Configuration Exercise: Configuring and Tuning EIGRP

In this exercise, you first configure EIGRP and investigate its default behavior. You next configure EIGRP summarization, a stub, and a default route.

Introduction to the Configuration Exercises

This book uses Configuration Exercises to help you practice configuring routers with the commands and topics presented. If you have access to real hardware, you can try these exercises on your routers. See Appendix B, "Configuration Exercise Equipment Requirements and Backbone Configurations," for a list of recommended equipment and initial configuration commands for the backbone routers. However, even if you do not have access to any routers, you can go through the exercises, and keep a log of your own running configurations, or just read through the solution. Commands used and solutions to the Configuration Exercises are provided within the exercises.

In the Configuration Exercises, the network is assumed to consist of two pods, each with four routers. The pods are interconnected to a backbone. You configure pod 1. No interaction between the two pods is required, but you might see some routes from the other pod in your routing tables in some exercises if you have it configured. In most of the exercises, the backbone has only one router; in some cases, another router is added to the backbone. Each Configuration Exercise assumes that you have completed the previous chapters' Configuration Exercises on your pod.

NOTE

Throughout this exercise, the pod number is referred to as *x*, and the router number is referred to as *y*. Substitute the appropriate numbers as needed.

Exercise Objectives

The objectives of this exercise are as follows:

- Set up EIGRP
- Investigate the default behavior of EIGRP
- Optimize the EIGRP configuration

Visual Objective

Figure 3-37 illustrates the topology used and what you will accomplish in this exercise.

Figure 3-37 EIGRP Configuration Exercise Topology

Command List

In this exercise, you use the commands in Table 3-10, listed in logical order. Refer to this list if you need configuration command assistance during the exercise.

Table 3-10. EIGRP Configuration Exercise Commands

Command	Description
(config)#**router eigrp 1**	Enters configuration mode for EIGRP in autonomous system 1
(config-router)#**network 10.*x*.0.0 0.0.255.255**	Specifies that EIGRP should run within network 10.*x*.0.0/16
(config-router)#**no auto-summary**	Turns off automatic summarization at classful network boundaries
#**show ip protocols**	Displays the parameters and current state of all the active routing protocol processes
#**debug ip eigrp**	Displays EIGRP updates
(config-if)#**ip summary-address eigrp 1 10.*x*.0.0 255.255.0.0**	Creates and advertises a summary route 10.*x*.0.0/16 for EIGRP autonomous system 1 out of this interface
(config-router)#**eigrp stub**	Specifies that this router should behave as an EIGRP stub router
#**show ip eigrp neighbors detail**	Displays detailed EIGRP neighbor information
(config-if)#**ip summary-address eigrp 1 0.0.0.0 0.0.0.0**	Creates and advertises a default route for EIGRP autonomous system 1 out of this interface and suppresses all other specific routes
#**show ip eigrp topology**	Displays the EIGRP topology table
#**show ip eigrp traffic**	Displays EIGRP traffic statistics
#**show ip eigrp interfaces**	Displays information about interfaces configured for EIGRP
#**show ip eigrp neighbors**	Displays EIGRP neighbor information

CAUTION

Although the command syntax is shown in this table, the addresses shown are typically for the P*x*R1 and P*x*R3 routers. Be careful when addressing your routers! Refer to the exercise instructions and the appropriate visual objective diagram for addressing details.

NOTE

The exercise tasks include answers and solutions. Some answers cover multiple steps; the answers are given after the last step to which that answer applies.

Task 1: Configuring Basic EIGRP

In this task, you configure EIGRP on each router in your pod so that there are EIGRP routes from the core, between edge routers, and between the edge and the internal routers. Follow these steps:

- **Step 1** Shut down the serial interface between the internal routers (s0/0/0 on PxR3 and PxR4); this link is not used in this exercise.

  Solution:

  The following shows the required step on the P1R3 router:

```
P1R3(config)#int s0/0/0
P1R3(config-if)#shutdown
```

- **Step 2** Configure EIGRP on each router in your pod in autonomous system 1, using the appropriate network and wildcard values to include all interfaces in the EIGRP routing process. Disable autosummarization on the edge routers.

  Solution:

  The following shows the required steps on the P1R1 and P1R3 routers:

```
P1R1(config)#router eigrp 1
P1R1(config-router)#network 10.1.0.0 0.0.255.255
P1R1(config-router)#network 172.31.1.0 0.0.0.255
P1R1(config-router)#no auto-summary

P1R3(config-if)#router eigrp 1
P1R3(config-router)#network 10.1.0.0 0.0.255.255
```

- **Step 3** Verify that the routing protocols are set up correctly using the **show ip protocols** command. Make sure that the autonomous system number is correct and that all neighbors are exchanging routes.

  Solution:

  The following shows example output on the P1R1 router:

```
P1R1#show ip protocols
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
```

```
  Routing for Networks:
    10.1.0.0/16
    172.31.1.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.1.3              90      00:00:37
    10.1.0.2              90      00:00:35
    172.31.1.3            90      00:00:35
  Distance: internal 90 external 170
P1R1#
```

- **Step 4** Verify that routes from other routers in your pod and from the backbone router BBR1 are being recognized via EIGRP on each router.

  Solution:

  The following shows example output on the P1R1 router:

```
P1R1#show ip route
<output omitted>
Gateway of last resort is not set

     172.31.0.0/24 is subnetted, 2 subnets
D       172.31.2.0 [90/21024000] via 172.31.1.3, 00:04:41, Serial0/0/0
C       172.31.1.0 is directly connected, Serial0/0/0
     10.0.0.0/24 is subnetted, 4 subnets
D       10.1.2.0 [90/20514560] via 10.1.0.2, 00:10:08, Serial0/0/1
C       10.1.1.0 is directly connected, FastEthernet0/0
C       10.1.0.0 is directly connected, Serial0/0/1
D       10.254.0.0 [90/20514560] via 172.31.1.3, 00:04:42, Serial0/0/0
P1R1#
```

  The highlighted routes are being learned by EIGRP.

- **Step 5** Use **debug ip eigrp** on the internal routers in your pod to monitor the EIGRP queries.
- **Step 6** Shut down the serial interface between the edge routers (the S0/0/1 interface on P*x*R1 and P*x*R2).
- **Step 7** View the EIGRP queries sent to the internal routers.

  Solution:

  The following shows the required command on the P1R3 router, the configuration on the P1R1 router, and example output on the P1R3 router:

```
P1R3#debug ip eigrp
IP-EIGRP Route Events debugging is on
P1R3#

P1R1(config)#int s0/0/1
P1R1(config-if)#shutdown

P1R3#
*Mar  6 02:19:11.363: IP-EIGRP(Default-IP-Routing-Table:1): Processing
incoming QUERY
  packet
```

```
*Mar  6 02:19:11.367: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.0.0/24 M
  4294967295 - 0 4294967295 SM 4294967295 - 0 4294967295
*Mar  6 02:19:11.367: IP-EIGRP(Default-IP-Routing-Table:1): 10.1.0.0/24
routing table
  not updated thru 10.1.1.1
*Mar  6 02:19:11.367: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.2.0/24 M
  4294967295 - 20000000 4294967295 SM 4294967295 - 20000000 4294967295
*Mar  6 02:19:11.367: IP-EIGRP(Default-IP-Routing-Table:1): 10.1.2.0/24
routing table
  not updated thru 10.1.1.1
*Mar  6 02:19:11.387: IP-EIGRP(Default-IP-Routing-Table:1): 10.1.0.0/24-
not in IP
  routing table
*Mar  6 02:19:11.387: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.0.0/24 metric
  4294967295 - 20000000 4294967295
*Mar  6 02:19:11.387: IP-EIGRP(Default-IP-Routing-Table:1): 10.1.2.0/24 -
not in IP
  routing table
*Mar  6 02:19:11.387: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.2.0/24 metric
  4294967295 - 20000000 4294967295
P1R3#
```

P1R3 receives a query for network 10.1.0.0/24 from P1R1; 10.1.0.0/24 is unreachable, as indicated by the infinite metric 4294967295. P1R3 replies to the query, indicating that 10.1.0.0/24 is unreachable (using the same infinite metric).

- **Step 8** Turn off all debugging.

  Solution:

  The following shows the required command on the P1R3 router:

```
P1R3#no debug all
All possible debugging has been turned off
P1R3#
```

- **Step 9** Reenable the serial interface between the edge routers (the S0/0/1 interface on P*x*R1 and P*x*R2).

  Solution:

  The following shows the required configuration on the P1R1 router:

```
P1R1(config)#int s0/0/1
P1R1(config-if)#no shutdown
```

Task 2: Configuring EIGRP Summarization

In this task, you configure EIGRP route summarization. This will add stability and speed convergence of the network by controlling the scope of queries, minimizing update traffic, and minimizing routing table size. Follow these steps:

- **Step 1** Telnet to BBR1 (172.31.*x*.3) and verify that you see the specific subnet routes from your pod.

  Solution:

  The following shows sample output on the BBR1 router:

```
BBR1>show ip route eigrp
      10.0.0.0/24 is subnetted, 7 subnets
D       10.1.2.0 [90/20514560] via 172.31.1.2, 00:00:28, Serial0/0/0.1
D       10.1.1.0 [90/20514560] via 172.31.1.1, 00:00:29, Serial0/0/0.1
D       10.1.0.0 [90/21024000] via 172.31.1.2, 00:00:32, Serial0/0/0.1
                 [90/21024000] via 172.31.1.1, 00:00:32, Serial0/0/0.1
BBR1>
```

- **Step 2** Manually configure the edge routers (P*x*R1 and P*x*R2) to summarize the pod EIGRP routes to BBR1 into a single 10.*x*.0.0/16 advertisement (where *x* is your pod number).

  Solution:

  The following shows the required configuration on the P1R1 router:

```
P1R1(config)#int s0/0/0
P1R1(config-if)#ip summary-address eigrp 1 10.1.0.0 255.255.0.0
P1R1(config-if)#
```

  Both edge routers require the same summarization configuration.

- **Step 3** Telnet to BBR1 (172.31.*x*.3) and verify that you see only the summary route and not the more specific routes from your pod. If both edge routers are configured correctly, you should see two equal-cost paths available to BBR1.

  Solution:

  The following shows sample output on the BBR1 router:

```
BBR1>show ip route eigrp
  10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D    10.1.0.0/16 [90/20514560] via 172.31.1.2, 00:00:33, Serial0/0/0.1
                 [90/20514560] via 172.31.1.1, 00:00:33, Serial0/0/0.1
BBR1>
```

  Only the summarized 10.1.0.0/16 route is displayed; there are two equal-cost routes to this network, via P1R1 and P1R2.

Task 3: Configuring the EIGRP Stub

Having optimized BBR1's routing table by summarizing the routes from the pod's edge routers to the core BBR1 router, you now limit the query traffic from the pod's edge routers to its internal routers. Follow these steps:

- **Step 1** Configure the internal routers (P*x*R3 and P*x*R4) as EIGRP stubs. Remember that this bounds queries but does not affect the routing table.

  Solution:

  The following shows the required configuration on the P1R3 router:

```
P1R3(config)#router eigrp 1
P1R3(config-router)#eigrp stub
```

- **Step 2** Verify that the edge router recognizes its internal EIGRP neighbor as a stub.

  Solution:

  The following shows sample output on the P1R1 router. The highlighted lines indicate that P1R1 sees P1R3 (10.1.1.3) as a stub:

```
P1R1#show ip eigrp neighbors detail
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q
Seq
                                            (sec)         (ms)       Cnt
Num
1   10.1.1.3                Fa0/0             10 00:02:05   12   200  0
12
    Version 12.4/1.2, Retrans: 0, Retries: 0
    Stub Peer Advertising ( CONNECTED SUMMARY ) Routes
    Suppressing queries
0   10.1.0.2                Se0/0/1           12 00:06:46   25  1140  0
40
   Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 8
2   172.31.1.3              Se0/0/0          159 00:18:03  225  1350  0
4340
   Restart time 00:04:37
   Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 6
P1R1#
```

- **Step 3** The stub designation bounds query traffic and helps the router avoid getting into a stuck-in-active state, where EIGRP is unable to resolve routes for long periods. To demonstrate this situation, use the **debug ip eigrp** command on the internal router.
- **Step 4** Shut down the serial interface between the edge routers (the S0/0/1 interface between P*x*R1 and P*x*R2).
- **Step 5** Compared to the time before the internal routers were configured as stubs, notice that no queries are now being sent to the internal router. You should *not* see the "processing incoming QUERY" debug message on the internal routers, because they are configured as stub routers.

Solution:

The following shows the required command on the P1R3 router, the configuration on the P1R1 router, and example output on the P1R3 router. Queries are no longer being sent to the internal routers. P1R1 only sends the Update packet to P1R3:

```
P1R3#debug ip eigrp
IP-EIGRP Route Events debugging is on

P1R1(config)#int s0/0/1
P1R1(config-if)#shutdown
P1R3#
*Mar  6 02:32:34.507: IP-EIGRP(Default-IP-Routing-Table:1): Processing
incoming UPDATE packet
*Mar  6 02:32:34.507: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.0.0/24 M
  4294967295 - 0 4294967295 SM 4294967295 - 0 4294967295
*Mar  6 02:32:34.507: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.2.0/24 M
  4294967295 - 20000000 4294967295 SM 4294967295 - 20000000 4294967295
*Mar  6 02:32:34.523: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.0.0/24 metric
  4294967295 - 0 4294967295
*Mar  6 02:32:34.523: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.2.0/24 metric
  4294967295 - 20000000 4294967295
*Mar  6 02:32:34.543: IP-EIGRP(Default-IP-Routing-Table:1): Processing
incoming REPLY packet
*Mar  6 02:32:34.543: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.0.0/24 M
  4294967295 - 0 4294967295 SM 4294967295 - 0 4294967295
*Mar  6 02:32:34.543: IP-EIGRP(Default-IP-Routing-Table:1): Int
10.1.2.0/24 M
  4294967295 - 20000000 4294967295 SM 4294967295 - 20000000 4294967295
P1R3#
```

- **Step 6** Turn off debugging on the internal routers (P*x*R3 and P*x*R4).

  Solution:

  The following shows the required command on the P1R3 router:

  ```
  P1R3#no debug all
  All possible debugging has been turned off
  P1R3#
  ```

- **Step 7** Reenable the serial interface between the edge routers (the S0/0/1 interface between P*x*R1 and P*x*R2).

  Solution:

  The following shows the required configuration on the P1R1 router:

  ```
  P1R1(config)#int s0/0/1
  P1R1(config-if)#no shutdown
  ```

Task 4: Configuring an EIGRP Default Route

In this task, you advertise a default route from the edge routers to the internal routers via EIGRP. This change adds stability and speed convergence to the network by minimizing update traffic and routing table size. Follow these steps:

- **Step 1** Send a default route from the edge routers to the internal routers, and filter all specific routes. You can do this by configuring a summary route of 0.0.0.0 0.0.0.0 on each edge router, on the interface to the internal router.

  Solution:

  The following shows the required configuration on the P1R1 router:

```
P1R1(config)#int fa0/0
P1R1(config-if)#ip summary-address eigrp 1 0.0.0.0 0.0.0.0
```

- **Step 2** Examine the routing table on the internal routers. You should see the default routes and the connected routes, but the more specific routes from the edge router should have been filtered.

  Solution:

  The following shows sample output on the P1R3 router. Notice that the gateway of last resort is also now set on the internal routers:

```
P1R3#show ip route
<output omitted>
Gateway of last resort is 10.1.1.1 to network 0.0.0.0

      10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
D*   0.0.0.0/0 [90/30720] via 10.1.1.1, 00:01:58, FastEthernet0/0
```

- **Step 3** Ping the TFTP server (10.254.0.254) from the internal router to verify connectivity.

  Solution:

  The following shows sample output on the P1R3 router. The ping is successful:

```
P1R3#ping 10.254.0.254

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.254.0.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
P1R3#
```

- **Step 4** Examine the EIGRP topology table, EIGRP traffic statistics, information about interfaces configured for EIGRP, and EIGRP neighbors.

Solution:

The following shows sample output on the P1R1 router:

```
P1R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.31.1.1)

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

P 0.0.0.0/0, 1 successors, FD is 28160
     via Summary (28160/0), Null0
P 10.1.2.0/24, 1 successors, FD is 20514560
     via 10.1.0.2 (20514560/28160), Serial0/0/1
P 10.1.1.0/24, 1 successors, FD is 28160
     via Connected, FastEthernet0/0
P 10.1.0.0/16, 1 successors, FD is 28160
     via Summary (28160/0), Null0
P 10.1.0.0/24, 1 successors, FD is 20512000
     via Connected, Serial0/0/1
P 172.31.2.0/24, 1 successors, FD is 21024000
     via 172.31.1.3 (21024000/20512000), Serial0/0/0
P 172.31.1.0/24, 1 successors, FD is 20512000
     via Connected, Serial0/0/0
P 10.254.0.0/24, 1 successors, FD is 20514560
     via 172.31.1.3 (20514560/28160), Serial0/0/0

P1R1#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 1
  Hellos sent/received: 907/905
  Updates sent/received: 341/35
  Queries sent/received: 6/7
  Replies sent/received: 7/6
  Acks sent/received: 33/40
  Input queue high water mark 2, 0 drops
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 150
  PDM Process ID: 88

P1R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                    Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer
Routes
Fa0/0              1        0/0         4       0/10          50
0
Se0/0/1            1        0/0        35       5/190        346
0
Se0/0/0            2        0/0        75       5/190        748
0
P1R1#

P1R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q
Seq
                                            (sec)         (ms)       Cnt Num
```

```
0   10.1.0.2                Se0/0/1           14 00:07:39   35  1140  0
65
1   10.1.1.3                Fa0/0             13 00:14:21    4   200  0
18
2   172.31.1.3              Se0/0/0          139 00:30:19  151  1140  0
4341
P1R1#
```

- **Step 5** Save your configurations to NVRAM.

  Solution:

  The following shows how to perform the required step on the P1R1 router:

```
P1R1#copy run start
Destination filename [startup-config]?
Building configuration...[OK]
```

Exercise Verification

You have successfully completed this exercise when you achieve the following results:

- You have successfully implemented EIGRP and have observed EIGRP query traffic.
- You have summarized your pod addresses to the core.
- You have optimized performance on the internal routers.

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.