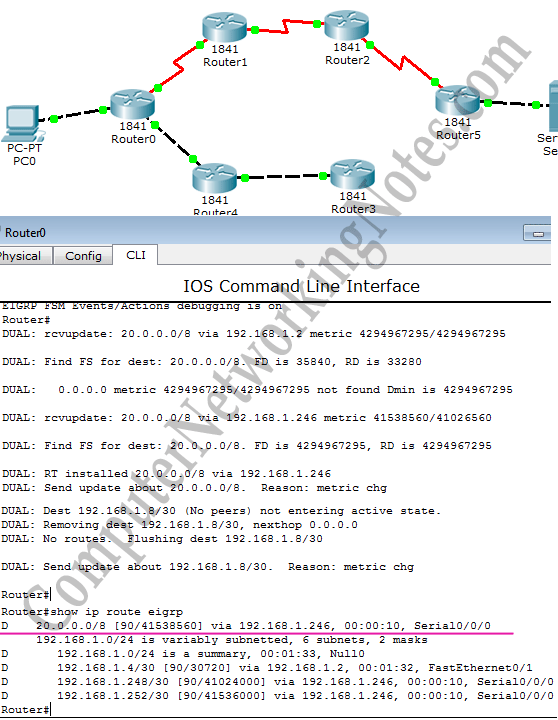
Now suppose route1 is down. We can simulate this situation by removing the cable attached between Router3 [Fa0/1] and Router5 [Fa0/1].



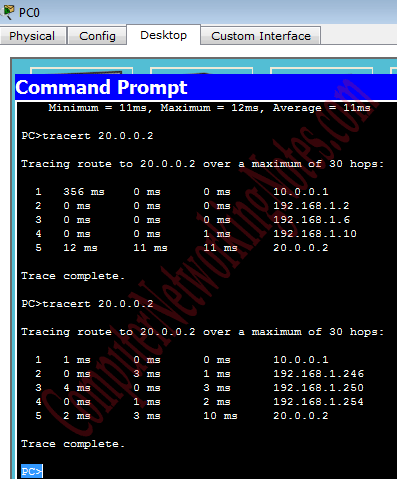
Okay our primary route went down. What will be happen now?

EIGRP will look in topology table for next available routes. If single alternative is available, it will be selected. If multiple routes are available, it will select the route with the lowest metric value.

We can use **show ip route eigrp** command again to see the selected route.



Run **tracert** command again from PC0 to verify the change.



That’s all for this article. Before closing just do a quick recap of important commands.

**EIGRP configuration commands cheat sheet**

|  |  |
| --- | --- |
| Command | Description |
| Router(config)#router eigrp 20 | Enable EIGRP with AS number 20. AS number must be same on all routers to become EIGRP neighbor. |
| Router(config-router)#network 10.10.0.0 | Enable EIGRP on interfaces which belongs to network 10.0.0.0/8. [Classful implementation]. |
| Router(config-router)#network 10.10.0.0 0.0.255.255 | Enable EIGRP on interfaces which belongs to network 10.10.0.0/16. [Classless implementation – Wildcard mask method]. |
| Router(config-router)#network 10.10.0.0 255.255.0.0 | Enable EIGRP on interfaces which belongs to network 10.10.0.0/16. [Classless implementation – Subnet mask method]. |
| Router(config-router)#no network 10.10.0.0 | Disable EIGRP on interfaces which belongs to network 10.0.0.0/8. |
| Router(config-router)#no network 10.10.0.0 0.0.255.255 | Disable EIGRP on interfaces which belongs to network 10.10.0.0/16. |
| Router(config-router)#no network 10.10.0.0 255.255.0.0 | Disable EIGRP on interfaces which belongs to network 10.10.0.0/16. |
| Router(config-router) #metric weights tos k1 k2 k3 k4 k5 | Enable/Disable K values used in metric calculation formula.  Default values are tos=0, k1=1, k2=0, k3=1, k4=0, k5=0  Tos(type of service), K1(bandwidth), K2(load), K3(delay), K4(reliability), K5(MTU).  By default only K1 and K3 are enabled. |
| Router(config-router)#auto-summary | Enable auto summarization feature of EIGRP. ( Default – disable ) |
| Router(config-router)#no auto-summary | Disable auto summarization feature of EIGRP. |
| Router(config)#no router eigrp 20 | Disable EIGRP routing process 20. |
| Router(config-if)#bandwidth 64 | Set bandwidth to 64Kbps. Used to influence the metric calculation. |
| Router#show ip eigrp neighbors | Display the neighbor table in brief. |
| Router#show ip eigrp neighbors detail | Display the neighbor table in detail. Used to verify whether a neighbor is configured as stub router or not. |
| Router#show ip eigrp interfaces | Display information about all EIGRP interfaces. |
| Router#show ip eigrp interfaces serial 0/0 | Display information about a particular EIGRP interface. |
| Router#show ip eigrp interfaces 20 | Display information about EIGRP interfaces running AS process 20. |
| Router#show ip eigrp topology | Displays the topology table. |
| Router#show ip eigrp traffic | Displays the number and type of packets sent and received. |
| Router#show ip route eigrp | Display EIGRP route from routing table. |
| Router#debug eigrp fsm | Displays the events or actions related to feasible successor metrics (FSM). |
| Router#debug eigrp packet | Displays the events or actions related to EIGRP packets. |
| Router#no debug eigrp fsm | Turn off debug message related to feasible successor metrics (FSM). |
| Router#no debug eigrp packet | Turn off debug message related to EIGRP packets. |