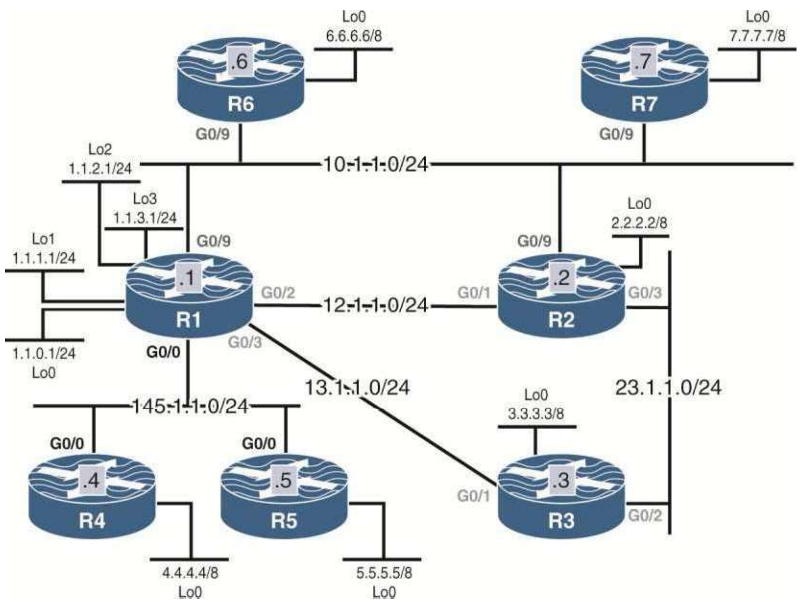
Lab 1
EIGRP Named Mode



Lab Setup:

To copy and paste the initial configurations, go to the Initial-config folder → EIGRP folder → Lab-1.

Task 1

Configure EIGRP on R1, R2, and R3 based on the following policy:

Router	Interface	AS Number
R1	G0/9 G0/0 G0/2 G0/3 Loopback0–Loopback3	200 100 100 100 100
R2	G0/9	200

	G0/1 G0/3 Loopback0	100 100 100
R3	G0/1 G0/2 Loopback0	100 100 100

- R1 should be configured to use unicast to establish an EIGRP neighbor adjacency with R2.
- R1 should use multicast to establish an EIGRP neighbor adjacency with R3.
- R1, R2, and R3 should use an EIGRP named mode configuration to accomplish this task.

EIGRP named mode is a new configuration mode for EIGRP that was introduced in IOS 15.0(1)M, 12.2(33)SRE, 12.2(33)XNE, and IOS XE release 2.5. It contains many enhancements to EIGRP, most notably unifying EIGRP configuration commands to a single location in the router configuration.

Prior to EIGRP named mode, some EIGRP configuration commands such as authentication and summarization, had to be configured under a specific interface. Certain commands such as redistribution and setting the router ID could be configured in global configuration mode. EIGRP named mode allows all of these configurations to exist in one location using global configuration mode. Interface-level commands are configured using the **af-interface** command within the EIGRP router configuration.

EIGRP named mode also unifies IPv4 and IPv6 configuration. Previously, EIGRP IPv6 configuration needed to be enabled separately using the **ipv6 router eigrp** command in global configuration mode. EIGRP named mode supports both IPv4 and IPv6 using address families. The **address-family ipv6 and address-family ipv4** commands configure the protocol to run for IPv6 and IPv4 respectively.

Basic EIGRP named mode configuration is similar to normal IPv4 configuration. The **router eigrp** command is issued to enable the configuration. The difference is, instead of specifying the AS number, an alphanumeric string (text containing letters or numbers) should be entered that designates the EIGRP virtual instance that is being defined. This virtual instance name does not have to match between EIGRP neighbors. Only the actual AS number configured for the routing protocol process must match between EIGRP neighbors. The actual AS number is configured using the **address-family [ipv4 | ipv6] unicast autonomous-system** command with the numeric AS number included.

Unlike RIP, automatic summarization is disabled by default in EIGRP. The show run does not display the default setting of auto summary even if **no auto-summary** has been configured. However, the setting will be displayed if auto summary is turned on.

The following configurations are made on R1, R2 and R3 to run EIGRP in named mode and advertise the networks specified in the task:

R1 is first configured to run EIGRP named mode for the virtual instance named tst with the **router eigrp tst** command. Then the command **address-family ipv4 unicast autonomous-system 100** is entered to create and enter the IPv4 address-family configuration mode for AS 100. EIGRP is then enabled on the interface with the **network** command as specified in the task above:

On R1:

```
R1(config)#router eigrp tst
R1(config-router)#address-family ipv4 unicast autonomous-system 100
R1(config-router-af)#network 12.1.1.1 0.0.0.0
R1(config-router-af)#network 13.1.1.1 0.0.0.0
R1(config-router-af)#network 145.1.1.1 0.0.0.0
R1(config-router-af)#network 1.1.0.1 0.0.0.0
R1(config-router-af)#network 1.1.1.1 0.0.0.0
R1(config-router-af)#network 1.1.2.1 0.0.0.0
R1(config-router-af)#network 1.1.3.1 0.0.0.0
```

Note : **address-family ipv4 as 100** the shorter version of the command **address-family ipv4 unicast autonomous-system 100** is also accepted by IOS.

By default when EIGRP is enabled on an interface, it multicast EIGRP packets to the 224.0.0.10 address to discover neighbors. The task however requires R1 to use unicast to form an EIGRP neighbor adjacency with R2. This hints towards using the **neighbor** command on R1 and R2 to statically form EIGRP adjacencies with each other. The **neighbor** command deactivates the transmission of the EIGRP multicast packets on the specified interface and will cause the router to drop any packets multicasted to the address 224.0.0.10. R1 is configured below to form a static EIGRP adjacency with R2 using the **neighbor 12.1.1.2 G0/2** command. When using the **neighbor** command, it is important to specify the interface over which the local router will attempt to form unicast adjacencies with the neighbor.

```
R1(config-router-af)#neighbor 12.1.1.2 g0/2
```

The task specifies that R1 and R2 should also participate in EIGRP AS 200 over their g0/9 interface. EIGRP named mode only supports a single ASN per address-family in a single virtual instance. Because of this, it is not possible to configure a second ASN in the same **tst** EIGRP virtual instance configured on R1 and R2. In order to complete this part of the task, a second EIGRP virtual instance called **tst200** is created for IPv4 AS 200 on R1. This is done with the **address-family ipv4 unicast autonomous-system 200** command under the **router eigrp tst200** configuration mode. EIGRP instance for AS 200 is enabled on the G0/9 interface with the **network 10.1.1.1 0.0.0.0** command:

```
R1(config)#router eigrp tst200
R1(config-router)#address-family ipv4 as 200
R1(config-router-af)#network 10.1.1.1 0.0.0.0
```

Next, EIGRP named mode is configured on R2. First, a virtual instance called **tst** is created to hold the configuration for EIGRP AS 100. Then, the interfaces g0/1, g0/3, and loopback 0 are all added to the EIGRP AS 100 process as indicated by the task. Finally, a static **neighbor** command is added to R2's configuration to allow it to form a neighbor adjacency with R1. Without this step, R2 would not be able to become neighbors with R1 because of R1's previous static neighbor configuration. The reason is because R1 is expecting unicast hello packets from R2's interface IP address while R2 is multicasting hellos onto the link. This mismatch causes R1 to ignore the hellos received from R2.

The final, complete configuration for the virtual instance **tst** on R2 is as follows:

On R2:

```
R2(config)#router eigrp tst
R2(config-router)#address-family ipv4 as 100
R2(config-router-af)#network 12.1.1.2 0.0.0.0
R2(config-router-af)#network 23.1.1.2 0.0.0.0
R2(config-router-af)#network 2.2.2.2 0.0.0.0
R2(config-router-af)#neighbor 12.1.1.1 g0/1
```

As soon as R2 is configured with the neighbor statement, a console message indicating that a neighbor adjacency has been established with neighbor 12.1.1.1 appears as seen below:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 12.1.1.1 (GigabitEthernet0/1)
is up: new adjacency
```

As with R1, R2 needs to run EIGRP ASN 200 over its G0/9 interface. This requires the creation of a second virtual instance named **tst200.** Within this virtual instance, R2 is configured to run EIGRP in named mode for AS 200 on its G0/9 interface using the **network** command as shown below:

```
R2(config)#router eigrp tst200
R2(config-router)#address-family ipv4 as 200
R2(config-router-af)#network 10.1.1.2 0.0.0.0
```

Finally, the same EIGRP named mode virtual instance **tst** is configured on R3 as well. Within this virtual instance, R3's g0/1, g0/2, and loopback 0 interfaces are enabled to run for EIGRP AS 100 using **network** commands. Log messages indicating successful EIGRP neighbor adjacencies to R1 and R2 will appear on R3 as shown below:

On R3:

```
R3(config)#router eigrp tst
R3(config-router)#address-family ipv4 as 100
R3(config-router-af)#network 23.1.1.3 0.0.0.0
R3(config-router-af)#network 13.1.1.3 0.0.0.0
R3(config-router-af)#network 3.3.3.3 0.0.0.0
```

You should see the following console messages:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 23.1.1.2 (GigabitEthernet0/2)
is up: new adjacency

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 13.1.1.1 (GigabitEthernet0/1)
is up: new adjacency
```

To verify the configuration:

After completing the above configurations, the routing tables on all three routers are examined with the **show ip route eigrp 100 | begin Gate** command to verify the EIGRP routes they learn from each other:

On R1:

```
R1#show ip route eigrp 100 | begin Gateway
Gateway of last resort is not set

D     2.0.0.0/8 [90/10880] via 12.1.1.2, 00:01:21, GigabitEthernet0/2
D     3.0.0.0/8 [90/10880] via 13.1.1.3, 00:01:20, GigabitEthernet0/3
      23.0.0.0/24 is subnetted, 1 subnets
 D        23.1.1.0 [90/15360] via 13.1.1.3, 00:01:21, GigabitEthernet0/3
                   [90/15360] via 12.1.1.2, 00:01:21, GigabitEthernet0/2
```

On R2:

```
R2#show ip route eigrp 100 | begin Gateway
Gateway of last resort is not set

      1.0.0.0/24 is subnetted, 4 subnets
D        1.1.0.0 [90/10880] via 12.1.1.1, 00:03:33, GigabitEthernet0/1
D        1.1.1.0 [90/10880] via 12.1.1.1, 00:03:33, GigabitEthernet0/1
D        1.1.2.0 [90/10880] via 12.1.1.1, 00:03:33, GigabitEthernet0/1
D        1.1.3.0 [90/10880] via 12.1.1.1, 00:03:33, GigabitEthernet0/1
D     3.0.0.0/8 [90/10880] via 23.1.1.3, 00:03:25, GigabitEthernet0/3
      13.0.0.0/24 is subnetted, 1 subnets
 D        13.1.1.0 [90/15360] via 23.1.1.3, 00:03:35, GigabitEthernet0/3
                   [90/15360] via 12.1.1.1, 00:03:35, GigabitEthernet0/1
```

```
      145.1.0.0/24 is subnetted, 1 subnets
D        145.1.1.0 [90/15360] via 12.1.1.1, 00:03:33, GigabitEthernet0/1
```

On R3:

```
R3#show ip route eigrp 100 | begin Gateway
Gateway of last resort is not set

      1.0.0.0/24 is subnetted, 4 subnets
D        1.1.0.0 [90/10880] via 13.1.1.1, 00:00:43, GigabitEthernet0/1
D        1.1.1.0 [90/10880] via 13.1.1.1, 00:00:43, GigabitEthernet0/1
D        1.1.2.0 [90/10880] via 13.1.1.1, 00:00:43, GigabitEthernet0/1
D        1.1.3.0 [90/10880] via 13.1.1.1, 00:00:43, GigabitEthernet0/1
D     2.0.0.0/8 [90/10880] via 23.1.1.2, 00:00:43, GigabitEthernet0/2
      12.0.0.0/24 is subnetted, 1 subnets
D        12.1.1.0 [90/15360] via 23.1.1.2, 00:00:43, GigabitEthernet0/2
                  [90/15360] via 13.1.1.1, 00:00:43, GigabitEthernet0/1
      145.1.0.0/24 is subnetted, 1 subnets
D        145.1.1.0 [90/15360] via 13.1.1.1, 00:00:43, GigabitEthernet0/1
```

Task 2

Configure R4 and R5 in EIGRP AS 100. You must use named mode to accomplish this task.

As in the earlier task, R4 and R5 are configured to form EIGRP neighbor adjacencies using EIGRP named mode as seen below. The **network** command under the address-family IPv4 EIGRP configuration mode enables and advertises the networks on the interfaces specified in the task:

On R4:

```
R4(config)#router eigrp tst
R4(config-router)#address-family ipv4 as 100
R4(config-router-af)#network 145.1.1.4 0.0.0.0
R4(config-router-af)#network 4.4.4.4 0.0.0.0
```

You should see the following console message:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 145.1.1.1 (GigabitEthernet0/0)
is up: new adjacency
```

On R5:

```
R5(config)#router eigrp tst
R5(config-router)#address-family ipv4 as 100
R5(config-router-af)#network 5.5.5.5 0.0.0.0
R5(config-router-af)#network 145.1.1.5 0.0.0.0
```

You should see the following console messages:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 145.1.1.4 (GigabitEthernet0/0)
is up: new adjacency

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 145.1.1.1 (GigabitEthernet0/0)
is up: new adjacency
```

On configuring above, R4 and R5 form EIGRP adjacencies with each other and R1. This is confirmed with the **show ip eigrp neighbors** command. The **show ip route eigrp | begin Gateway** command is then used to verify the EIGRP routes learned by R4 and R5 as seen below:

To verify the configuration:

On R4:

```
R4#show ip eigrp neighbors
EIGRP-IPv4 VR(tst) Address-Family Neighbors for AS(100)

H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   145.1.1.5               Gi0/0                    12 00:04:00    1   100  0  6
0   145.1.1.1               Gi0/0                    11 00:04:46 1021  5000  0  16

R4#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/24 is subnetted, 4 subnets
D        1.1.0.0 [90/10880] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
D        1.1.1.0 [90/10880] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
D        1.1.2.0 [90/10880] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
D        1.1.3.0 [90/10880] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
D     2.0.0.0/8 [90/16000] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
D     3.0.0.0/8 [90/16000] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
D     5.0.0.0/8 [90/10880] via 145.1.1.5, 00:03:31, GigabitEthernet0/0
      12.0.0.0/24 is subnetted, 1 subnets
D        12.1.1.0 [90/15360] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
      13.0.0.0/24 is subnetted, 1 subnets
D        13.1.1.0 [90/15360] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
```

```
      23.0.0.0/24 is subnetted, 1 subnets
D        23.1.1.0 [90/20480] via 145.1.1.1, 00:04:17, GigabitEthernet0/0
```

On R5:

```
R5#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/24 is subnetted, 4 subnets
D        1.1.0.0 [90/10880] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
D        1.1.1.0 [90/10880] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
D        1.1.2.0 [90/10880] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
D        1.1.3.0 [90/10880] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
D     2.0.0.0/8 [90/16000] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
D     3.0.0.0/8 [90/16000] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
D     4.0.0.0/8 [90/10880] via 145.1.1.4, 00:00:42, GigabitEthernet0/0
      12.0.0.0/24 is subnetted, 1 subnets
D        12.1.1.0 [90/15360] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
      13.0.0.0/24 is subnetted, 1 subnets
D        13.1.1.0 [90/15360] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
      23.0.0.0/24 is subnetted, 1 subnets
D        23.1.1.0 [90/20480] via 145.1.1.1, 00:00:42, GigabitEthernet0/0
```

Task 3

Configure R1, R4, and R5 to use unicast to establish their EIGRP neighbor adjacency.

The following first configures R1 to statically form EIGRP neighbor adjacencies with R4 and R5 using the **neighbor** command with G0/0 specified as the interface:

On R1:

```
R1(config)#router eigrp tst
R1(config-router)#address-family ipv4 as 100
R1(config-router-af)#neighbor 145.1.1.4 g0/0
R1(config-router-af)#neighbor 145.1.1.5 g0/0

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 145.1.1.5 (GigabitEthernet0/0)
is down: Static peer configured
```

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 145.1.1.4 (GigabitEthernet0/0)
is down: Static peer configured
```

On issuing the static neighborship configuration on R1, the CLI displays certain log messages as seen above indicating EIGRP neighborships to R4 and R5 have gone down. Following this, a **show ip eigrp neighbors** command on R1 reveals it hasn't restored the EIGRP neighborships with R4 and R5:

```
R1#show ip eigrp neighbors
EIGRP-IPv4 VR(tst) Address-Family Neighbors for AS(100)
H   Address                 Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                                   (sec)         (ms)       Cnt Num
1   13.1.1.3                Gi0/3                    11 1d23h       1   100  0  24
0   12.1.1.2                Gi0/2                    13 1d23h       1   100  0  23

! No EIGRP adjacencies with R4 and R5
```

Why hasn't R1 re-established its EIGRP adjacency to these routers? The answer lies in what happens when you enable static configuration for EIGRP. As mentioned in Task 1, after configuring static neighbors, a router expects to receive only unicast hellos from the neighbor on that interface. The use of EIGRP multicast is entirely deactivated on that interface. This means, if a router receives a multicast hellos on that interface, it will ignore it. This can be seen in action by turning on EIGRP debugging with the **debug eigrp packets** command on R1:

```
R1#debug eigrp packets

    (UPDATE, REQUEST, QUERY, REPLY, HELLO, UNKNOWN, PROBE, ACK, STUB,
SIAQUERY, SIAREPLY)
EIGRP Packet debugging is on

*Aug  3 12:42:53.753: EIGRP: Ignore multicast Hello Gi0/0 145.1.1.4
*Aug  3 12:42:54.016: EIGRP: Ignore multicast Hello Gi0/0 145.1.1.5
```

R4 and R5 continue to multicast EIGRP hellos in order to discover potential neighbors. R1 configured for static EIGRP neighborships ignores the multicast hellos received from R4 and R5 as seen above. In fact, R4 and R5 are expecting to receive multicast hellos and as such ignore the unicast hellos transmitted by R1:

On R4 and R5:

```
*Aug 3 12:49:58.589: EIGRP: Ignore unicast Hello from GigabitEthernet0/0
145.1.1.1
```

This means, on a multiaccess segment, EIGRP cannot unicast to one or some neighbors and multicast to others. If static neighborships are intended, all devices on that segment need to be configured to manually form EIGRP neighbor adjacencies with each other. To complete this task, static neighbor configurations are made on R4 and R5 as well:

On R4:

```
R4(config)#router eigrp tst
R4(config-router)#address-family ipv4 as 100
R4(config-router-af)#neighbor 145.1.1.1 g0/0
R4(config-router-af)#neighbor 145.1.1.5 g0/0
```

You should see the following console messages:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 145.1.1.1 (GigabitEthernet0/0)
is up: new adjacency
```

On R5:

```
R5(config)#router eigrp tst
R5(config-router)#address-family ipv4 as 100
R5(config-router-af)#neighbor 145.1.1.1 g0/0
R5(config-router-af)#neighbor 145.1.1.4 g0/0
```

You should see the following console messages:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 145.1.1.1 (GigabitEthernet0/0)
is up: new adjacency

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 145.1.1.4 (GigabitEthernet0/0)
is up: new adjacency
```

On completing the above configuration, log messages confirm the adjacencies. The **show ip eigrp neighbors** command also verifies the same and the **show ip route eigrp** command output verifies the EIGRP routes learned:

To verify the configuration:

On R4:

```
R4#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/24 is subnetted, 4 subnets
D        1.1.0.0 [90/10880] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
D        1.1.1.0 [90/10880] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
D        1.1.2.0 [90/10880] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
D        1.1.3.0 [90/10880] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
D     2.0.0.0/8 [90/16000] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
```

```
D     3.0.0.0/8 [90/16000] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
      12.0.0.0/24 is subnetted, 1 subnets
D        12.1.1.0 [90/15360] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
      13.0.0.0/24 is subnetted, 1 subnets
D        13.1.1.0 [90/15360] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
      23.0.0.0/24 is subnetted, 1 subnets
D        23.1.1.0 [90/20480] via 145.1.1.1, 00:00:41, GigabitEthernet0/0
```

Task 4

Configure R6 in EIGRP AS 200. This router should run EIGRP AS 200 on its G0/9 and Loopback0 interfaces. You should use an EIGRP named mode configuration to accomplish this task.

The following configures EIGRP named mode for AS 200 on R6 and the **network** command enables EIGRP on the interfaces specified:

On R6:

```
R6(config)#router eigrp tst200
R6(config-router)#address-family ipv4 as 200
R6(config-router-af)#netw 6.6.6.6 0.0.0.0
R6(config-router-af)#netw 10.1.1.6 0.0.0.0
```

You should see the following console message:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 200: neighbor 10.1.1.2 (GigabitEthernet0/9)
is up: new adjacency

%DUAL-5-NBRCHANGE: EIGRP-IPv4 200: neighbor 10.1.1.1 (GigabitEthernet0/9)
is up: new adjacency
```

The above results in EIGRP adjacencies between R1 and R6. R1's routing table for EIGRP AS 200 verifies the EIGRP routes it learns from R6:

To verify the configuration:

On R1:

```
R1#show ip route eigrp 200 | begin Gateway
```

```
Gateway of last resort is not set

D      6.0.0.0/8 [90/10880] via 10.1.1.6, 00:00:39, GigabitEthernet9
```

Task 5

Configure OSPF Area 0 on R6's G0/9 and R7's G0/9 and Loopback0 interfaces. The router ID of these routers should be configured as 0.0.0.x, where x is the router number.

The following enables OSPF 1 for Area 0 on R6 and R7's G0/9 interfaces using the **network** command. OSPF is also enabled on the loopback interface on R7. Additionally, as required by the task, the OSPF router IDs are manually set using the **router-id** command:

On R6:

```
R6(config)#router ospf 1
R6(config-router)#router-id 0.0.0.6
R6(config-router)#network 10.1.1.6 0.0.0.0 area 0
```

On R7:

```
R7(config)#router ospf 1
R7(config-router)#router-id 0.0.0.7
R7(config-router)#network 10.1.1.7 0.0.0.0 area 0
R7(config-router)#network 7.7.7.7 0.0.0.0 area 0
```

The following console log message appears on the router verifying the OSPF adjacencies:

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.6 on GigabitEthernet0/9 from LOADING
to FULL, Loading Done
```

To verify the configuration:

On R6:

```
R6#show ip route ospf | begin Gateway
Gateway of last resort is not set

      7.0.0.0/32 is subnetted, 1 subnets
O        7.7.7.7 [110/2] via 10.1.1.7, 00:00:27, GigabitEthernet0/9
```

Task 6

Configure R6 to redistribute OSPF into EIGRP such that R1 and R2 go directly to R7 to reach the 7.0.0.0/8 network.

This task first requires configuring R6 to redistribute OSPF routes into EIGRP 200. Redistribution for EIGRP named mode is done in topology base configuration mode. Topology base configuration mode is entered using the **topology base** command within EIGRP router address-family configuration mode within an EIGRP virtual instance. There are many EIGRP functions that are executed in topology base configuration mode. The output below lists the variety of command options available:

On R6:

```
R6(config)#router eigrp tst200
R6(config-router)#address-family ipv4 as 200
R6(config-router-af)#topology base
R6(config-router-af-topology)#?
Address Family Topology configuration commands:
  auto-summary          Enable automatic netw number summarization
  default               Set a command to its defaults
  default-information   Control distribution of default information
  default-metric        Set metric of redistributed routes
  distance              Define an administrative distance
  distribute-list       Filter entries in eigrp updates
  eigrp                 EIGRP specific commands
  exit-af-topology      Exit from Address Family Topology configuration mode
  maximum-paths         Forward packets over multiple paths
  metric                Modify metrics and parameters for advertisement
  no                    Negate a command or set its defaults
  offset-list           Add or subtract offset from EIGRP metrics
  redistribute          Redistribute IPv4 routes from another routing protocol
  snmp                  Modify snmp parameters
  summary-metric        Specify summary to apply metric/filtering
  timers                Adjust topology specific timers
  traffic-share         How to compute traffic share over alternate paths
  variance              Control load balancing variance
```

The **redistribute** command in topology base configuration mode activates redistribution of other protocols into EIGRP. When routes from other routing sources are redistributed into EIGRP, EIGRP, like all other routing protocols, needs to be able to assign a metric value to the redistributed prefixes. This is no problem if the source routing protocol is another EIGRP virtual instance running on the router because the metric values used by the source protocol will be the same as EIGRP's own metrics. When redistributing routes from a different routing protocol, however, EIGRP cannot accept the metric values. Some routing protocols have a default metric that is assigned to all redistributed prefixes, but EIGRP does not have such a function. Instead, the administrator must manually provide the component metric information as a seed metric to be used for all redistributed prefixes. Without this seed metric, EIGRP will not redistribute any routes from the intended source protocol. The seed metric can be set using the **metric** option along with the **redistribute** command as in the below:

```
R6(config)#router eigrp tst200
R6(config-router)#address-family ipv4 as 200
R6(config-router-af)#topology base
R6(config-router-af-topology)#redistribute ospf 1 metric 10000 1000 255 1
1500
```

The above configuration causes R6 to advertise the redistribute routes as EIGRP external routes to it's EIGRP neighbor R1. This can be verified on R1 with the **show ip route 7.7.7.7** command as seen below:

To verify the configuration:

On R1:

```
R1#show  ip route 7.7.7.7

Routing entry for 7.7.7.7/32
  Known via "eigrp 200", distance 170, metric 5637120, type external
  Redistributing via eigrp 200
  Last update from 10.1.1.6 on GigabitEthernet9, 00:03:28 ago
  Routing Descriptor Blocks:
  * 10.1.1.6, from 10.1.1.6, 00:03:28 ago, via GigabitEthernet0/9
      Route metric is 5637120, traffic share count is 1
      Total delay is 10010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

Notice in the output above the next hop IP Address for the 7.7.7.7 prefix on R1 is R6's interface 10.1.1.6. The same occurs on R2. Because R6 is the next-hop for the prefix, whenever R1 or R2 try to route packets to the 7.7.7.7 prefix, they will forward the packers to R6 which will then forward to R7. The task requirements state that R1 and R2 should route directly to R7 for the 7.0.0.0/8 network. This would be a simple task if R1 and R2 were also running OSPF, but, in this case, R1 and R2 are not configured for OSPF and R7 isn't running EIGRP. The only way R1 and R2 receive the 7.0.0.0/8 prefixes is via redistribution into EIGRP from R6.

In order to meet the task requirement, changes must be made such that the R6 advertises R7, the original next-hop from OSPF, as the next-hop in its EIGRP advertisement to R1 and R2. How can this be done? By enabling the third-party next-hop feature.

The third-party next-hop feature prevents an advertising router from setting itself as the next hop for EIGRP routes it advertises. The feature assumes that if all routers are in the same subnet, then they have reachability to each other. Advertising the original next-hop prevents suboptimal routing situations like what is currently present in the topology between R1, R2, R6, and R7.

In other words, R6 will retain the original next-hop address of R7 when advertising the 7.7.7.7 prefix to R1 and R2 via EIGRP. Since R1 and R2 belong to the same subnet as R7, the 10.1.1.0/24, they can directly route packets to R7 instead of sending it to R6.

Certain protocols such as BGP have this feature turned on by default, EIGRP requires an extra configuration step to enable this feature. By default as seen above, R6 resets the next hop to itself when advertising the 7.7.7.7 prefix. It can be prevented from doing this by using the **no next-hop-self** command in af-interface configuration mode for its G0/9. This mode is entered from router address-family configuration mode using the af-interface command:

On R6:

```
R6(config)#router eigrp tst200
R6(config-router)#address-family ipv4 as 200
R6(config-router-af)#af-interface g0/9
R6(config-router-af-interface)#?
Address Family Interfaces configuration commands:
  add-paths             Advertise add paths
  authentication        authentication subcommands
  bandwidth-percent     Set percentage of bandwidth percentage limit
  bfd                   Enable Bidirectional Forwarding Detection
  dampening-change      Percent interface metric must change to cause update
  dampening-interval    Time in seconds to check interface metrics
  default               Set a command to its defaults
  exit-af-interface     Exit from Address Family Interface configuration mode
  hello-interval        Configures hello interval
  hold-time             Configures hold time
  next-hop-self         Configures EIGRP next-hop-self
  no                    Negate a command or set its defaults
  passive-interface     Suppress address updates on an interface
  shutdown              Disable Address-Family on interface
  split-horizon         Perform split horizon
  summary-address       Perform address summarization

R6(config-router-af-interface)#no next-hop-self
```

Once next-hop-self is disabled, the true next hop is now being retained by R6 when advertising to R1 and R2. The **show ip route 7.7.7.7** output on R1 and R2's correctly list R7 as the next hop for the 7.7.7.7 prefix as seen below:

To verify the configuration:

On R1:

```
R1#show ip route 7.7.7.7

Routing entry for 7.7.7.7/32
  Known via "eigrp 200", distance 170, metric 5637120, type external
  Redistributing via eigrp 200
  Last update from 10.1.1.7 on GigabitEthernet9, 00:02:25 ago
  Routing Descriptor Blocks:
  * 10.1.1.7, from 10.1.1.6, 00:02:25 ago, via GigabitEthernet0/9
      Route metric is 5637120, traffic share count is 1
      Total delay is 10010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

Task 7

Configure the hello interval of all routers in AS 200 to be twice the default hello interval.

The hello interval or the frequency at which EIGRP hello packets are transmitted defaults to 5 seconds. This is confirmed with the **show ip eigrp 200 interface detail | include interval** output below from R1:

```
R1#show ip eigrp 200 interface detail | include interval

  Hello-interval is 5, Hold-time is 15
```

The task requires doubling the hello interval on all routers configured for EIGRP AS 200, meaning, R1, R2 and R6. The hello interval can be modified with the **hello-interval** command under the EIGRP af-interface configuration mode. The value is set to 10 seconds (two times the default interval of 5) on the routers as seen below:

On R1 and R2:

```
Rx(config)#router eigrp tst200
Rx(config-router)#address-family ipv4 as 200
Rx(config-router-af)#af-interface g0/9
Rx(config-router-af-interface)#hello-interval 10
```

On R6:

```
R6(config)#router eigrp tst200
```

```
R6(config-router)#address-family ipv4 as 200
R6(config-router-af)#af-interface g0/9
R6(config-router-af-interface)#hello-interval 10
```

The **show ip eigrp 200 interface detail | include interval** command output on these routers verify the change:

To verify the configuration:

On R1:

```
R1#show ip eigrp 200 interface detail | include interval

  Hello-interval is 10, Hold-time is 15
```

On R6:

```
R1#show ip eigrp 200 interface detail | include interval

  Hello-interval is 5, Hold-time is 15
  Hello-interval is 10, Hold-time is 15
```

Note: When the hello interval is changed the hold-time is NOT changed automatically.

Task 8

Configure R4 such that in the worst-case scenario, it uses 10% of the bandwidth for its EIGRP updates. This policy should apply to the existing and future interfaces.

By default, EIGRP will ONLY consume 50 percent of its interface's bandwidth for EIGRP control packets. This default can be changed using the **bandwidth-percent** command in interface configuration mode or in the **af-interface** configuration mode under the address-family. The following modifies the utilization to 10 percent on R4 under the **af-interface default** configuration mode. The **af-interface default** configuration mode is a special interface configuration mode. Any configuration made in this mode is automatically applied to all interfaces on the router as a default setting:

On R4:

```
R4(config)#router eigrp tst
R4(config-router)#address-family ipv4 as 100
R4(config-router-af)#af-interface default
```

```
R4(config-router-af-interface)#bandwidth-percent 10
```

The **show ip eigrp interface detail** output verifies the change:

To verify the configuration:

```
R4#show ip eigrp interface detail | include BW

  Interface BW percentage is 10
  Interface BW percentage is 10
```

Task 9

Configure R1 to summarize its loopback interfaces and advertise a single summary in the EIGRP AS 100 routing domain.

Following loopback interfaces configured on R1 are to be summarized into a 1.1.0.0/22 summary address:

On R1:

```
R1#show ip interface brief | include Loopback

Loopback0              1.1.0.1        YES manual up                    up
Loopback1              1.1.1.1        YES manual up                    up
Loopback2              1.1.2.1        YES manual up                    up
Loopback3              1.1.3.1        YES manual up                    up
```

The **af-interface default** configuration mode in EIGRP named mode however does not include a summarization option as seen below:

```
R1(config)#router eigrp tst
R1(config-router)#address-family ipv4 autonomous-system 100
R1(config-router-af)#af-interface default
R1(config-router-af-interface)#?
Address Family Interfaces configuration commands:
  add-paths            Advertise add paths
  authentication       authentication subcommands
  bandwidth-percent    Set percentage of bandwidth percentage limit
  bfd                  Enable Bidirectional Forwarding Detection
  dampening-change     Percent interface metric must change to cause update
  dampening-interval   Time in seconds to check interface metrics
  default              Set a command to its defaults
```

```
  exit-af-interface   Exit from Address Family Interface configuration mode
  hello-interval      Configures hello interval
  hold-time           Configures hold time
  next-hop-self       Configures EIGRP next-hop-self
  no                  Negate a command or set its defaults
  passive-interface   Suppress address updates on an interface
  shutdown            Disable Address-Family on interface
  split-horizon       Perform split horizon

! No summarization option
```

This means any summarization in EIGRP named mode needs to be done individually for each interface under the **af-interface** configuration mode with the **summary-address** command. This is seen below on R1 where the **summary-address 1.1.0.0 255.255.252.0** command is configured under each interface on R1 that is enabled for EIGRP:

```
R1(config)#router eigrp tst
R1(config-router)#address-family ipv4 as 100
R1(config-router-af)#af-interface g0/2
R1(config-router-af-interface)#summary-address 1.1.0.0 255.255.252.0
R1(config-router-af)#af-interface g0/3
R1(config-router-af-interface)#summary-address 1.1.0.0 255.255.252.0
R1(config-router-af)#af-interface g0/0
R1(config-router-af-interface)#summary-address 1.1.0.0 255.255.252.0
```

The result of the above configuration can be verified on the neighboring routers as seen below. R2, R3, R4 and R5 each receive a summary route from R1:

To verify the configuration:

On R2:

```
R2#sh ip route eigrp | b Gate
Gateway of last resort is not set

      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/10880] via 12.1.1.1, 00:01:36, GigabitEthernet0/1
D     3.0.0.0/8 [90/10880] via 23.1.1.3, 06:57:00, GigabitEthernet0/3
D     4.0.0.0/8 [90/16000] via 12.1.1.1, 06:41:09, GigabitEthernet0/1
D     6.0.0.0/8 [90/10880] via 10.1.1.6, 05:27:20, GigabitEthernet0/9
      7.0.0.0/32 is subnetted, 1 subnets
D EX     7.7.7.7 [170/5637120] via 10.1.1.7, 05:27:20, GigabitEthernet0/9
      13.0.0.0/24 is subnetted, 1 subnets
 D        13.1.1.0 [90/15360] via 23.1.1.3, 06:57:03, GigabitEthernet0/3
                   [90/15360] via 12.1.1.1, 06:57:03, GigabitEthernet0/1
```

```
      145.1.0.0/24 is subnetted, 1 subnets
D        145.1.1.0 [90/15360] via 12.1.1.1, 06:56:58, GigabitEthernet0/1
```

On R3:

```
R3#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/10880] via 13.1.1.1, 00:02:15, GigabitEthernet0/1
D     2.0.0.0/8 [90/10880] via 23.1.1.2, 06:57:37, GigabitEthernet0/2
D     4.0.0.0/8 [90/16000] via 13.1.1.1, 06:41:48, GigabitEthernet0/1
      12.0.0.0/24 is subnetted, 1 subnets
D        12.1.1.0 [90/15360] via 23.1.1.2, 06:57:37, GigabitEthernet0/2
                  [90/15360] via 13.1.1.1, 06:57:37, GigabitEthernet0/1
      145.1.0.0/24 is subnetted, 1 subnets
D        145.1.1.0 [90/15360] via 13.1.1.1, 06:57:37, GigabitEthernet0/1
```

On R4:

```
R4#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/10880] via 145.1.1.1, 00:03:11, GigabitEthernet0/0
D     2.0.0.0/8 [90/16000] via 145.1.1.1, 06:42:55, GigabitEthernet0/0
D     3.0.0.0/8 [90/16000] via 145.1.1.1, 06:42:55, GigabitEthernet0/0
      12.0.0.0/24 is subnetted, 1 subnets
D        12.1.1.0 [90/15360] via 145.1.1.1, 06:42:55, GigabitEthernet0/0
      13.0.0.0/24 is subnetted, 1 subnets
D        13.1.1.0 [90/15360] via 145.1.1.1, 06:42:55, GigabitEthernet0/0
      23.0.0.0/24 is subnetted, 1 subnets
D        23.1.1.0 [90/20480] via 145.1.1.1, 06:42:55, GigabitEthernet0/0
```

On R5:

```
R5#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/10880] via 145.1.1.1, 00:03:02, GigabitEthernet0/0
D     2.0.0.0/8 [90/16000] via 145.1.1.1, 00:03:02, GigabitEthernet0/0
D     3.0.0.0/8 [90/16000] via 145.1.1.1, 00:03:02, GigabitEthernet0/0
D     4.0.0.0/8 [90/10880] via 145.1.1.4, 00:02:51, GigabitEthernet0/0
      12.0.0.0/24 is subnetted, 1 subnets
```

```
D        12.1.1.0 [90/15360] via 145.1.1.1, 00:03:02, GigabitEthernet0/0
      13.0.0.0/24 is subnetted, 1 subnets
D        13.1.1.0 [90/15360] via 145.1.1.1, 00:03:02, GigabitEthernet0/0
      23.0.0.0/24 is subnetted, 1 subnets
D        23.1.1.0 [90/20480] via 145.1.1.1, 00:03:02, GigabitEthernet0/0
```

Task 10

Configure R1 to limit the number of received prefixes from R5 to 10. R1 should be configured to receive a warning message once 50% of this threshold is reached and a warning message for every additional route that exceeds the threshold. You should configure Lo1–Lo10 on R5 by copying and pasting the initial configuration, called **EIGRP-Lab-1-Task10**.

EIGRP's **maximum-prefix** configuration option allows for controlling the number of EIGRP prefixes a router learns from a specific peer under an address-family. This function is enabled on a per-neighbor basis and is configured in address-family configuration mode the **neighbor** *neighbor-address* **maximum-prefix** *maximum-value* command.

This task requires R1 to be configured such that it only accepts 10 prefixes from its neighbor R5. In addition, IOS should also display a warning message once it receives 50 percent of the maximum allowed prefixes from that neighbor. Meaning, when R1 has received 5 prefixes from R5, it should display a warning. This can be achieved by appending additional arguments to the **neighbor maximum-prefix** command by setting the value to 50 before the **warning-only** keyword as shown below:

On R1:

```
R1(config)#router eigrp tst
R1(config-router)#address-family ipv4 as 100
R1(config-router-af)#neighbor 145.1.1.5 maximum-prefix 10 50 warning-only
```

To test the configuration, first EIGRP is enabled on the Loopback 1, Loopback 2 and Loopback 3 addresses on R5:

On R5:

```
R5(config)#router eigrp tst
R5(config-router)#address-family ipv4 as 100
R5(config-router-af)#network 51.1.1.5 0.0.0.0
R5(config-router-af)#network 52.1.1.5 0.0.0.0
R5(config-router-af)#network 53.1.1.5 0.0.0.0
```

In total, since R1 has received only 4 EIGRP prefixes from R5, it will not display any warning messages. Next, EIGRP is enabled on another loopback on R5:

```
R5(config-router-af)#network 54.1.1.5 0.0.0.0
```

The above results in R1 displaying a warning message that the threshold prefix level of 5 has been reached:

On R1:

We can see that the first message was received:

```
%DUAL-4-PFXLIMITTHR: EIGRP-IPv4 100: neighbor threshold prefix level(5)
reached.
```

EIGRP is then enabled on six more loopbacks on R5:

On R5:

```
R5(config-router-af)#network 55.1.1.5 0.0.0.0
R5(config-router-af)#network 56.1.1.5 0.0.0.0
R5(config-router-af)#network 57.1.1.5 0.0.0.0
R5(config-router-af)#network 58.1.1.5 0.0.0.0
R5(config-router-af)#network 59.1.1.5 0.0.0.0
```

Since R5 is now advertising 11 prefixes in total to R1, the following log message stating that the prefix limit has been reached is displayed on R1:

On R1:

```
%DUAL-3-PFXLIMIT: EIGRP-IPv4 100: neighbor prefix limit reached(10).
```

Task 11

Configure R1 to limit the number of prefixes received from R4 to five. R1 should be configured to tear down the adjacency if R4 exceeds the specified threshold. Copy and paste the **EIGRP-Lab-1-Task11** initial configuration on R4.

This task takes a similar direction as Task 10. It requires configuring R1 such that it only accepts a maximum of five EIGRP prefixes from R4. In this case, if R1 receives more than five prefixes from R4, it should tear down its EIGRP adjacency with R4.

To complete the task, we configure the **neighbor 145.1.1.4 maximum-prefix 5** command under the address-family configuration mode on R1 with no other options specified.

On R1:

```
R1(config)#router eigrp tst
R1(config-router)#address-family ipv4 as 100
R1(config-router-af)#neighbor 145.1.1.4 maximum-prefix 5
```

With no other options specified, by default, the threshold is set to 80 percent. Meaning, if R1 receives 80 percent of the configured maximum prefix value, it will display a warning message. If it reaches the maximum prefix value, the neighborship to the neighbor is shut down.

To verify the configuration, R4's loopback 1 interface is advertised in EIGRP. No warning message is displayed since R1 is receiving only two EIGRP prefixes (4.0.0.0/8 and 41.0.0.0/8) from R4:

On R4:

```
R4(config)#router eigrp tst
R4(config-router)#address-family ipv4 as 100
R4(config-router-af)#network 41.1.1.4 0.0.0.0
```

To test the warning message, two more networks are advertised into EIGRP on R4 (Lo2, Lo3 and Lo4):

```
R4(config-router-af)#network 42.1.1.4 0.0.0.0
R4(config-router-af)#network 43.1.1.4 0.0.0.0
```

R1 now receiving 80 percent of the maximum allowed prefixes, displays a warning message indicating the threshold prefix level:

```
%DUAL-4-PFXLIMITTHR: EIGRP-IPv4 100: neighbor threshold prefix level(3)
reached.
```

Finally, R4 is configured to advertise two more loopback addresses into EIGRP. The result is R1 displaying the message **prefix limit reached(5)** and tearing down its EIGRP adjacency with R4 as shown below. The **show ip eigrp neighbor** command verifies the same:

```
R4(config-router-af)#network 44.1.1.4 0.0.0.0
```

On R1:

```
%DUAL-3-PFXLIMIT: EIGRP-IPv4 100: neighbor prefix limit reached(5).

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 145.1.1.4
(GigabitEthernet0/0) is down: prefix-limit exceeded
```

We can see that the adjacency is torn down, let's verify the neighbor table of R4:

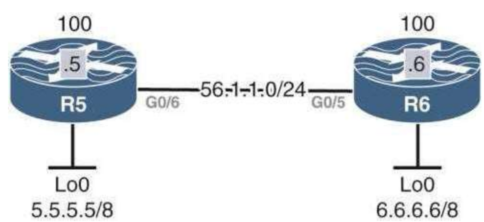
```
R4#show ip eigrp neighbor

EIGRP-IPv4 VR(tst) Address-Family Neighbors for AS(100)
H   Address         Interface              Hold Uptime    SRTT   RTO  Q  Seq
                                           (sec)          (ms)       Cnt Num
1   145.1.1.5       Gi0/0                   11 00:19:06     7    100  0  29

R4#show ip route eigrp | begin Gateway
Gateway of last resort is not set

D      5.0.0.0/8 [90/10880] via 145.1.1.5, 00:20:25, GigabitEthernet0/0
D      51.0.0.0/8 [90/10880] via 145.1.1.5, 00:11:07, GigabitEthernet0/0
D      52.0.0.0/8 [90/10880] via 145.1.1.5, 00:11:02, GigabitEthernet0/0
D      53.0.0.0/8 [90/10880] via 145.1.1.5, 00:10:58, GigabitEthernet0/0
D      54.0.0.0/8 [90/10880] via 145.1.1.5, 00:10:33, GigabitEthernet0/0
D      55.0.0.0/8 [90/10880] via 145.1.1.5, 00:09:08, GigabitEthernet0/0
D      56.0.0.0/8 [90/10880] via 145.1.1.5, 00:08:49, GigabitEthernet0/0
D      57.0.0.0/8 [90/10880] via 145.1.1.5, 00:08:28, GigabitEthernet0/0
D      58.0.0.0/8 [90/10880] via 145.1.1.5, 00:08:16, GigabitEthernet0/0
D      59.0.0.0/8 [90/10880] via 145.1.1.5, 00:08:10, GigabitEthernet0/0
D      60.0.0.0/8 [90/10880] via 145.1.1.5, 00:07:42, GigabitEthernet0/0
```

We can see that R4 only exchanges routes with R5 and NOT R1.

Task 12

Erase the startup configuration and reload the routers before proceeding to the next lab.

Lab 2
EIGRP and Bidirectional Forwarding Detection (BFD)



Task 1

Configure the routers based on the previous diagram. *Do not* configure any routing protocol.

Based on the diagram, the following configures IP Addressing on R5 and R6's G0/6 and G0/5 interfaces respectively. In addition, a loopback interface is created on each router and IP addresses are assigned to it as shown in the diagram:

On R5:

```
R5(config)#interface g0/6
R5(config-if)#ip address 56.1.1.5 255.255.255.0
R5(config-if)#no shut

R5(config)#interface lo0
R5(config-if)#ip address 5.5.5.5 255.0.0.0
```

On R6:

```
R6(config)#interface g0/5
R6(config-if)#ip address 56.1.1.6 255.255.255.0
R6(config-if)#no shut

R6(config)#interface lo0
R6(config-if)#ip address 6.6.6.6 255.0.0.0
```

Task 2

Configure EIGRP AS 100 on all directly connected interfaces of these two routers and ensure reachability. R5 should be configured using EIGRP classical mode, and R6 should use the EIGRP named mode configuration style.

Following configurations enable EIGRP using the classic mode on R5 and the named mode on R6. EIGRP is enabled on their directly connected interfaces and loopback 0 with the **network** statement

On R5:

```
R5(config)#router eigrp 100
R5(config-router)#network 5.5.5.5 0.0.0.0
R5(config-router)#network 56.1.1.5 0.0.0.0
```

On R6:

```
R6(config)#router eigrp aaa
R6(config-router)#address ipv4 as 100
R6(config-router-af)#network 6.6.6.6 0.0.0.0
R6(config-router-af)#network 56.1.1.6 0.0.0.0
```

You should see the following console message:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 56.1.1.5
(GigabitEthernet0/5) is up: new adjacency
```

To verify the configuration:

The **show ip route eigrp** on both routers verify they have learned EIGRP routes from each other:

```
R6#show ip route eigrp | begin Gate
Gateway of last resort is not set

D     5.0.0.0/8 [90/2570240] via 56.1.1.5, 00:01:22, GigabitEthernet0/5
```

On R5:

```
R5#show ip route eigrp | begin Gate
Gateway of last resort is not set

D     6.0.0.0/8 [90/2848] via 56.1.1.6, 00:02:48, GigabitEthernet0/6
```

Task 3

Configure and test BFD on these two routers.

Being able to detect whether a routing peer is down rapidly is imperative to the route reconvergence process. It is so essential that each routing protocol institutes its own method of neighbor-down detection. These detection mechanisms rely on the receipt of hello or keepalive packets at a predefined interval. The problem with such systems is that most routing protocols are unable to achieve sub-second convergence with their default timers. Even with some sub-second timers, the processing overhead required to keep track of this process can become a resource drain on the router. In addition, if the network environment includes multiple routing protocols, the differing timer values can make it inconsistent for building automated processes around detection and recovery for neighbor down events.

The Bidirectional Forwarding Detection (BFD) protocol supported on Cisco platforms is a protocol that is designed to provide fast neighbor down detection due to a failure in the forwarding path to a specific destination. This protocol can be used to replace the normal routing protocol hello and keepalive mechanisms. When used in this way, it provides a more consistent and predictable neighbor-down detection system.

BFD works by establishing BFD sessions between neighbors. Each session is configured with different parameters that determine when the router will consider the neighbor peer as having gone down. Once BFD neighbors are established, BFD can be configured to send BFD echo packets at a specified interval to the BFD neighbor. These echo packets are forwarded in hardware, requiring little processor interaction to process.

The BFD echo packet is a special packet sent by the router with the source and destination address equal to its own interface IP address. When received by the BFD neighbor, the BFD neighbor forwards this packet back to the local router in hardware. When the local router receives this packet back, it knows the remote peer is still online.

The BFD session state can be linked to routing protocol neighborships. If the BFD neighbor session goes down, then the routing protocol neighborship is torn down. This process is called registering the protocol to participate in BFD.

BFD is configured to run in two steps:

1. Configure the BFD session on a per-interface basis
2. Register the BFD session with the routing protocol of choice

The following configures BFD to run between R5 and R6 for fast failure detection.

In the first step, BFD is enabled on the G0/6 and G0/5 interfaces on R5 and R6 with the **bfd interval *milliseconds* min_rx *milliseconds* multiplier *interval-multiplier*** command. This command allows for specifying certain BFD parameters such as how often to transmit UDP echoes. The **interval** timer dictates the rate at which the UDP echoes are generated. This value is set to 100ms on both R5 and R6. This means every 100ms R5 and R6 will send a UDP BFD echo to each other.

The **min_rx** is the minimum receive time. It is the interval at which the local router expects to receive BFD control packets. If a control packet is not received within this time, then the router considers that packet as a missed packet. If the value specified here is 100 ms, the neighboring router should be configured to send BFD packets equal or less than 100ms. In this case, this value is set to 100 ms as well on both routers. Finally, the **multiplier** value is set to 3. The multiplier value specifies the number of packets that can be missed before BFD declares the peer down.

STEP 1:

On R5:

```
R5(config)#interface g0/6
R5(config-if)#bfd interval 100 min_rx 100 multiplier 3
```

You should see the following console message:

```
*Aug  4 12:58:35.439: %BFD-6-BFD_IF_CONFIGURE: BFD-SYSLOG: bfd config
apply, idb:GigabitEthernet0/6
```

On R6:

```
R6(config)#interface g0/5
R6(config-if)#bfd interval 100 min_rx 100 multiplier 3
```

In Step 2, BFD is registered with EIGRP. BFD can be applied to a single interface, or all interfaces. For this task, BFD for EIGRP is enabled only on the interface connecting R5 and R6. As such, in case of the classic mode EIGRP configuration on R5, BFD is registered with EIGRP under the router configuration mode with the **bfd interface G0/6** command. In case of EIGRP named mode on R6, BFD is registered to run on the G0/5 interface under the **af-interface G0/5** configuration mode with the **bfd** command:

STEP 2:

On R5:

```
R5(config)#router eigrp 100
R5(config-router)#bfd interface g0/6
```

On R6:

```
R6(config)#router eigrp aaa
R6(config-router)#address-family ipv4 as 100
R6(config-router-af)#af-interface g0/5
R6(config-router-af-interface)#bfd
```

The **show bfd neighbor** command can be used to verify the BFD session:

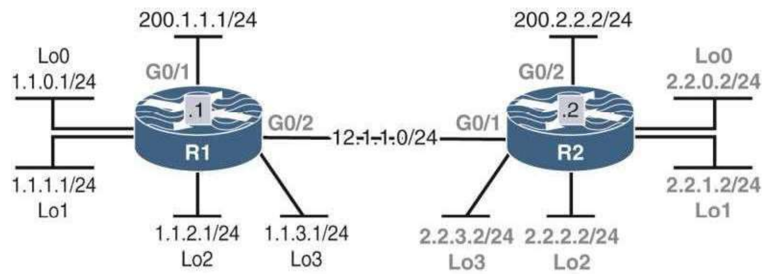
To verify the configuration:

```
R6#sh bfd neighbors

IPv4 Sessions
NeighAddr                              LD/RD         RH/RS     State     Int
56.1.1.5                                1/1          Up        Up        Gi0/5
```

Task 4

Erase the startup configuration of these two routers and reload the devices before proceeding to the next lab.

Lab 3
EIGRP Stub



Lab Setup:

To copy and paste the initial configurations, go to the Initial-config folder → EIGRP folder → Lab-3.

Task 1

Configure EIGRP AS 100 on the G0/2 and G0/1 interfaces of R1 and R2, respectively, as well as on all loopback interfaces of these two routers. On R1 configure EIGRP using the classic mode, and on R2 configure EIGRP in named mode to accomplish this task. *Do not* run EIGRP on the G0/1 interface of R1 or the G0/2 interface of R2.

The following configures EIGRP in classic mode on R1 and EIGRP in named mode on R2. The **network** command then enables EIGRP on all the interfaces except the G0/1 and G0/2 interfaces on R1 andd R2 respectively:

On R1:

```
R1(config)#router eigrp 100
R1(config-router)#network 1.1.0.1 0.0.0.0
R1(config-router)#network 1.1.1.1 0.0.0.0
R1(config-router)#network 1.1.2.1 0.0.0.0
R1(config-router)#network 1.1.3.1 0.0.0.0
```

```
R1(config-router)#network 12.1.1.1 0.0.0.0
```

On R2:

```
R2(config)#router eigrp tst
R2(config-router)#address-family ipv4 as 100
R2(config-router-af)#network 2.2.0.2 0.0.0.0
R2(config-router-af)#network 2.2.1.2 0.0.0.0
R2(config-router-af)#network 2.2.2.2 0.0.0.0
R2(config-router-af)#network 2.2.3.2 0.0.0.0
R2(config-router-af)#network 12.1.1.2 0.0.0.0
```

You should see the following console message:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: neighbor 12.1.1.1 (GigabitEthernet0/1)
is up: new adjacency
```

To verify the configuration:

The **show ip route eigrp** command output on each router verifies the EIGRP learned routes:

On R1:

```
R1#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      2.0.0.0/24 is subnetted, 4 subnets
D        2.2.0.0 [90/2848] via 12.1.1.2, 00:00:40, GigabitEthernet0/2
D        2.2.1.0 [90/2848] via 12.1.1.2, 00:00:40, GigabitEthernet0/2
D        2.2.2.0 [90/2848] via 12.1.1.2, 00:00:40, GigabitEthernet0/2
D        2.2.3.0 [90/2848] via 12.1.1.2, 00:00:40, GigabitEthernet0/2
```

On R2:

```
R2#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/24 is subnetted, 4 subnets
D        1.1.0.0 [90/2570240] via 12.1.1.1, 00:01:06, GigabitEthernet0/1
D        1.1.1.0 [90/2570240] via 12.1.1.1, 00:01:06, GigabitEthernet0/1
D        1.1.2.0 [90/2570240] via 12.1.1.1, 00:01:06, GigabitEthernet0/1
D        1.1.3.0 [90/2570240] via 12.1.1.1, 00:01:06, GigabitEthernet0/1
```

Task 2

Configure R1 and R2 to summarize their loopback interfaces in EIGRP.

In the classic EIGRP, summarization is performed under the interface. R1 running EIGRP in classic mode is configured with the **ip summary-address eigrp 100.1.0.0 255.255.252.0** command to summarize its loopback addresses.

On R1:

```
R1(config)#interface g0/2
R1(config-if)#ip summary-address eigrp 100 1.1.0.0 255.255.252.0
```

For named mode, summarization is configured with the **summary-address** command under the **af-interface** for the **address-family for IPv4** for EIGRP process **tst.** This is seen below on R2. Notice how IOS allows the use of prefix length while summarizing instead of entering the full subnet mask:

On R2:

```
R2(config)#router eigrp tst
R2(config-router)#address-family ipv4 as 100
R2(config-router-af)#af-interface g0/1
R2(config-router-af-interface)#summary-address 2.2.0.0/22
```

The result of the above configurations can be viewed on the routing tables on R1 and R2 as seen below. As a result of summarization, R1 no longer advertises specific prefixes to R2. It instead advertises the summary route 1.1.0.0/22. Similarly, R2 no longer advertises specific prefixes, instead advertises the summary route 2.2.0.0/22:

To verify the configuration:

On R1:

```
R1#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
D        1.1.0.0/22 is a summary, 00:01:56, Null0
      2.0.0.0/22 is subnetted, 1 subnets
D        2.2.0.0 [90/2848] via 12.1.1.2, 00:01:16, GigabitEthernet0/2
```

On R2:

```
R2#show ip route eigrp | begin Gateway
Gateway of last resort is not set
```

```
      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/2570240] via 12.1.1.1, 00:02:24, GigabitEthernet0/1
      2.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
D        2.2.0.0/22 is a summary, 00:01:44, Null0
```

Notice the null route created on each router. A null route is created when EIGRP summarization is configured. The null interface is basically a discard interface. A router may receive traffic to an address that is covered by the summary range, but the address does not actually exist. In this case, traffic to a non-existent IP address is forwarded to the null interface to be discarded.

Earlier versions of IOS allowed removing the null route by setting the administrative distance of a null 0 route to 255. A route with an AD of 255 is deemed unreachable and isn't installed in the routing table. The distance value was appended to the summary command. For example : **ip summary-address eigrp 100 1.1.0.0 255.255.252.0 255.** Newer IOS versions however have deprecated the use of this syntax preventing the removal of Null0 route. Issuing this command on the router results in the following message:

```
R1(config-if)#ip summary-address eigrp 100 10.1.0.0 255.255.0.0 255
```

%EIGRP: summary-address accepted but distance option deprecated; use summary-metric command for distance.

Task 3

Configure the following static routes on R1 and R2 and redistribute them into EIGRP:

- On R1: 11.0.0.0/8 via G0/1
- On R2: 22.0.0.0/8 via G0/2

Static routes are configured with the **ip route** command on R1 and R2. These routes are then redistributed into EIGRP using the **redistribute static** command. In classic EIGRP configuration mode on R1, this command is issued in EIGRP router configuration mode. In named EIGRP configuration mode, redistribution configuration is applied under the address-family IPv4 topology base configuration mode as seen on R2 below:

On R1:

```
R1(config)#ip route 11.0.0.0 255.0.0.0 g0/3

R1(config)#router eigrp 100
R1(config-router)#redistribute static
```

On R2:

```
R2(config)#ip route 22.0.0.0 255.0.0.0 g0/3

R2(config)#router eigrp tst
R2(config-router)#address-family ipv4 as 100
R2(config-router-af)#topology base
R2(config-router-af-topology)#redistribute static
```

Notice in the above how the **redistribute** statement does not include a seed metric. In earlier sections, it was explained that EIGRP requires a seed metric whenever redistributing routes from another source routing protocol that isn't EIGRP. EIGRP, however, does not require a seed metric whenever redistributing static or directly-connected routes from the routing table. EIGRP simply derives its component metrics from the exit interface of the static or directly-connected route.

The **show ip route eigrp** command output below verifies the redistributed static routes at each end as external EIGRP routes:

To verify the configuration:

On R1:

```
R1#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
D        1.1.0.0/22 is a summary, 00:05:32, Null0
      2.0.0.0/22 is subnetted, 1 subnets
D        2.2.0.0 [90/2848] via 12.1.1.2, 00:04:52, GigabitEthernet0/2
D EX  22.0.0.0/8 [170/3072] via 12.1.1.2, 00:00:42, GigabitEthernet0/2
```

On R2:

```
R2#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/2570240] via 12.1.1.1, 00:06:07, GigabitEthernet0/1
      2.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
D        2.2.0.0/22 is a summary, 00:05:27, Null0
D EX  11.0.0.0/8 [170/15360] via 12.1.1.1, 00:01:48, GigabitEthernet0/1
```

Task 4

Advertise the G0/1 interface of R1 and the G0/2 interface of R2 into RIPv2 and disable auto-summarization. You should redistribute RIPv2 into EIGRP and use any metric for the redistributed routes.

The following first configures RIP on R1 and R2. The RIP version is specified with the **version 2** command and auto-summarization is disabled with the **no auto-summary** command. RIP is then enabled on the interfaces specified in the task with the **network** statement:

On R1:

```
R1(config)#router rip
R1(config-router)#no auto-summary
R1(config-router)#version 2
R1(config-router)#network 200.1.1.0
```

On R2:

```
R2(config)#router rip
R2(config-router)#no auto-summary
R2(config-router)#version 2
R2(config-router)#network 200.2.2.0
```

R1 and R2 are then configured to redistribute RIP into EIGRP. In classic EIGRP configuration mode, the **redistribute** command is issued in EIGRP router configuration mode. In named EIGRP configuration mode, redistribution configuration is applied under the address-family IPv4 topology base configuration mode:

On R1:

```
R1(config)#router eigrp 100
R1(config-router)#redistribute rip metric 1000000 10 255 1 1500
```

On R2:

```
R2(config)#router eigrp tst
R2(config-router)#address-family ipv4 as 100
R2(config-router-af)#topology base
R2(config-router-af-topology)#redistribute rip metric 1000000 10 255 1
1500
```

To verify the configuration:

The **show ip route eigrp** command output on both routers verify the redistributed routes as EIGRP external routes:

On R1:

```
R1#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
D        1.1.0.0/22 is a summary, 00:08:40, Null0
      2.0.0.0/22 is subnetted, 1 subnets
D        2.2.0.0 [90/2848] via 12.1.1.2, 00:08:40, GigabitEthernet0/2
D EX  22.0.0.0/8 [170/3072] via 12.1.1.2, 00:08:40, GigabitEthernet0/2
D EX  200.2.2.0/24 [170/5376] via 12.1.1.2, 00:00:34, GigabitEthernet0/2
```

On R2:

```
R2#show ip route eigrp | begin Gateway
Gateway of last resort is not set

      1.0.0.0/22 is subnetted, 1 subnets
D        1.1.0.0 [90/2570240] via 12.1.1.1, 00:18:39, GigabitEthernet0/1
      2.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
D        2.2.0.0/22 is a summary, 00:17:59, Null0
D EX  11.0.0.0/8 [170/15360] via 12.1.1.1, 00:14:20, GigabitEthernet0/1
D EX  200.1.1.0/24 [170/61440] via 12.1.1.1, 00:01:54, GigabitEthernet0/1
```

Task 5

Configure EIGRP stub routing on R1 by using the command **eigrp stub** connected. Test this option and verify the routes in the routing tables of both routers.

EIGRP supports what is known as the EIGRP stub routing feature. It is designed as a mechanism to help constrain EIGRP query messages that are generated whenever an EIGRP router loses a route in its topology table. Under normal circumstances, whenever an EIGRP router loses a route in its topology table, it sends an EIGRP QUERY message to all of its neighbors. The QUERY message is basically the router's way of asking if its neighbors has another route to reach the network the local router lost. The local router will wait until it receives REPLY messages from all neighbors to which it sent a QUERY. This process is called "going active" for a route because EIGRP is actively trying to determine the status of the route.

Some routers will not have alternate routes to a lost network. This applies particularly to routers in a hub-and-spoke network. In this case, spokes receive their routes from the hub. If the hub goes down, there is no reason for the hub to query the spokes for alternate paths. The hub is essentially wasting its efforts. The EIGRP stub router feature gives stub routers (routers that only have a single connection to the network like the spokes in the hub-and-spoke example) a way to signal to the upstream neighbor that they have no alternate routes to networks. A router configured as stub sets the stub flag in its EIGRP hello packet it sends to its neighbors. Upon seeing this, the neighbor will not query the stub router for lost routes.

The stub configuration can also be used to limit which routes the stub router will advertise to its neighbors. The major options are "connected", "summary-only", and "redistributed." The next tasks demonstrate the effects of each configuration command.

In case of the classic EIGRP mode, the **eigrp stub** command under the EIGRP router configuration mode is used to configure EIGRP stub routing. In the case of EIGRP named mode, stub routing is configured in address-family for IPv4 configuration mode. Various options are available when configuring EIGRP stub routing as seen below:

EIGRP Stub routing configuration in classic mode

```
R1(config)#router eigrp 100
R1(config-router)#eigrp stub ?
Connected.     Do advertise connected routes
leak-map.      Allow dynamic prefixes based on the leak-map
receive-only.  Set receive only neighbor
redistributed Do advertise redistributed routes
static         Do advertise static routes
summary        Do advertise summary routes
```

EIGRP Stub routing configuration in named mode

```
R2(config)#router eigrp tst
R2(config-router)#address-family ipv4 unicast autonomous-system 100
R2(config-router-af)#eigrp stub ?
  connected      Do advertise connected routes
  leak-map       Allow dynamic prefixes based on the leak-map
  receive-only   Set receive only neighbor
  redistributed Do advertise redistributed routes
  static         Do advertise static routes
  summary        Do advertise summary routes
```

```
  <cr>
```

This task requires configuring R1 as stub with the **eigrp stub connected** command. The **eigrp stub connected** option causes the router to advertise only those connected routes that are covered by an EIGRP **network** statement. R1's configuration below shows a few **network** statements issued to advertise the connected loopbacks on R1 into EIGRP:

On R1:

```
R1#show run | section router eigrp

router eigrp 100
 network 1.1.0.1 0.0.0.0
 network 1.1.1.1 0.0.0.0
 network 1.1.2.1 0.0.0.0
 network 1.1.3.1 0.0.0.0
 network 12.1.1.1 0.0.0.0
 redistribute static
 redistribute rip metric 1000000 10 255 1 1500
```

With the **eigrp stub connected** command issued on R1 below, R1 will advertise the highlighted connected networks to R2. If any of the network statements covering the loopback addresses are removed from the above, those will not be advertised to R2.

On R1:

```
R1(config)#router eigrp 100
R1(config-router)#eigrp stub connected
```

R2's routing table is observed below. Notice how only connected routes covered by **network** statements on R1 are advertised to R2. The summary that was previously advertised to R2 is withdrawn and R1 instead advertises the specific prefixes that match **network** statements. In addition, the static and RIP routes that were redistributed into EIGRP on R1 are not present in R2's routing table. This is a result of configuring **eigrp stub connected** command that prevents R1 from advertising the D EX routes to R2.

To verify the configuration:

On R2:

```
R2#show ip route eigrp | include 12.1.1.1

D        1.1.0.0 [90/2570240] via 12.1.1.1, 00:02:21, GigabitEthernet0/1
```

```
D        1.1.1.0 [90/2570240] via 12.1.1.1, 00:02:21, GigabitEthernet0/1
D        1.1.2.0 [90/2570240] via 12.1.1.1, 00:02:21, GigabitEthernet0/1
D        1.1.3.0 [90/2570240] via 12.1.1.1, 00:02:21, GigabitEthernet0/1
```

Task 6

Remove the **eigrp stub** connected option configured in the previous task and reconfigure EIGRP stub routing on R1 by using the **eigrp stub summary** command. Test this option and verify the routes in the routing tables of both routers.

The earlier configuration is removed from R1 with the **no eigrp stub connected** command:

On R1:

```
R1(config)#router eigrp 100
R1(config-router)#no eigrp stub connected
```

This task requires configuring the **eigrp stub summary** command. This stub configuration with the summary option will cause the router to advertise summary routes. This means, only EIGRP summary routes are advertised to neighbors. Any other networks covered by the network statement will not be advertised. This would include routes that are not covered by the summary network. For example, if R1 advertised a route to the 1.1.4.0/24 network, it would not be advertised to R2 even though the summary 1.1.0.0/22 does not cover that network.

Task 2 of this section configured R1 to advertise a 1.1.0.0/22 summary route out its G0/2 interface to R2. Prior to this however, following is the state of the routing table on R2. R2 receives a summary route and EIGRP external routes from R1:

To verify the configuration:

On R2:

```
R2#show ip route eigrp | include 12.1.1.1

D        1.1.0.0 [90/2570240] via 12.1.1.1, 00:00:06, GigabitEthernet0/1
D EX  11.0.0.0/8 [170/15360] via 12.1.1.1, 00:00:06, GigabitEthernet0/1
D EX  200.1.1.0/24 [170/61440] via 12.1.1.1, 00:00:06, GigabitEthernet0/1
```

The **eigrp stub summary** command is issued on R1:

On R1: