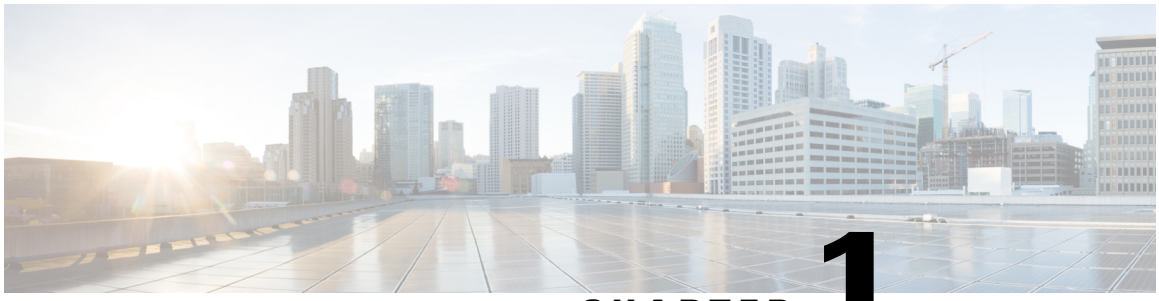
IP Routing: EIGRP Configuration Guide, Cisco IOS Release 15E

First Published: November 28, 2012

Last Modified: March 08, 2013

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Configuring EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco. The convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP, and IGRP is now obsolete.

The convergence technology of EIGRP is based on an algorithm referred to as the Diffusing Update Algorithm (DUAL). The algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize. Devices that are not affected by topology changes are not involved in recomputations.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Configuring EIGRP

EIGRP Features

- Increased network width—With IP Routing Information Protocol (RIP), the largest possible width of your network is 15 hops. When EIGRP is enabled, the largest possible width is increased to 255 hops, and the EIGRP metric is large enough to support thousands of hops. The default maximum number of EIGRP hops is 100.
- Fast convergence—The DUAL algorithm allows routing information to converge quickly.
- Partial updates—EIGRP sends incremental updates (instead of sending the entire contents of the routing table) when the state of a destination changes. This feature minimizes the bandwidth required for EIGRP packets.
- Neighbor discovery mechanism—This is a simple, protocol-independent hello mechanism used to learn about neighboring devices.
- Variable-Length Subnet Masks (VLSMs).
- Arbitrary route summarization.
- Scaling—EIGRP scales to large networks.

EIGRP Autonomous System Configuration

Configuring the **router eigrp** command with the *autonomous-system-number* argument creates an EIGRP configuration called the EIGRP autonomous system configuration, or EIGRP classic mode. The EIGRP autonomous system configuration creates an EIGRP routing instance that can be used for exchanging routing information.

In EIGRP autonomous system configurations, EIGRP VPNs can be configured only under IPv4 address family configuration mode. A virtual routing and forwarding (VRF) instance and a route distinguisher must be defined before the address family session can be created.

When the address family is configured, we recommend that you configure an autonomous system number either by using the *autonomous-system-number* argument with the **address-family** command or by using the **autonomous-system** command.

EIGRP Named Configuration

Configuring the **router eigrp** command with the *virtual-instance-name* argument creates an EIGRP configuration referred to as the EIGRP named configuration or EIGRP named mode. An EIGRP named configuration does not create an EIGRP routing instance by itself; it is a base configuration that is required to define address-family configurations that are used for routing.

In EIGRP named configurations, EIGRP VPNs can be configured in IPv4 and IPv6 named configurations. A VRF instance and a route distinguisher must be defined before the address family session can be created.

A single EIGRP routing process can support multiple VRFs. The number of VRFs that can be configured is limited only by the available system resources on the device, which is determined by the number running processes and available memory. However, only a single VRF can be supported by each VPN, and redistribution between different VRFs is not supported.

EIGRP IPv6 VRF-Lite

The EIGRP IPv6 VRF-Lite feature provides EIGRP IPv6 support for multiple VRFs and simplifies the management and troubleshooting of traffic belonging to a specific VRF.

Note The EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations.

EIGRP vNETs

The EIGRP vNET feature uses Layer 3 routing techniques to provide limited fate sharing (the term fate sharing refers to the failure of interconnected systems; that is, different elements of a network are interconnected in such a way that they either fail together or not at all), traffic isolation, and access control with simple configurations. EIGRP virtual network (vNET) configurations are supported in both autonomous-system configurations and named configurations.

The vNET feature allows you to have multiple virtual networks by utilizing a single set of routers and links provided by the physical topology. Routers and links can be broken down into separate virtual networks using separate routing tables and routing processes by using vNETs and VRF configuration commands. The virtual networks facilitate traffic isolation and limited fate sharing. EIGRP's primary role in vNETs is to populate routing tables used by each vNET so that appropriate forwarding can take place. In the vNET model, each vNET effectively has its own complete set of EIGRP processes and resources, thus minimizing the possibility of actions within one vNET affecting another vNET.

The vNET feature supports command inheritance that allows commands entered in interface configuration mode to be inherited by every vNET configured on that interface. These inherited commands, including EIGRP interface commands, can be overridden by vNET-specific configurations in vNET submodes under the interface.

The following are some of the limitations of EIGRP vNETs:

- EIGRP does not support Internetwork Packet Exchange (IPX) within a vNET.
- vNET and VRF configurations are mutually exclusive on an interface. Both VRFs and vNETs can be configured on the router, but they cannot both be defined on the same interface. A VRF cannot be configured within a vNET and a vNET cannot be configured within a VRF.
- Each vNET has its own routing table, and routes cannot be redistributed directly from one vNET into another. EIGRP uses the route replication functionality to meet the requirements of shared services and to copy routes from one vNET Routing Information Base (RIB) to other vNET RIBs.

EIGRP vNET Interface and Command Inheritance

A vNET router supports two types of interfaces: Edge interface and core (shared) interface.

An edge interface is an ingress point for vNET-unaware networks and is restricted to a single VRF. Use the **vrf forwarding** command to associate the edge interface with a VRF. The **vrf forwarding** command also allows entry into VRF submodes used to define interface settings on a per-VRF basis.

A vNET core interface is used to connect vNET-aware systems and can be shared by multiple vNETs. Use the **vnet trunk** command to enable a core interface.

When the **vnet trunk** command exists on an interface, with or without a VRF list, any EIGRP interface commands on that interface will be applied to the EIGRP instance for every vNET on that interface, including the instance running on the base or the global RIB. If the **vnet trunk** command is deleted from the interface, EIGRP interface commands will remain on and apply to only the global EIGRP instance. If an EIGRP interface command is removed from the main interface, the command will also be removed from every vNET on that interface.

End systems or routing protocol peers reached through an edge interface are unaware of vNETs and do not perform the vNET tagging done in the core of the vNET network.

EIGRP also supports the capability of setting per-vNET interface configurations, which allow you to define interface attributes that influence EIGRP behavior for a single vNET. In the configuration hierarchy, a specific vNET interface setting has precedence over settings applied to the entire interface and inherited by each vNET configured on that interface.

EIGRP provides interface commands to modify the EIGRP-specific attributes of an interface, and these interface commands can be entered directly on the interface for EIGRP autonomous system configurations, or in address family interface configuration mode for the EIGRP named mode configurations.

EIGRP Neighbor Relationship Maintenance

Neighbor relationship maintenance is the process that devices use to dynamically learn of other devices on their directly attached networks. Devices must also discover when their neighbors become unreachable or inoperative. Neighbor relationship maintenance is achieved with low overhead by devices when they periodically send small hello packets to each other. As long as hello packets are received, the Cisco software can determine whether a neighbor is alive and functioning. After the status of the neighbor is determined, neighboring devices can exchange routing information.

The reliable transport protocol is responsible for the guaranteed, ordered delivery of Enhanced Interior Gateway Routing Protocol (EIGRP) packets to all neighbors. The reliable transport protocol supports intermixed transmission of multicast and unicast packets. Some EIGRP packets (such as updates) must be sent reliably; this means that the packets require acknowledgment from the destination. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, hello packets need not be sent reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello packet with an indication in the packet informing receivers that the packet need not be acknowledged. The reliable transport protocol can send multicast packets quickly when unacknowledged packets are pending, thereby ensuring that the convergence time remains low in the presence of varying speed links.

Some EIGRP remote unicast-listen (any neighbor that uses unicast to communicate) and remote multicast-group neighbors may peer with any device that sends a valid hello packet, thus causing security concerns. By authenticating the packets that are exchanged between neighbors, you can ensure that a device accepts packets only from devices that know the preshared authentication key.

Neighbor Authentication

The authentication of packets being sent between neighbors ensures that a device accepts packets only from devices that have the same preshared key. If this authentication is not configured, you can intentionally or

accidentally add another device to the network or send packets with different or conflicting route information onto the network, resulting in topology corruption and denial of service (DoS).

Enhanced Interior Gateway Routing Protocol (EIGRP) authentication is configurable on a per-interface basis; packets exchanged between neighbors connected through an interface are authenticated. EIGRP supports message digest algorithm 5 (MD5) authentication to prevent the introduction of unauthorized information from unapproved sources. MD5 authentication is defined in RFC 1321.

DUAL Finite State Machine

The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as the metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on feasible successors. A successor is a neighboring device (used for packet forwarding) that has the least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but only neighbors advertising the destination, a recomputation must occur to determine a new successor. The time required to recompute the route affects the convergence time. Recomputation is processor-intensive, and unnecessary recomputation must be avoided. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, DUAL will use any feasible successors it finds to avoid unnecessary recomputation.

Protocol-Dependent Modules

Protocol-dependent modules are responsible for network-layer protocol-specific tasks. An example is the EIGRP module, which is responsible for sending and receiving EIGRP packets that are encapsulated in the IP. The EIGRP module is also responsible for parsing EIGRP packets and informing DUAL about the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IP routing table. Also, EIGRP is responsible for redistributing routes learned from other IP routing protocols.

EIGRP Metric Weights

You can use the **metric weights** command to adjust the default behavior of Enhanced Interior Gateway Routing Protocol (EIGRP) routing and metric computations. EIGRP metric defaults (K values) have been carefully selected to provide optimal performance in most networks.

Note Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default K values without guidance from an experienced network designer.

By default, the EIGRP composite cost metric is a 32-bit quantity that is the sum of segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. The formula used to scale and invert the bandwidth value is 10^7/minimum bandwidth in kilobits per second. However, with the EIGRP Wide Metrics feature, the EIGRP composite cost metric is scaled to include 64-bit metric calculations for EIGRP named mode configurations.

For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Gigabit Ethernet (GE), and serial lines running from 9600 bits per second to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Mismatched K Values

EIGRP K values are the metrics that EIGRP uses to calculate routes. Mismatched K values can prevent neighbor relationships from being established and can negatively impact network convergence. The example given below explains this behavior between two EIGRP peers (Device-A and Device-B).

The following configuration is applied to Device-A. The K values are changed using the **metric weights** command. A value of 2 is entered for the *k1* argument to adjust the bandwidth calculation. A value of 1 is entered for the *k3* argument to adjust the delay calculation.

```
Device(config)# hostname Device-A
Device-A(config)# interface serial 0
Device-A(config-if)# ip address 10.1.1.1 255.255.255.0
Device-A(config-if)# exit
Device-A(config)# router eigrp name1
Device-A(config-router)# address-family ipv4 autonomous-system 4533
Device-A(config-router-af)# network 10.1.1.0 0.0.0.255
Device-A(config-router-af)# metric weights 0 2 0 1 0 0 1
```

The following configuration is applied to Device-B, and the default K values are used. The default K values are 1, 0, 1, 0, 0, and 0.

```
Device(config)# hostname Device-B
Device-B(config)# interface serial 0
Device-B(config-if)# ip address 10.1.1.2 255.255.255.0
Device-B(config-if)# exit
Device-B(config)# router eigrp name1
Device-B(config-router)# address-family ipv4 autonomous-system 4533
Device-B(config-router-af)# network 10.1.1.0 0.0.0.255
Device-B(config-router-af)# metric weights 0 1 0 1 0 0 0
```

The bandwidth calculation is set to 2 on Device-A and set to 1 (by default) on Device-B. This configuration prevents these peers from forming a neighbor relationship.

The following error message is displayed on the console of Device-B because the K values are mismatched:

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1 (Ethernet0/0) is
 down: K-value mismatch
```

The following are two scenarios where the above error message can be displayed:

- Two devices are connected on the same link and configured to establish a neighbor relationship. However, each device is configured with different K values.
- One of two peers has transmitted a “peer-termination” message (a message that is broadcast when an EIGRP routing process is shut down), and the receiving device does not support this message. The receiving device will interpret this message as a K-value mismatch.

EIGRP Wide Metrics

The Enhanced Interior Gateway Routing Protocol (EIGRP) composite cost metric (calculated using the bandwidth, delay, reliability, load, and K values) is not scaled correctly for high-bandwidth interfaces or Ethernet channels, resulting in incorrect or inconsistent routing behavior. The lowest delay that can be configured for an interface is 10 microseconds. As a result, high-speed interfaces, such as 10 Gigabit Ethernet (GE) interfaces, or high-speed interfaces channeled together (GE ether channel) will appear to EIGRP as a single GE interface. This may cause undesirable equal-cost load balancing. To resolve this issue, the EIGRP Wide Metrics feature supports 64-bit metric calculations and Routing Information Base (RIB) scaling that

provide the ability to support interfaces (either directly or via channeling techniques like port channels or ether channels) up to approximately 4.2 terabits.

Note The 64-bit metric calculations work only in EIGRP named mode configurations. EIGRP classic mode uses 32-bit metric calculations.

To accommodate interfaces with bandwidths above 1 gigabit and up to 4.2 terabits and to allow EIGRP to perform path selections, the EIGRP composite cost metric formula is modified. The paths are selected based on the computed time. The time that information takes to travel through links is measured in picoseconds. The interfaces can be directly capable of these high speeds, or the interfaces can be bundles of links with an aggregate bandwidth greater than 1 gigabit.

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

Default K values are as follows:

- K1 = K3 = 1
- K2 = K4 = K5 = 0
- K6 = 0

The EIGRP Wide Metrics feature also introduces K6 as an additional K value for future use.

By default, the path selection scheme used by EIGRP is a combination of throughput (rate of data transfer) and latency (time taken for data transfer), and the formula for calculating the composite cost metric is as follows:

Composite Cost Metric = (K1*Minimum Throughput) + (K3*Total Latency)

Minimum Throughput = (10^7* 65536)/Bw), where 65536 is the wide-scale constant.

Total Latency for bandwidths below 1 gigabit = (Delay*65536)/10, where 65536 is the wide-scale constant.

Total Latency for bandwidths above 1 gigabit = (10^7* 65536/10)/ Bw, 65536 is the wide-scale constant.

With the calculation of larger bandwidths, EIGRP can no longer fit the computed metric into a 4-byte unsigned long value that is needed by the Cisco RIB. To set the RIB scaling factor for EIGRP, use the **metric rib-scale** command. When you configure the **metric rib-scale** command, all EIGRP routes in the RIB are cleared and replaced with the new metric values.

Goodbye Message

The goodbye message is a feature designed to improve EIGRP network convergence. The goodbye message is broadcast when an EIGRP routing process is shut down to inform adjacent peers about an impending topology change. This feature allows supporting EIGRP peers to synchronize and recalculate neighbor relationships more efficiently than would occur if the peers discovered the topology change after the hold timer expired.

The following message is displayed by devices that run a supported release when a goodbye message is received:

```
*Apr 26 13:48:42.523: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1   (Ethernet0/0)
is down: Interface Goodbye received
```

A Cisco device that runs a software release that does not support the goodbye message can misinterpret the message as a K-value mismatch and display the following error message:

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor    10.1.1.1 (Ethernet0/0)
is down: K-value mismatch
```

Note The receipt of a goodbye message by a nonsupporting peer does not disrupt normal network operation. The nonsupporting peer terminates the session when the hold timer expires. The sending and receiving devices reconverge normally after the sender reloads.

Routing Metric Offset Lists

An offset list is a mechanism for increasing incoming and outgoing metrics to routes learned via EIGRP. Optionally, you can limit the offset list with either an access list or an interface.

Note Offset lists are available only in IPv4 configurations. IPv6 configurations do not support offset lists.

EIGRP Cost Metrics

When EIGRP receives dynamic raw radio link characteristics, it computes a composite EIGRP cost metric based on a proprietary formula. To avoid churn in the network as a result of a change in the link characteristics, a tunable dampening mechanism is used.

EIGRP uses metric weights along with a set of vector metrics to compute the composite metric for local RIB installation and route selections. The EIGRP composite cost metric is calculated using the formula:

EIGRP composite cost metric = 256*((K1*Bw) + (K2*Bw)/(256 – Load) + (K3*Delay)*(K5/(Reliability + K4)))

EIGRP uses one or more vector metrics to calculate the composite cost metric. The table below lists EIGRP vector metrics and their descriptions.

Table 1: EIGRP Vector Metrics

Vector Metric	Description
bandwidth	The minimum bandwidth of the route, in kilobits per second. It can be 0 or any positive integer. The bandwidth for the formula is scaled and inverted by the following formula: (10^7/minimum bandwidth (Bw) in kilobits per second)
delay	Route delay, in tens of microseconds.

Vector Metric	Description
delay reliability	The likelihood of successful packet transmission, expressed as a number between 0 and 255, where 255 means 100 percent reliability and 0 means no reliability.
load	The effective load of the route, expressed as a number from 0 to 255 (255 is 100 percent loading).
mtu	The minimum maximum transmission unit (MTU) size of the route, in bytes. It can be 0 or any positive integer.

EIGRP monitors metric weights on an interface to allow the tuning of EIGRP metric calculations and indicate the type of service (ToS). The table below lists the K values and their defaults.

Table 2: EIGRP K-Value Defaults

Setting	Default Value
K1	1
K2	0
K3	1
K4	0
K5	0

Most configurations use the delay and bandwidth metrics, with bandwidth taking precedence. The default formula of 256*(Bw + Delay) is the EIGRP metric. The bandwidth for the formula is scaled and inverted by the following formula:

(10^7/minimum Bw in kilobits per second)

Note You can change the weights, but these weights must be the same on all devices.

For example, look at a link whose bandwidth to a particular destination is 128 k and the delay is 84,000 microseconds.

By using a cut-down formula, you can simplify the EIGRP metric calculation to 256*(Bw + Delay), thus resulting in the following value:

Metric = 256*(10^7/128 + 84000/10) = 256*86525 = 22150400

To calculate route delay, divide the delay value by 10 to get the true value in tens of microseconds.

When EIGRP calculates the delay for Mobile Ad Hoc Networks (MANET) and the delay is obtained from a device interface, the delay is always calculated in tens of microseconds. In most cases, when using MANET, you will not use the interface delay, but rather the delay that is advertised by the radio. The delay you will receive from the radio is in microseconds, so you must adjust the cut-down formula as follows:

Metric = $(256*(10^7/128) + (84000*256)/10) = 20000000 + 2150400 = 22150400$

Route Summarization

You can configure EIGRP to perform automatic summarization of subnet routes into network-level routes. For example, you can configure subnet 172.16.1.0 to be advertised as 172.16.0.0 over interfaces that have been configured with subnets of 192.168.7.0. Automatic summarization is performed when two or more **network** router configuration or address family configuration commands are configured for an EIGRP process. This feature is enabled by default.

Route summarization works in conjunction with the **ip summary-address eigrp** command available in interface configuration mode for autonomous system configurations and with the **summary-address** (EIGRP) command for named configurations. You can use these commands to perform additional summarization. If automatic summarization is in effect, there usually is no need to configure network-level summaries using the **ip summary-address eigrp** command.

Summary Aggregate Addresses

You can configure a summary aggregate address for a specified interface. If there are specific routes in the routing table, EIGRP will advertise the summary address of the interface with a metric equal to the minimum metric of the specific routes.

Floating Summary Routes

A floating summary route is created by applying a default route and an administrative distance at the interface level or address family interface level. You can use a floating summary route when configuring the **ip summary-address eigrp** command for autonomous system configurations or the **summary-address** command for named configurations. The following scenarios illustrate the behavior of floating summary routes.

The figure below shows a network with three devices, Router-A, Router-B, and Router-C. Router-A learns a default route from elsewhere in the network and then advertises this route to Router-B. Router-B is configured so that only a default summary route is advertised to Router-C. The default summary route is applied to serial interface 0/1 on Router-B with the following autonomous system configuration:

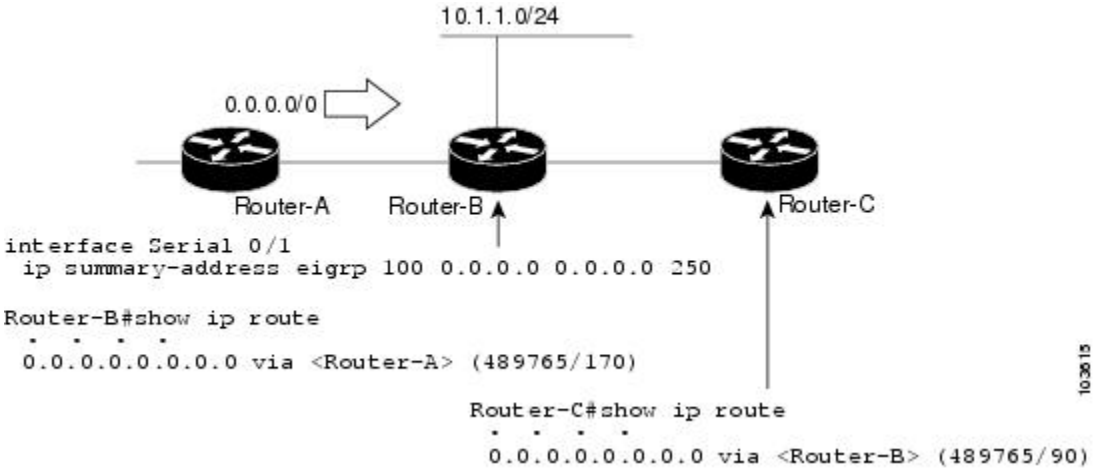
```
Router-B(config)# interface Serial 0/1
Router-B(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0
```

The default summary route is applied to serial interface 0/1 on Router-B with the following named configuration:

```
Router-B(config)# router eigrp 1
Router-B(config-router)# address-family ipv4 unicast vrf vrf1 autonomous-system 1
```

```
Router-B(config-router-af)# af-interface serial0/1
Router-B(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0 95
```

Figure 1: Floating Summary Route Applied to Router-B



The configuration of the default summary route on Router-B sends a 0.0.0.0/0 summary route to Router-C and blocks all other routes, including the 10.1.1.0/24 route, from being advertised to Router-C. However, this configuration also generates a local discard route—a route for 0.0.0.0/0 on the null 0 interface with an administrative distance of 5—on Router-B. When this route is created, it overrides the EIGRP-learned default route. Router-B will no longer be able to reach destinations that it would normally reach through the 0.0.0.0/0 route.

This problem is resolved by applying a floating summary route to the interface on Router-B that connects to Router-C. The floating summary route is applied by configuring an administrative distance for the default summary route on the interface of Router-B with the following statement for an autonomous system configuration:

```
Router-B(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0 250
```

The floating summary route is applied by configuring an administrative distance for the default summary route on the interface of Router-B with the following statement for a named configuration:

```
Router-B(config)# router eigrp 1
Router-B(config-router)# address-family ipv4 unicast vrf vrf1 autonomous-system 1
Router-B(config-router-af)# af-interface serial0/1
Router-B(config-router-af-interface)# summary-address eigrp 100 0.0.0.0 0.0.0.0 250
```

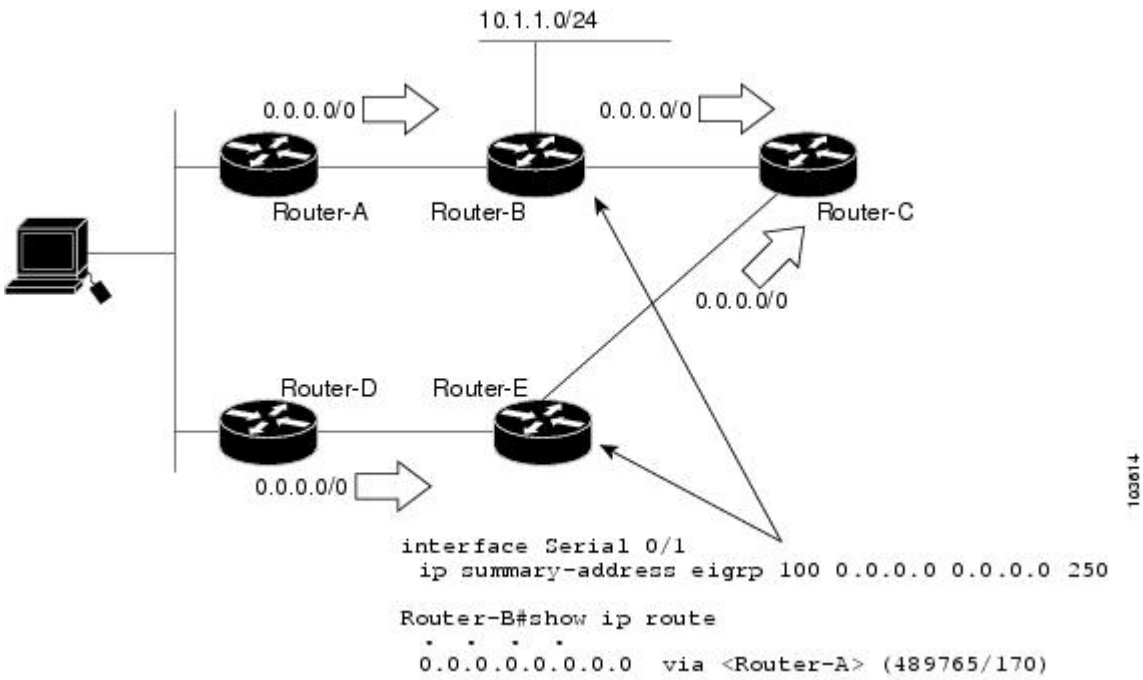
The administrative distance of 250, applied in the **summary-address** command, is now assigned to the discard route generated on Router-B. The 0.0.0.0/0, from Router-A, is learned through EIGRP and installed in the local routing table. Routing to Router-C is restored.

If Router-A loses the connection to Router-B, Router-B will continue to advertise a default route to Router-C, which allows traffic to continue to reach destinations attached to Router-B. However, traffic destined to networks connected to Router-A or behind Router-A will be dropped when the traffic reaches Router-B.

The figure below shows a network with two connections from the core, Router-A and Router-D. Both Router-B and Router-E have floating summary routes configured on the interfaces connected to Router-C. If the

connection between Router-E and Router-C fails, the network will continue to operate normally. All traffic will flow from Router-C through Router-B to hosts attached to Router-A and Router-D.

Figure 2: Floating Summary Route Applied for Dual-Homed Remotes



However, if the link between Router-A and Router-B fails, the network may incorrectly direct traffic because Router-B will continue to advertise the default route (0.0.0.0/0) to Router-C. In this scenario, Router-C still forwards traffic to Router-B, but Router-B drops the traffic. To avoid this problem, you should configure the summary address with an administrative distance only on single-homed remote devices or areas that have only one exit point between two segments of the network. If two or more exit points exist (from one segment of the network to another), configuring the floating default route can result in the formation of a black hole route (a route that has quick packet dropping capabilities).

EIGRP Route Authentication

EIGRP route authentication provides MD5 authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

Each key has its own key identifier (specified with the **key** *number* key chain configuration command), which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and the MD5 authentication key in use.

You can configure multiple keys with specific lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in the order from lowest to highest, and uses the first valid key that it encounters. Note that the device needs to know the time to configure keys with lifetimes.

Hello Packets and the Hold-Time Intervals

You can adjust the interval between hello packets and the hold time. Hello packets and hold-time intervals are protocol-independent parameters that work for IP and Internetwork Packet Exchange (IPX).

Routing devices periodically send hello packets to each other to dynamically learn of other devices on their directly attached networks. This information is used to discover neighbors and to learn when neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast multiaccess (NBMA) media, where the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower, as specified with the **bandwidth** interface configuration command. The default hello interval remains 5 seconds for high-speed NBMA networks. Note that for the purposes of EIGRP, Frame Relay and Switched Multimegabit Data Service (SMDS) networks may or may not be considered to be NBMA. These networks are considered NBMA only if the interface has not been configured to use physical multicasting.

You can configure the hold time on a specified interface for a particular EIGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval or 15 seconds. For slow-speed NBMA networks, the default hold time is 180 seconds.

On very congested and large networks, the default hold time might not be sufficient for all devices to receive hello packets from their neighbors. In such cases, you may want to increase the hold time.

Note Do not adjust the hold time without informing your technical support personnel.

Split Horizon

Split horizon controls the sending of EIGRP update and query packets. Split horizon is a protocol-independent parameter that works for IP and IPX. When split horizon is enabled on an interface, update and query packets are not sent to destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a device out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDS), situations can arise for which this behavior is less than ideal. In such situations and in networks that have EIGRP configured, you may want to disable split horizon.

EIGRP Dual DMVPN Domain Enhancement

The EIGRP Dual DMVPN Domain Enhancement feature supports the **no next-hop self** command on dual Dynamic Multipoint VPN (DMVPN) domains in both IPv4 and IPv6 configurations.

EIGRP, by default, sets the local outbound interface as the next-hop value while advertising a network to a peer, even when advertising routes out of the interface on which the routes were learned. This default setting can be disabled by using the **no ip next-hop-self** command in autonomous system configurations or the **no next-hop-self** command in named configurations. When the **next-hop self** command is disabled, EIGRP does

not advertise the local outbound interface as the next hop if the route has been learned from the same interface. Instead, the received next-hop value is used to advertise learned routes. However, this functionality only evaluates the first entry in the EIGRP table. If the first entry shows that the route being advertised is learned on the same interface, then the received next hop is used to advertise the route. The **no next-hop-self** configuration ignores subsequent entries in the table, which may result in the **no-next-hop-self** configuration being dishonored on other interfaces.

The EIGRP Dual DMVPN Domain Enhancement feature introduces the **no-ecmp-mode** keyword, which is an enhancement to the **no next-hop-self** and **no ip next-hop-self** commands. When this keyword is used, all routes to a network in the EIGRP table are evaluated to check whether routes advertised from an interface were learned on the same interface. If a route advertised by an interface was learned on the same interface, the **no next-hop-self** configuration is honored and the received next hop is used to advertise this route.

Link Bandwidth Percentage

By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth when configured with the **bandwidth** interface configuration command for autonomous system configurations and with the **bandwidth-percent** command for named configurations. You might want to change the bandwidth value if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (which may have been configured to influence route metric calculations). This is a protocol-independent parameter that works for IP and IPX.

EIGRP Stub Routing

The EIGRP stub routing feature improves network stability, reduces resource utilization, and simplifies the stub device configuration.

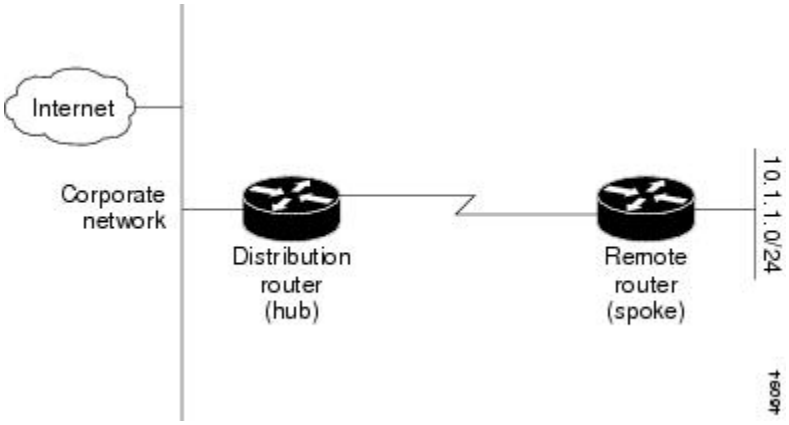
Stub routing is commonly used in hub-and-spoke network topologies. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote device (the spoke) that is connected to one or more distribution devices (the hub). The remote device is adjacent to one or more distribution devices. The only route for IP traffic to reach the remote device is through a distribution device. This type of configuration is commonly used in WAN topologies, where the distribution device is directly connected to a WAN. The distribution device can be connected to many remote devices, which is often the case. In a hub-and-spoke topology, the remote device must forward all nonlocal traffic to a distribution device, so it becomes unnecessary for the remote device to have a complete routing table. Generally, the distribution device need not send anything more than a default route to the remote device.

When using the EIGRP stub routing feature, you need to configure the distribution and remote devices to use EIGRP and configure only the remote device as a stub. Only specified routes are propagated from the remote (stub) device. The stub device responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” A device that is configured as a stub will send a special peer information packet to all neighboring devices to report its status as a stub device.

Any neighbor that receives a packet informing it of the stub status will not query the stub device for any routes, and a device that has a stub peer will not query that peer. The stub device will depend on the distribution device to send proper updates to all peers.

The figure below shows a simple hub-and-spoke network.

Figure 3: Simple Hub-and-Spoke Network



The stub routing feature by itself does not prevent routes from being advertised to the remote device. In the above example, the remote device can access the corporate network and the Internet only through the distribution device. Having a complete route table on the remote device would serve no functional purpose because the path to the corporate network and the Internet would always be through the distribution device. The large route table would only reduce the amount of memory required by the remote device. Bandwidth and memory can be conserved by summarizing and filtering routes in the distribution device. The remote device need not receive routes that have been learned from other networks because the remote device must send all nonlocal traffic, regardless of the destination, to the distribution device. If a true stub network is desired, the distribution device should be configured to send only a default route to the remote device. The EIGRP stub routing feature does not automatically enable summarization on distribution devices. In most cases, the network administrator will need to configure summarization on distribution devices.

Note When configuring the distribution device to send only a default route to the remote device, you must use the **ip classless** command on the remote device. By default, the **ip classless** command is enabled in all Cisco images that support the EIGRP stub routing feature.

Without the EIGRP stub routing feature, even after routes that are sent from the distribution device to the remote device have been filtered or summarized, a problem might occur. If a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution device, which in turn would send a query to the remote device, even if routes are being summarized. If there is a communication problem (over the WAN link) between the distribution device and the remote device, an EIGRP stuck in active (SIA) condition could occur and cause instability elsewhere in the network. The EIGRP stub routing feature allows a network administrator to prevent queries from being sent to the remote device.

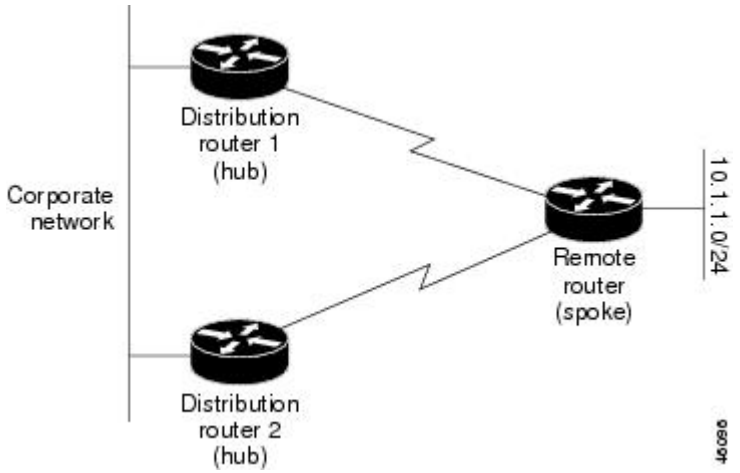
Dual-Homed Remote Topology

In addition to a simple hub-and-spoke network, where a remote device is connected to a single distribution device, the remote device can be dual-homed to two or more distribution devices. This configuration adds redundancy and introduces unique issues, and the stub feature helps to address some of these issues.

A dual-homed remote device will have two or more distribution (hub) devices. However, the principles of stub routing are the same as they are with a hub-and-spoke topology. The figure below shows a common

dual-homed remote topology with one remote device: however, 100 or more devices could be connected on the same interfaces on distribution router 1 and distribution router 2. The remote device will use the best route to reach its destination. If distribution router 1 experiences a failure, the remote device can still use distribution router 2 to reach the corporate network.

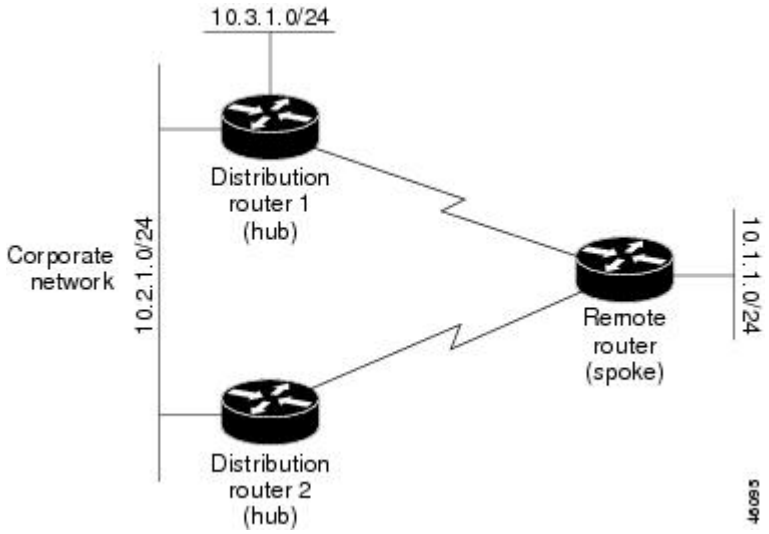
Figure 4: Simple Dual-Homed Remote Topology



The figure above shows a simple dual-homed remote topology with one remote device and two distribution devices. Both distribution devices maintain routes to the corporate network and stub network 10.1.1.0/24.

Dual-homed routing can introduce instability into an EIGRP network. In the figure below, distribution router 1 is directly connected to network 10.3.1.0/24. If summarization or filtering is applied on distribution router 1, the device will advertise network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution router 2 and the remote device).

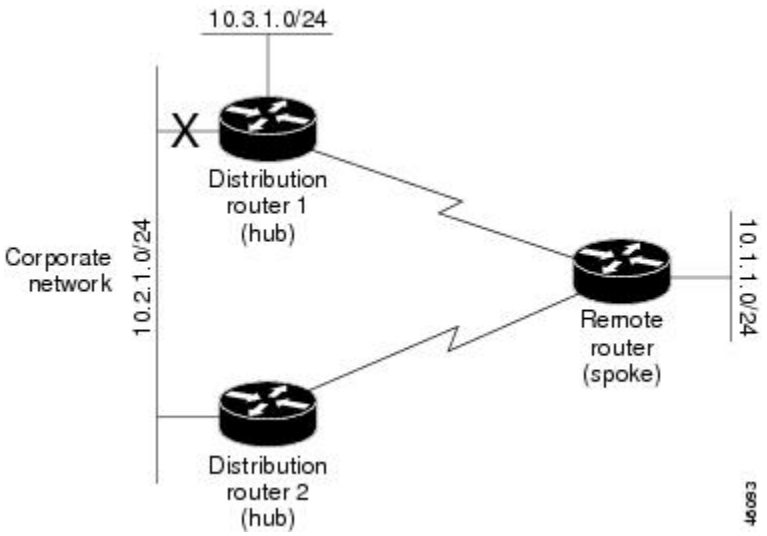
Figure 5: Dual-Homed Remote Topology with Distribution Router 1 Connected to Two Networks



The figure above shows a simple dual-homed remote topology, where distribution router 1 is connected to both network 10.3.1.0/24 and network 10.2.1.0/24.

If the 10.2.1.0/24 link between distribution router 1 and distribution router 2 fails, the lowest cost path to network 10.3.1.0/24 from distribution router 2 will be through the remote device (see the figure below). This route is not desirable because the traffic that was previously traveling across the corporate network 10.2.1.0/24 would now be sent across a much lower bandwidth connection. The overutilization of the lower bandwidth WAN connection can cause many problems that might affect the entire corporate network. The use of the lower bandwidth route that passes through the remote device may cause WAN EIGRP distribution devices to be dropped. Serial lines on distribution and remote devices may also be dropped, and EIGRP SIA errors on the distribution and core devices can occur.

Figure 6: Dual-Homed Remote Topology with a Failed Route to a Distribution Device



It is not desirable for traffic from distribution router 2 to travel through any remote device to reach network 10.3.1.0/24. Backup routes can be used if links are sized to manage the load. However, most networks, of the type shown in the figure above, have remote devices located at remote offices with relatively slow links. To ensure that traffic from distribution devices are not routed through a remote device, you can configure route summarization on the distribution device and the remote device.

It is typically undesirable for traffic from a distribution device to use a remote device as a transit path. A typical connection from a distribution device to a remote device would have much less bandwidth than a connection at the network core. Attempting to use a remote device with a limited bandwidth connection as a transit path would generally produce excessive congestion at the remote device. The EIGRP stub routing feature can prevent this problem by preventing the remote device from advertising core routes back to the distribution devices. In the above example, routes learned by the remote device from distribution router 1 will not be advertised to distribution router 2. Therefore, distribution router 2 will not use the remote device as a transit for traffic destined to the network core.

The EIGRP stub routing feature provides network stability. If the network is not stable, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit devices. Instead, distribution devices to which the stub device is connected answer queries on behalf of the stub device. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is

enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote devices to prevent those devices from appearing as transit paths to hub devices.

Caution The EIGRP stub routing feature should be used only on stub devices. A stub device is defined as a device connected to the network core or distribution layer through which core transit traffic should not flow. A stub device should not have any EIGRP neighbors other than distribution devices. Ignoring this restriction will cause undesirable behavior.

Note Multiaccess interfaces such as ATM, Gigabit Ethernet, Frame Relay, ISDN PRI, and X.25 are supported by the EIGRP stub routing feature only when all devices on that interface, except the hub, are configured as stub devices.

EIGRP Stub Routing Leak Map Support

In EIGRP stub routing configurations where there is a remote site with more than one device, only one of the remote devices can be configured as the stub device. If you have two distribution layer devices and two devices at a remote site, there is no way to declare both remote devices as stub devices. If one remote device is configured as a stub device, the other remote device can neither learn routes towards the network core if the link between the stub device and the distribution layer device fails nor route around the failed link.

The stub device cannot readvertise routes learned from any neighboring EIGRP device. To resolve this issue, a leak map configuration that allows a selected set of learned routes to be readvertised to other peers can be added to the EIGRP stub routing feature. The set of routes allowed through the stub device are specified using a standard route map so that routes can be matched based on tags, prefixes, or interfaces. These routes are marked using the site-of-origin code mechanism, which prevents routes permitted through the stub from being readvertised into the core of the network.

Use the **eigrp stub leak-map** command to configure the EIGRP stub routing feature to reference a leak map that identifies routes that are allowed to be advertised on an EIGRP stub device that would normally have been suppressed.

How to Configure EIGRP

Enabling EIGRP Autonomous System Configuration

Perform this task to enable EIGRP and create an EIGRP routing process. EIGRP sends updates to interfaces in specified networks. If you do not specify the network of an interface, the interface will not be advertised in any EIGRP update.

Configuring the **router eigrp** *autonomous-system-number* command creates an EIGRP autonomous system configuration that creates an EIGRP routing instance, which can be used for tagging routing information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **network** *network-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 1`	Configures an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**network** *network-number* **Example:** `Device(config-router)# network 172.16.0.0`	Associates a network with an EIGRP routing process.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Enabling the EIGRP Named Configuration

Perform this task to enable EIGRP and to create an EIGRP routing process. EIGRP sends updates to interfaces in specified networks. If you do not specify the network of an interface, the interface will not be advertised in any EIGRP update.

Configuring the **router eigrp** *virtual-instance-name* command creates an EIGRP named configuration. The EIGRP named configuration does not create an EIGRP routing instance by itself. The EIGRP named configuration is the base configuration, which is required to define address family configurations used for routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Enabling the EIGRP IPv6 VRF-Lite Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000`	Enables EIGRP IPv6 VRF-Lite and enters address family configuration mode.
Step 5	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring Optional EIGRP Parameters in an Autonomous System Configuration

Perform this task to configure optional EIGRP parameters, which include applying offsets to routing metrics, adjusting EIGRP metrics, and disabling automatic summarization in an EIGRP autonomous system configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **network** *ip-address* [*wildcard-mask*]
5. **passive-interface** [**default**] [*interface-type interface-number*]
6. **offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*]
7. **metric weights** *tos k1 k2 k3 k4 k5*
8. **no auto-summary**
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 1`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**passive-interface** [**default**] [*interface-type interface-number*] **Example:** `Device(config-router)# passive-interface`	(Optional) Suppresses EIGRP hello packets and routing updates on interfaces while still including the interface addresses in the topology database.
Step 6	**offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*] **Example:** `Device(config-router)# offset-list 21 in 10 gigabitethernet 0/0/1`	(Optional) Applies an offset to routing metrics.
Step 7	**metric weights** *tos k1 k2 k3 k4 k5* **Example:** `Device(config-router)# metric weights 0 2 0 2 0 0`	(Optional) Adjusts the EIGRP metric or K value. • EIGRP uses the following formula to determine the total metric to the network: EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))) **Note** If K5 is 0, then (K5/ (Reliability + K4)) is defined as 1.
Step 8	**no auto-summary**	(Optional) Disables automatic summarization.

	Command or Action	Purpose
	Example: `Device(config-router)# no auto-summary`	**Note** Automatic summarization is enabled by default.
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Optional EIGRP Parameters in a Named Configuration

Perform this task to configure optional EIGRP named configuration parameters, which includes applying offsets to routing metrics, adjusting EIGRP metrics, setting the RIB-scaling factor, and disabling automatic summarization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**unicast**] [**vrf** *vrf-name*] [**multicast**] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **metric weights** *tos k1 k2 k3 k4 k5 k6*
7. **af-interface** {**default** | *interface-type interface-number*}
8. **passive-interface**
9. **bandwidth-percent** *maximum-bandwidth-percentage*
10. **exit-af-interface**
11. **topology** {**base** | *topology-name* **tid** *number*}
12. **offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*]
13. **no auto-summary**
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	
	• **address-family ipv4** [**unicast**] [**vrf** *vrf-name*] [**multicast**] **autonomous-system** *autonomous-system-number*
	
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	Example:
	
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 6	**metric weights** *tos k1 k2 k3 k4 k5 k6* **Example:** `Device(config-router-af)# metric weights 0 2 0 2 0 0 0`	(Optional) Adjusts the EIGRP metric or K value. • EIGRP uses the following formula to determine the total 32-bit metric to the network: EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))) • EIGRP uses the following formula to determine the total 64-bit metric to the network:

	Command or Action	Purpose
		EIGRP Metric = 256*((K1*Throughput) + (K2*Throughput)/(256-Load) + (K3*Latency)+ (K6*Extended Attributes))*(K5/(Reliability + K4))) **Note** If K5 is 0, then (K5/ (Reliability + K4)) is defined as 1.
Step 7	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 8	**passive-interface** **Example:** `Device(config-router-af-interface)# passive-interface`	Suppresses EIGRP hello packets and routing updates on interfaces while still including the interface addresses in the topology database.
Step 9	**bandwidth-percent** *maximum-bandwidth-percentage* **Example:** `Device(config-router-af-interface)# bandwidth-percent 75`	Configures the percentage of bandwidth that may be used by an EIGRP address family on an interface.
Step 10	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 11	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 12	**offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*] **Example:** `Device(config-router-af-topology)# offset-list 21 in 10 gigabitethernet 6/2`	(Optional) Applies an offset to routing metrics.
Step 13	**no auto-summary**	(Optional) Disables automatic summarization.

	Command or Action	Purpose
	Example: `Device(config-router-af-topology)# no auto-summary`	**Note** Automatic summarization is enabled by default.
Step 14	**end** **Example:** `Device(config-router-af-topology)# end`	Returns to privileged EXEC mode.

Configuring the EIGRP Redistribution Autonomous System Configuration

Perform this task to configure redistribution of non-EIGRP protocol metrics into EIGRP metrics and to configure the EIGRP administrative distance in an EIGRP autonomous system configuration.

You must use a default metric to redistribute a protocol into EIGRP, unless you use the **redistribute** command.

Note Metric defaults have been carefully set to work for a wide variety of networks. Take great care when changing these values.

Default metrics are supported only when you are redistributing from EIGRP or static routes.

An administrative distance is a rating of the trustworthiness of a routing information source, such as an individual router or a group of routers. Numerically, an administrative distance is an integer from 0 to 255. In general, the higher the value the lower the trust rating. An administrative distance of 255 means the routing information source cannot be trusted at all and should be ignored.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **network** *ip-address* [*wildcard-mask*]
5. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*autonomous-system-number*] [**metric** {*metric-value* | **transparent**}] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**]
6. **distance eigrp** *internal-distance external-distance*
7. **default-metric** *bandwidth delay reliability loading mtu*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 1`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [*autonomous-system-number*] [**metric** {*metric-value* | **transparent**}] [**metric-type** *type-value*] [**match** {**internal** | **external 1** | **external 2**}] [**tag** *tag-value*] [**route-map** *map-tag*] [**subnets**] **Example:** `Device(config-router)# redistribute rip level1`	Redistributes routes from one routing domain into another routing domain.
Step 6	**distance eigrp** *internal-distance external-distance* **Example:** `Device(config-router)# distance eigrp 80 130`	Allows the use of two administrative distances—internal and external.
Step 7	**default-metric** *bandwidth delay reliability loading mtu* **Example:** `Device(config-router)# default-metric 1000 100 250 100 1500`	Sets metrics for EIGRP.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Route Summarization Autonomous System Configuration

Perform this task to configure EIGRP to perform automatic summarization of subnet routes into network-level routes in an EIGRP autonomous system configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **exit**
5. **interface** *type number*
6. **ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*]
7. **ip bandwidth-percent eigrp** *as-number percent*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Enters interface configuration mode.
Step 6	**ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*] **Example:** `Device(config-if)# ip summary-address eigrp 100 10.0.0.0 0.0.0.0`	(Optional) Configures a summary aggregate address.
Step 7	**ip bandwidth-percent eigrp** *as-number percent* **Example:** `Device(config-if)# ip bandwidth-percent eigrp 209 75`	(Optional) Configures the percentage of bandwidth that may be used by EIGRP on an interface.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Route Summarization Named Configuration

Perform this task to configure EIGRP to perform automatic summarization of subnet routes into network-level routes in an EIGRP named configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **af-interface {default |** *interface-type interface-number***}**
6. **summary-address** *ip-address mask* [*administrative-distance* [**leak-map** *leak-map-name*]]
7. **exit-af-interface**
8. **topology** {**base** | *topology-name* **tid** *number*}
9. **summary-metric** *network-address subnet-mask bandwidth delay reliability load mtu*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**af-interface {default |** *interface-type interface-number***}** **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**summary-address** *ip-address mask* [*administrative-distance* [**leak-map** *leak-map-name*]] **Example:** `Device(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0`	Configures a summary address for EIGRP.
Step 7	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 8	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 9	**summary-metric** *network-address subnet-mask bandwidth delay reliability load mtu* **Example:** `Device(config-router-af-topology)# summary-metric 192.168.0.0/16 10000 10 255 1 1500`	(Optional) Configures a fixed metric for an EIGRP summary aggregate address.
Step 10	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Event Logging Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **eigrp event-log-size** *size*
5. **eigrp log-neighbor-changes**
6. **eigrp log-neighbor-warnings** [*seconds*]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**eigrp event-log-size** *size* **Example:** `Device(config-router)# eigrp event-log-size 5000010`	(Optional) Sets the size of the EIGRP event log.
Step 5	**eigrp log-neighbor-changes** **Example:** `Device(config-router)# eigrp log-neighbor-changes`	(Optional) Enables logging of EIGRP neighbor adjacency changes. • By default, the system logs EIGRP neighbor adjacency changes to help you monitor the stability of the routing system and detect problems.

	Command or Action	Purpose
Step 6	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router)# eigrp log-neighbor-warnings 300`	(Optional) Enables the logging of EIGRP neighbor warning messages.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Event Logging Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **eigrp log-neighbor-warnings** [*seconds*]
6. **eigrp log-neighbor-changes**
7. **topology** {**base** | *topology-name* **tid** *number*}
8. **eigrp event-log-size** *size*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	Example:
	
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router-af)# eigrp log-neighbor-warnings 300`	(Optional) Enables the logging of EIGRP neighbor warning messages.
Step 6	**eigrp log-neighbor-changes** **Example:** `Device(config-router-af)# eigrp log-neighbor-changes`	(Optional) Enables logging of EIGRP neighbor adjacency changes. • By default, the system logs EIGRP neighbor adjacency changes to help you monitor the stability of the routing system and detect problems.
Step 7	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.

	Command or Action	Purpose
Step 8	**eigrp event-log-size** *size* **Example:** `Device(config-router-af-topology)# eigrp event-log-size 10000`	(Optional) Sets the size of the EIGRP event log.
Step 9	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring Equal and Unequal Cost Load Balancing Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **traffic-share balanced**
5. **maximum-paths** *number-of-paths*
6. **variance** *multiplier*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**traffic-share balanced** **Example:** `Device(config-router)# traffic-share balanced`	Controls how traffic is distributed among routes when multiple routes for the same destination network have different costs.
Step 5	**maximum-paths** *number-of-paths* **Example:** `Device(config-router)# maximum-paths 5`	Controls the maximum number of parallel routes that an IP routing protocol can support.
Step 6	**variance** *multiplier* **Example:** `Device(config-router)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Equal and Unequal Cost Load Balancing Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **topology** {**base** | *topology-name* **tid** *number*}
6. **traffic-share balanced**
7. **maximum-paths** *number-of-paths*
8. **variance** *multiplier*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 6	**traffic-share balanced** **Example:** `Device(config-router-af-topology)# traffic-share balanced`	Controls how traffic is distributed among routes when multiple routes for the same destination network have different costs.
Step 7	**maximum-paths** *number-of-paths* **Example:** `Device(config-router-af-topology)# maximum-paths 5`	Controls the maximum number of parallel routes that an IP routing protocol can support.
Step 8	**variance** *multiplier* **Example:** `Device(config-router-af-topology)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 9	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Defining an Autonomous System for EIGRP Route Authentication

Before You Begin

Before you configure EIGRP route authentication, you must enable EIGRP. In this task, EIGRP is defined with an autonomous system number.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip authentication mode eigrp** *autonomous-system* **md5**
5. **ip authentication key-chain eigrp** *autonomous-system key-chain*
6. **exit**
7. **key chain** *name-of-chain*
8. **key** *key-id*
9. **key-string** *text*
10. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
11. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/1`	Configures an interface type and enters interface configuration mode.
Step 4	**ip authentication mode eigrp** *autonomous-system* **md5** **Example:** `Device(config-if)# ip authentication mode eigrp 1 md5`	Enables MD5 authentication in EIGRP packets.
Step 5	**ip authentication key-chain eigrp** *autonomous-system key-chain*	Enables authentication of EIGRP packets.

	Command or Action	Purpose
	Example: `Device(config-if)# ip authentication key-chain eigrp 1 keychain1`	
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 7	**key chain** *name-of-chain* **Example:** `Device(config)# key chain keychain1`	Identifies a key chain and enters key chain configuration mode.
Step 8	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies the key number and enters key chain key configuration mode.
Step 9	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string 0987654321`	Identifies the key string.
Step 10	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite`	(Optional) Specifies the time period during which the key can be received.
Step 11	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite`	(Optional) Specifies the time period during which the key can be sent.
Step 12	**end** **Example:** `Device(config-keychain-key)# end`	Exits key chain key configuration mode and returns to privileged EXEC mode.

Defining a Named Configuration for EIGRP Route Authentication

Before You Begin

Before you configure EIGRP route authentication, you must enable EIGRP. In this task, EIGRP is defined with a virtual instance name.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **af-interface** {**default** | *interface-type interface-number*}
7. **authentication key-chain** *name-of-chain*
8. **authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**}
9. **exit-af-interface**
10. **exit-address-family**
11. **exit**
12. **key chain** *name-of-chain*
13. **key** *key-id*
14. **key-string** *text*
15. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
16. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	Example:
	
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 6	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface ethernet 0/0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 7	**authentication key-chain** *name-of-chain* **Example:** `Device(config-router-af-interface)# authentication key-chain SITE1`	Specifies an authentication key chain for EIGRP.

	Command or Action	Purpose
Step 8	**authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**} **Example:** `Device(config-router-af-interface)# authentication mode md5`	Specifies the type of authentication used in an EIGRP address family for the EIGRP instance.
Step 9	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 10	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 12	**key chain** *name-of-chain* **Example:** `Device(config)# key chain keychain1`	Identifies a key chain and enters key chain configuration mode.
Step 13	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies the key number and enters key chain key configuration mode.
Step 14	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string 0987654321`	Identifies the key string.
Step 15	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite`	(Optional) Specifies the time period during which the key can be received.

	Command or Action	Purpose
Step 16	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite`	(Optional) Specifies the time period during which the key can be sent.
Step 17	**end** **Example:** `Device(config-keychain-key)# end`	Exits key chain key configuration mode and returns to privileged EXEC mode.

Adjusting the Interval Between Hello Packets and the Hold Time in an Autonomous System Configuration

Note Cisco recommends not to adjust the hold time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **exit**
5. **interface** *type number*
6. **ip hello-interval eigrp** *autonomous-system-number seconds*
7. **ip hold-time eigrp** *autonomous-system-number seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**exit** **Example:** `Device(config-router)# exit`	Exits to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/1`	Enters interface configuration mode.
Step 6	**ip hello-interval eigrp** *autonomous-system-number seconds* **Example:** `Device(config-if)# ip hello-interval eigrp 109 10`	Configures the hello interval for an EIGRP routing process.
Step 7	**ip hold-time eigrp** *autonomous-system-number seconds* **Example:** `Device(config-if)# ip hold-time eigrp 109 40`	Configures the hold time for an EIGRP routing process. **Note** Do not adjust the hold time without consulting your technical support personnel.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Adjusting the Interval Between Hello Packets and the Hold Time in a Named Configuration

Note Do not adjust the hold time without consulting your technical support personnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **af-interface** {**default** | *interface-type interface-number*}
6. **hello-interval** *seconds*
7. **hold-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

	Command or Action	Purpose

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**hello-interval** *seconds* **Example:** `Device(config-router-af-interface)# hello-interval 10`	Configures the hello interval for an EIGRP address family named configuration.
Step 7	**hold-time** *seconds* **Example:** `Device(config-router-af-interface)# hold-time 50`	Configures the hold time for an EIGRP address family named configuration.
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Disabling the Split Horizon Autonomous System Configuration

Split horizon controls the sending of EIGRP updates and query packets. When split horizon is enabled on an interface, updates and query packets are not sent for destinations for which this interface is the next hop. Controlling updates and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip split-horizon eigrp** *autonomous-system-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1`	Configures an interface and enters interface configuration mode.
Step 4	**no ip split-horizon eigrp** *autonomous-system-number* **Example:** `Device(config-if)# no ip split-horizon eigrp 101`	Disables split horizon.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling the Split Horizon and Next-Hop-Self Named Configuration

EIGRP, by default, sets the next-hop value to the local outbound interface address for routes that it is advertising, even when advertising those routes back from the same interface from where they were learned. Perform this

task to change this default setting and configure EIGRP to use the received next-hop value when advertising these routes. Disabling next-hop-self is primarily useful in DMVPN spoke-to-spoke topologies.

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **af-interface** {**default** | *interface-type interface-number*}
6. **no split-horizon**
7. **no next-hop-self [no-ecmp-mode]**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* • **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**no split-horizon** **Example:** `Device(config-router-af-interface)# no split-horizon`	Disables EIGRP split horizon.
Step 7	**no next-hop-self [no-ecmp-mode]** **Example:** `Device(config-router-af-interface)# no next-hop-self no-ecmp-mode`	(Optional) Instructs an EIGRP router to use the received next hop rather than the local outbound interface address as the next hop. • The **no-ecmp-mode** keyword is an enhancement to the **no next-hop-self** command. When this optional keyword is enabled, all paths to a network in the EIGRP table are evaluated to check whether routes advertised from an interface were learned on the same interface.
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Stub Routing Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **network** *ip-address* [**wildcard-mask**]
5. **eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**]
6. **end**
7. **show ip eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 1`	Configures a remote or distribution device to run an EIGRP process and enters router configuration mode.
Step 4	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.
Step 5	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Device(config-router)# eigrp stub connected static`	Configures a remote device as an EIGRP stub device.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	**show ip eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**] **Example:** `Device# show ip eigrp neighbors detail`	(Optional) Verifies that a remote device has been configured as a stub device with EIGRP. • Enter this command on the distribution device. The last line of the output displays the stub status of the remote or spoke device.

Configuring the EIGRP Stub Routing Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [**wildcard-mask**]
6. **eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**]
7. **exit-address-family**
8. **end**
9. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] [**neighbors**] [**static**] [**detail**] [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	Example:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.
Step 6	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Device(config-router-af) eigrp stub leak-map map1`	Configures a device as a stub using EIGRP.
Step 7	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 9	**show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] [**neighbors**] [**static**] [**detail**] [*interface-type interface-number*] **Example:** `Device# show eigrp address-family ipv4 neighbors detail`	(Optional) Displays neighbors discovered by EIGRP.

Configuring vNET Commands in an EIGRP Autonomous System

Perform this task to configure a vNET trunk interface that connects routers to provide the core interface to transport traffic for multiple virtual networks. Traffic carried over a trunk interface is tagged. This task illustrates how to configure a trunk interface with a base VRF instance and two named VRFs, VRF vrf1 and VRF vrf2, and how to associate a VRF with an EIGRP process.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **vnet tag** *number*
5. **description** *string*
6. **address-family ipv4**
7. **exit-address-family**
8. **exit**
9. **vrf definition** *vrf-name*
10. **vnet tag** *number*
11. **description** *string*
12. **address-family ipv4**
13. **exit-address-family**
14. **exit**
15. **interface** *type number*
16. **ip address** *ip-address mask*
17. **vnet trunk [list** *vrf-list-name***]**
18. **ip hello-interval eigrp** *as-number seconds*
19. **exit**
20. **router eigrp** *autonomous-system-number*
21. **address-family ipv4 [unicast] vrf** *vrf-name* **[autonomous-system** *autonomous-system-number***]**
22. **exit-address-family**
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Configures a VRF routing table instance and enters VRF configuration mode.
Step 4	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 100`	Specifies the global, numeric tag for the VRF. • The same tag number must be configured for the same virtual network on each edge and trunk interface.
Step 5	**description** *string* **Example:** `Router(config-vrf)# description guest access`	(Optional) Describes a VRF to help the network administrator identify the configuration file.
Step 6	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 9	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf2`	Configures a VRF routing table instance and enters VRF configuration mode.
Step 10	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 200`	Specifies the global, numeric tag for the VRF. • The same tag number must be configured for the same VRF on each edge and trunk interface.

	Command or Action	Purpose
Step 11	**description** *string* **Example:** `Router(config-vrf)# description finance`	(Optional) Describes a VRF to help the network administrator identify the configuration file.
Step 12	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 13	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 14	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 15	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 16	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.0.0.0`	Sets a primary address for the interface.
Step 17	**vnet trunk [list** *vrf-list-name***]** **Example:** `Router(config-if)# vnet trunk`	Configures vNET on an interface and creates a vNET subinterface. • By default, all VRFs defined with the **vrf definition** command run on all trunk interfaces on the router. Therefore, VRF vrf1 and VRF vrf2 are now running on this interface. • Use the **list** *vrf-list-name* keyword-argument pair to restrict VRFs running on this trunk interface.

	Command or Action	Purpose
Step 18	**ip hello-interval eigrp** *as-number seconds* **Example:** `Router(config-if)# ip hello-interval eigrp 1 121`	Configures an EIGRP IPv4 hello interval on the interface.
Step 19	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 20	**router eigrp** *autonomous-system-number* **Example:** `Router(config)# router eigrp 1`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP processes can be configured.
Step 21	**address-family ipv4 [unicast] vrf** *vrf-name* **[autonomous-system** *autonomous-system-number***]** **Example:** `Router(config-router)# address-family ipv4 vrf vrf1 autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv4 routing instance.
Step 22	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode and returns to router configuration mode.
Step 23	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring vNET Commands in EIGRP Named Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vrf definition** *vrf-name*
4. **vnet tag** *number*
5. **description** *string*
6. **address-family ipv4**
7. **exit-address-family**
8. **exit**
9. **vrf definition** *vrf-name*
10. **vnet tag** *number*
11. **description** *string*
12. **address-family ipv4**
13. **exit-address-family**
14. **exit**
15. **interface** *type number*
16. **ip address** *ip-address mask*
17. **vnet trunk [list** *vrf-list-name***]**
18. **ip hello-interval eigrp** *as-number seconds*
19. **exit**
20. **router eigrp** *virtual-instance-name*
21. **address-family ipv4 [multicast] [unicast] [vrf** *vrf-name***] autonomous-system** *autonomous-system-number*
22. **af-interface** {**default** | *interface-type interface-number*}
23. **hello-interval** *seconds*
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf1`	Configures a VRF routing table instance and enters VRF configuration mode.
Step 4	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 100`	Specifies the global, numeric tag for the VRF. • The same tag number must be configured for the same virtual network on each edge and trunk interface.
Step 5	**description** *string* **Example:** `Router(config-vrf)# description guest access`	(Optional) Describes a VRF to help the network administrator identify the configuration file.
Step 6	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 8	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 9	**vrf definition** *vrf-name* **Example:** `Router(config)# vrf definition vrf2`	Configures a VRF routing table instance and enters VRF configuration mode.
Step 10	**vnet tag** *number* **Example:** `Router(config-vrf)# vnet tag 200`	Specifies the global, numeric tag for the VRF. • The same tag number must be configured for the same VRF on each edge and trunk interface.

	Command or Action	Purpose
Step 11	**description** *string* **Example:** `Router(config-vrf)# description finance`	(Optional) Describes a VRF to help the network administrator identify the configuration file.
Step 12	**address-family ipv4** **Example:** `Router(config-vrf)# address-family ipv4`	Enters VRF address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 13	**exit-address-family** **Example:** `Router(config-vrf-af)# exit-address-family`	Exits VRF address family configuration mode and returns to VRF configuration mode.
Step 14	**exit** **Example:** `Router(config-vrf)# exit`	Exits VRF configuration mode and returns to global configuration mode.
Step 15	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 16	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 10.1.1.1 255.0.0.0`	Sets a primary address for the interface.
Step 17	**vnet trunk [list** *vrf-list-name***]** **Example:** `Router(config-if)# vnet trunk`	Configures vNET on an interface and creates a vNET subinterface. • By default, all VRFs defined with the **vrf definition** command run on all trunk interfaces on the router. Therefore, VRF vrf1 and VRF vrf2 are now running on this interface. • Use the **list** *vrf-list-name* keyword-argument pair to restrict VRFs running on this trunk interface.

	Command or Action	Purpose
Step 18	**ip hello-interval eigrp** *as-number seconds* **Example:** `Router(config-if)# ip hello-interval eigrp 1 121`	Configures an EIGRP IPv4 hello interval on the interface.
Step 19	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 20	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp vnet`	Enables an EIGRP routing process and enters router configuration mode.
Step 21	**address-family ipv4 [multicast] [unicast] [vrf** *vrf-name***] autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 unicast autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv4 routing instance.
Step 22	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Router(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 23	**hello-interval** *seconds* **Example:** `Router(config-router-af-interface)# hello-interval 121`	Configures the hello interval for an EIGRP address family named configuration.
Step 24	**end** **Example:** `Router(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining the EIGRP Autonomous System Configuration

This task is optional. Use the commands in any order desired to monitor and maintain EIGRP autonomous system configuration.

SUMMARY STEPS

1. **enable**
2. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **accounting**
3. **show ip eigrp events** [*starting-event-number ending-event-number*] [**type**]
4. **show ip eigrp interfaces** [**vrf** {*vrf-name*| *****}] [*autonomous-system-number*] [*type number*] [**detail**]
5. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]
6. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]
7. **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **traffic**

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

Step 2 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **accounting**
Displays prefix accounting information for EIGRP processes.

Example:

```
Device# show ip eigrp vrf VRF1 accounting
```

Step 3 **show ip eigrp events** [*starting-event-number ending-event-number*] [**type**]
Displays information about interfaces that are configured for EIGRP.

Example:

```
Device# show ip eigrp events
```

Step 4 **show ip eigrp interfaces** [**vrf** {*vrf-name*| *****}] [*autonomous-system-number*] [*type number*] [**detail**]
Displays neighbors discovered by EIGRP.

Example:

```
Device# show ip eigrp interfaces
```

Step 5 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]

Displays neighbors discovered by EIGRP

Example:
```
Device# show ip eigrp neighbors
```

Step 6 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **topology** [*ip-address* [*mask*]] | [**name**] [**active** | **all-links** | **detail-links** | **pending** | **summary** | **zero-successors**]
Displays entries in the EIGRP topology table.

Example:
```
Device# show ip eigrp topology
```

Step 7 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **traffic**
Displays the number of EIGRP packets sent and received.

Example:
```
Device# show ip eigrp traffic
```

Monitoring and Maintaining the EIGRP Named Configuration

This task is optional. Use the commands in any order desired to monitor and maintain the EIGRP named configuration.

SUMMARY STEPS

1. **enable**
2. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**
3. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **events** [*starting-event-number ending-event-number*] [**errmsg** [*starting-event-number ending-event-number*]] [**sia** [*starting-event-number ending-event-number*]] [**type**]
4. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **interfaces** [**detail**] [*interface-type interface-number*]
5. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **neighbors** [**static**] [**detail**] [*interface-type interface-number*]
6. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **timers**
7. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **topology** [*topology-name*] [*ip-address*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**route-type** {**connected** | **external** | **internal** | **local** | **redistributed** | **summary** | **vpn**}]
8. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **traffic**
9. **show eigrp plugins** [*plugin-name*] [**detailed**]
10. **show eigrp protocols** [**vrf** *vrf-name*]

DETAILED STEPS

Step 1 **enable**
Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

Step 2 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**
Displays prefix accounting information for EIGRP processes.

Example:

```
Device# show eigrp address-family ipv4 22 accounting
```

Step 3 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **events** [*starting-event-number ending-event-number*] [**errmsg** [*starting-event-number ending-event-number*]] [**sia** [*starting-event-number ending-event-number*]] [**type**]
Displays information about EIGRP address-family events.

Example:

```
Device# show eigrp address-family ipv4 3 events
```

Step 4 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **interfaces** [**detail**] [*interface-type interface-number*]
Displays information about interfaces that are configured for EIGRP.

Example:

```
Device# show eigrp address-family ipv4 4453 interfaces
```

Step 5 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **neighbors** [**static**] [**detail**] [*interface-type interface-number*]
Displays the neighbors that are discovered by EIGRP.

Example:

```
Device# show eigrp address-family ipv4 4453 neighbors
```

Step 6 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **timers**
Displays information about EIGRP timers and expiration times.

Example:

```
Device# show eigrp address-family ipv4 4453 timers
```

Step 7 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **topology** [*topology-name*] [*ip-address*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**route-type** {**connected** | **external** | **internal** | **local** | **redistributed** | **summary** | **vpn**}]

Displays entries in the EIGRP topology table.

Example:

```
Device# show eigrp address-family ipv4 4453 topology
```

Step 8 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **traffic**
Displays the number of EIGRP packets that are sent and received.

Example:

```
Device# show eigrp address-family ipv4 4453 traffic
```

Step 9 **show eigrp plugins** [*plugin-name*] [**detailed**]
Displays general information, including the versions of the EIGRP protocol features that are currently running on the device.

Example:

```
Device# show eigrp plugins
```

Step 10 **show eigrp protocols** [**vrf** *vrf-name*]
Displays further information about EIGRP protocols that are currently running on a device.

Example:

```
Device# show eigrp protocols
```

Configuration Examples for EIGRP

Example: Enabling EIGRP—Autonomous System Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
```

Example: Enabling EIGRP—Named Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
```

Example: Enabling EIGRP IPv6 VRF-Lite—Named Configuration

```
Device> enable
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 100:1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000
```

Example: EIGRP Parameters—Autonomous System Configuration

The following example shows how to configure optional EIGRP autonomous system configuration parameters, including applying offsets to routing metrics, adjusting EIGRP metrics, and disabling automatic summarization:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# passive-interface
Device(config-router)# offset-list 21 in 10 ethernet 0
Device(config-router)# metric weights 0 2 0 2 0 0
Device(config-router)# no auto-summary
Device(config-router)# exit
```

Example: EIGRP Parameters—Named Configuration

The following example shows how to configure optional EIGRP named configuration parameters, including applying offsets to routing metrics, adjusting EIGRP metrics, setting RIB-scaling factor, and disabling automatic summarization.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# metric weights 0 2 0 2 0 0 0
Device(config-router-af)# metric rib-scale 100
Device(config-router-af)# af-interface gigabitethernet 0/0/1
Device(config-router-af-interface)# passive-interface
Device(config-router-af-interface)# bandwidth-percent 75
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af-interface)# topology base
Device(config-router-af-topology)# offset-list 21 in 10 gigabitethernet 0/0/1
Device(config-router-af-topology)# no auto-summary
Device(config-router-af-topology)# exit-af-topology
```

Example: EIGRP Redistribution—Autonomous System Configuration

The following example shows how to configure redistribution of non-EIGRP protocol metrics into EIGRP metrics and configure the EIGRP administrative distance in an EIGRP autonomous system configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.0
Device(config-router)# redistribute rip
Device(config-router)# distance eigrp 80 130
Device(config-router)# default-metric 1000 100 250 100 1500
```

Example: EIGRP Route Summarization—Autonomous System Configuration

The following example shows how to configure route summarization on an interface and configure the automatic summary feature for an EIGRP autonomous system configuration. The following configuration causes EIGRP to summarize the network from Ethernet interface 0/0.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 101
Device(config-router)# exit
Device(config)# interface ethernet 0/0
Device(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0
Device(config-if)# ip bandwidth-percent eigrp 209 75
```

Note You should not use the **ip summary-address eigrp** summarization command to generate the default route (0.0.0.0) from an interface because this creates an EIGRP summary default route to the null 0 interface with an administrative distance of 5. The low administrative distance of this default route can cause this route to displace default routes learned from other neighbors through the routing table. If the default route learned from the neighbors is displaced by the summary default route, or if the summary route is the only default route present, all traffic destined for the default route will not leave the router; instead, traffic will be sent to the null 0 interface, where it is dropped. The recommended way to send only the default route out of a given interface is to use the **distribute-list** command. You can configure this command to filter all outbound route advertisements sent out from the interface with the exception of the default (0.0.0.0).

Example: EIGRP Route Summarization—Named Configuration

The following example shows how to configure route summarization on an interface and configure the automatic summary feature for an EIGRP named configuration. This configuration causes EIGRP to summarize network 192.168.0.0 only from Ethernet interface 0/0.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# topology base
Device(config-router-af-topology)# summary-metric 192.168.0.0/16 10000 10 255 1 1500
```

Example: EIGRP Event Logging—Autonomous System Configuration

The following example shows how to configure EIGRP event logging parameters, including setting the size of the EIGRP event log for an EIGRP autonomous system configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# eigrp event-log-size 5000
Device(config-router)# eigrp log-neighbor-changes
Device(config-router)# eigrp log-neighbor-warnings 300
```

Example: EIGRP Event Logging—Named Configuration

The following example shows how to configure EIGRP event logging parameters, including setting the size of the EIGRP event log for an EIGRP named configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# eigrp log-neighbor-warnings 300
Device(config-router-af)# eigrp log-neighbor-changes
Device(config-router-af)# topology base
Device(config-router-af-topology)# eigrp event-log-size 10000
```

Example: Equal and Unequal Cost Load Balancing—Autonomous System Configuration

The following example shows how to configure traffic distribution among routes, the maximum number of parallel routes, and load balancing in an EIGRP named configuration network:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# traffic-share balanced
Device(config-router)# maximum-paths 5
Device(config-router)# variance 1
```

Example: Equal and Unequal Cost Load Balancing—Named Configuration

The following example shows how to configure traffic distribution among routes, the maximum number of parallel routes, and load balancing in an EIGRP named configuration network:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# topology base
Device(config-router-af-topology)# traffic-share balanced
Device(config-router-af-topology)# maximum-paths 5
Device(config-router-af-topology)# variance 1
```

Example: EIGRP Route Authentication—Autonomous System Definition

The following example shows how to enable MD5 authentication on EIGRP packets in autonomous system 1.

Router A will accept and attempt to verify the MD5 digest of any EIGRP packet with a key equal to 1. It will also accept a packet with a key equal to 2. All other MD5 packets will be dropped. Router A will send all EIGRP packets with key 2.

Router B will accept key 1 or key 2 and will use key 1 to send MD5 authentication because key 1 is the first valid key of the key chain. Key 1 is not valid after December 4, 2006. After this date, key 2 is used to send MD5 authentication, and this key is valid until January 4, 2007.

The figure below shows the scenario.

Router A Configuration

```
Device> enable
Device(config)# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface ethernet 1/0
Device(config-if)# ip authentication mode eigrp 1 md5
Device(config-if)# ip authentication key-chain eigrp 1 key1
Device(config-if)# exit
Device(config)# key chain key1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 04:48:00 Dec 4 1996
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string 1234567890
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Router B Configuration

```
Device> enable
Device(config)# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface ethernet 1/0
Device(config-if)# ip authentication mode eigrp 1 md5
Device(config-if)# ip authentication key-chain eigrp 1 key2
Device(config-if)# exit
Device(config)# key chain key2
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string 1234567890
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Example: EIGRP Route Authentication—Named Configuration

The following example shows how to enable MD5 authentication on EIGRP packets in a named configuration.

Router A will accept and attempt to verify the MD5 digest of any EIGRP packet with a key equal to 1. It will also accept a packet with a key equal to 2. All other MD5 packets will be dropped. Router A will send all EIGRP packets with key 2.

Router B will accept key 1 or key 2 and will use key 1 to send MD5 authentication because key 1 is the first valid key of the key chain. Key 1 is not valid after December 4, 2006. After this date, key 2 will be used to send MD5 authentication because it is valid until January 4, 2007.

Router A Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# authentication key-chain SITE1
Device(config-router-af-interface)# authentication mode md5
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# exit-address-family
Device(config-router)# exit
Device(config)# key chain SITE1
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string 1234567890
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Router B Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name2
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.0
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# authentication key-chain SITE2
Device(config-router-af-interface)# authentication mode md5
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# exit-address-family
Device(config-router)# exit
Device(config)# key chain SITE2
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string 0987654321
Device(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Device(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
```

The following example shows how to configure advanced SHA authentication with password password1 and several key strings that will be rotated as time passes:

```
!
key chain chain1
 key 1
  key-string securetraffic
  accept-lifetime 04:00:00 Dec 4 2006 infinite
```

```
  send-lifetime 04:00:00 Dec 4 2010 04:48:00 Dec 4 2008
 !
 key 2
  key-string newertraffic
  accept-lifetime 01:00:00 Dec 4 2010 infinite
  send-lifetime 03:00:00 Dec 4 2010 infinite
 exit
!
router eigrp virtual-name
  address-family ipv6 autonomous-system 4453
    af-interface ethernet 0
       authentication mode hmac-sha-256 0 password1
       authentication key-chain key1
  !
!
```

Example: Adjusting the Interval Between Hello Packets and the Hold Time—Autonomous System Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Ethernet 0/1
Device(config-if)# ip hello-interval eigrp 109 10
Device(config-if)# ip hold-time eigrp 109 40
```

Example: Adjusting the Interval Between Hello Packets and the Hold Time—Named Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# hello-interval 10
Device(config-router-af-interface)# hold-time 50
```

Example: Disabling the Split Horizon—Autonomous System Configuration

Split horizon is enabled on all interfaces by default. The following example shows how to disable split horizon for an EIGRP autonomous system configuration:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Ethernet 0/1
Device(config-if)# no ip split-horizon eigrp 101
```

Example: Disabling the Split Horizon and Next-Hop-Self—Named Configuration

Split horizon is enabled on all interfaces by default. The following example shows how to disable split horizon in an EIGRP named configuration.

EIGRP, by default, sets the next-hop value to the local outbound interface address for routes that it advertises, even when advertising those routes back out of the same interface from where they were learned. The following example shows how to change this default to instruct EIGRP to use the received next-hop value when advertising these routes in an EIGRP named configuration. Disabling the **next-hop-self** command is primarily useful in DMVPN spoke-to-spoke topologies.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# no split-horizon
Device(config-router-af-interface)# no next-hop-self no-ecmp-mode
```

Example: EIGRP Stub Routing—Autonomous System Configuration

A device that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor devices by default. The following six keywords can be used with the **eigrp stub** command to modify this behavior:

- **connected**
- **leak-map**
- **receive-only**
- **redistributed**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command for an EIGRP autonomous system configuration.

Example: eigrp stub Command

In the following example, the **eigrp stub** command is used to configure the device as a stub that advertises connected and summary routes:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub
```

Example: eigrp stub connected static Command

In the following example, the **eigrp stub** command is used with the **connected** and **static** keywords to configure the device as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub connected static
```

Example: eigrp stub leak-map Command

In the following example, the **eigrp stub** command is issued with the **leak-map** *name* keyword-argument pair to configure the device to reference a leak map that identifies routes that would have been suppressed:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub leak-map map1
```

Example: eigrp stub receive-only Command

In the following example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the device as a receive-only neighbor (connected, summary, and static routes will not be sent):

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub receive-only
```

Example: eigrp stub redistributed Command

In the following example, the **eigrp stub** command is issued with the **redistributed** keyword to configure the device to advertise other protocols and autonomous systems:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub redistributed
```

Example: EIGRP Stub Routing—Named Configuration

A device that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor devices by default. The following six keywords can be used with the **eigrp stub** command to modify this behavior:

- **connected**
- **leak-map**
- **receive-only**
- **redistributed**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command for an EIGRP named configuration.

Example: eigrp stub Command

In the following example, the **eigrp stub** command is used to configure the device as a stub that advertises connected and summary routes:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af) eigrp stub
```

Example: eigrp stub connected static Command

In the following named configuration example, the **eigrp stub** command is issued with the **connected** and **static** keywords to configure the device as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub connected static
```

Example: eigrp stub leak-map Command

In the following named configuration example, the **eigrp stub** command is issued with the **leak-map** *name* keyword-argument pair to configure the device to reference a leak map that identifies routes that would normally have been suppressed:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub leak-map map1
```

Example: eigrp stub receive-only Command

In the following named configuration example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the device as a receive-only neighbor (connected, summary, and static routes will not be sent):

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub receive-only
```

Example: eigrp stub redistributed Command

In the following named configuration example, the **eigrp stub** command is issued with the **redistributed** keyword to configure the device to advertise other protocols and autonomous systems:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af) eigrp stub redistributed
```

Example: Command Inheritance and Virtual Network Interface Mode Override in an EIGRP Environment

Suppose a GigabitEthernet interface is configured with the following EIGRP commands:

```
interface gigabitethernet 0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 end
```

Because a trunk is configured, a VRF subinterface is automatically created and the commands on the main interface are inherited by the VRF subinterface (g0/0/0.3, where the number 3 is the tag number from vnet tag 3.)

Use the **show derived-config** command to display the hidden subinterface. The following sample output shows that all the commands entered on GigabitEthernet 0/0/0 have been inherited by GigabitEthernet 0/0/0.3:

```
Router# show derived-config interface gigabitethernet 0/0/0.3

Building configuration...
Derived configuration : 478 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET vrf1
 vrf forwarding vrf1
 encapsulation dot1Q 3
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 end
```

Use the virtual network interface mode to override the commands entered in interface configuration mode. For example:

```
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# vnet name vrf1
Router(config-if-vnet)# no ip authentication mode eigrp 1 md5
! disable authen for e0/0.3 only
Router(config-if-vnet)# ip authentication key-chain eigrp 1 y
! different key-chain
Router(config-if-vnet)# ip band eigrp 1 99
! higher bandwidth-percent
Router(config-if-vnet)# no ip dampening-change eigrp 1
! disable dampening-change
Router(config-if-vnet)# ip hello eigrp 1 7
Router(config-if-vnet)# ip hold eigrp 1 21
Router(config-if-vnet)# ip next-hop-self eigrp 1
! enable next-hop-self for e0/0.3
Router(config-if-vnet)# ip split-horizon eigrp 1
! enable split-horizon

Router(config-if-vnet)# do show running-config interface gigabitethernet 0/0/0
```

```
Building configuration...
Current configuration : 731 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
  vnet name vrf1
  ip split-horizon eigrp 1
  no ip authentication mode eigrp 1 md5
  ip authentication key-chain eigrp 1 y
  ip bandwidth-percent eigrp 1 99
  no ip dampening-change eigrp 1
  ip hello-interval eigrp 1 7
  ip hold-time eigrp 1 21
  !
end
```

Notice that g/0/0.3 is now using the override settings:

```
Router(config-if-vnet)# do show derived-config interface gigabitethernet 0/0.3

Building configuration...
Derived configuration : 479 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET vrf1
 vrf forwarding vrf1
 encapsulation dot1Q 3
 ip address 192.0.2.1 255.255.255.0
 no ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 y
 ip bandwidth-percent eigrp 1 99
 no ip dampening-change eigrp 1
 ip hello-interval eigrp 1 7
 ip hold-time eigrp 1 21
 ip next-hop-self eigrp 1
 ip split-horizon eigrp 1
 end
```

Commands entered in virtual network interface mode are sticky. That is, when you enter a command in this mode, the command will override the default value configured in interface configuration mode.

The following example shows how to change the default hello interval value in vrf 1. The example also shows sample outputs of the current and derived configurations.

```
Router(config)# interface gigabitethernet 0/0/0
Router(config-if)# ip address 192.0.2.1 255.255.255.0
Router(config-if)# vnet trunk
Router(config-if)# ip hello eigrp 1 7
Router(config-if)# do show run interface gigabitethernet 0/0/2

Building configuration...
Current configuration : 134 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
 ipv6 enable
 vnet global
 !
 end
```

```
Router(config-if)# do show derived interface gigabitethernet 0/0/0.3

Building configuration...

Derived configuration : 177 bytes
!
interface Ethernet0/0.3
 description Subinterface for VNET vrf1
 encapsulation dot1Q 3
 vrf forwarding vrf1
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
end

Router(config-if)# vnet name vrf1
Router(config-if-vnet)# ip hello-interval eigrp 1 10
Router(config-if-vnet)# do show run interface gigabitethernet 0/0/0

Building configuration...
Current configuration : 183 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
 ipv6 enable
 vnet name vrf1
  ip hello-interval eigrp 1 10
 !
 vnet global
 !
end

Router(config-if-vnet)# do show derived interface gigabitethernet 0/0/0.3

Building configuration...

Derived configuration : 178 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET vrf1
 encapsulation dot1Q 3
 vrf forwarding vrf1
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 10
end
```

Because of this sticky factor, to remove a configuration entry in virtual network interface mode, use the default form of that command. Some commands can also be removed using the **no** form.

```
R1(config-if-vnet)# default ip authentication mode eigrp 1 md5
R1(config-if-vnet)# no ip bandwidth-percent eigrp 1
R1(config-if-vnet)# no ip hello eigrp 1

R1(config-if-vnet)# do show running-config interface gigabitethernet 0/0/0

Building configuration...
Current configuration : 138 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 no ip address
 vnet name vrf1
 !
end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
CEF commands	*Cisco IOS IP Switching Command Reference*
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
NSF with SSO deployment	Cisco Nonstop Forwarding with Stateful Switchover Deployment Guide

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 4724	*Graceful Restart Mechanism for BGP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for EIGRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

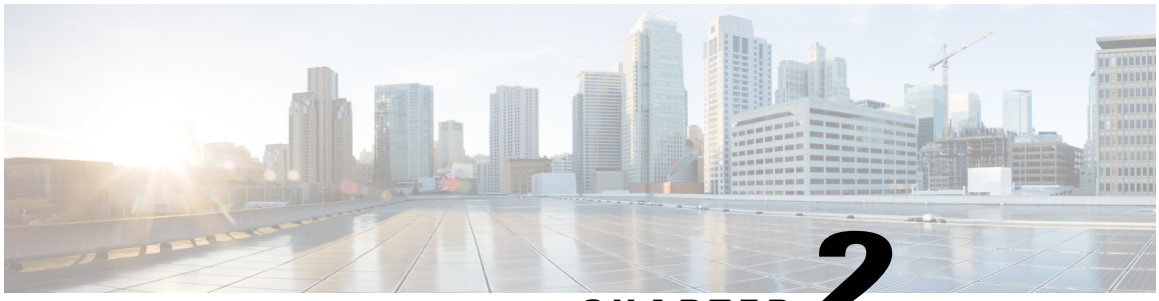
Table 3: Feature Information for Configuring EIGRP

Feature Name	Releases	Feature Information
EIGRP	Cisco IOS XE Release 2.1 15.2(1)E	

Feature Name	Releases	Feature Information
		EIGRP is an enhanced version of the IGRP developed by Cisco. EIGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP. The following commands were introduced or modified: **address-family (EIGRP)**, **af-interface**, **autonomous-system (EIGRP)**, **auto-summary (EIGRP)**, **bandwidth percent**, **clear eigrp address-family neighbors**, **clear ip eigrp neighbors**, **debug eigrp address-family neighbor**, **debug eigrp address-family notifications**, **default-information**, **default-metric (EIGRP)**, **distance (EIGRP)**, **eigrp event-log-size**, **eigrp log-neighbor-changes**, **eigrp log-neighbor-warnings**, **eigrp router-id**,**exit-address-family**, **exit-af-interface**, **exit-af-topology**, **hello-interval**, **hold-time**, **ip bandwidth-percent eigrp**, **ip hello-interval eigrp**, **ip hold-time eigrp**, **ip**, **ip next-hop-self eigrp**, **ip split-horizon eigrp** , **ip summary-address eigrp**, **match extcommunity**, **metric maximum-hops**, **metric weights (EIGRP)**,

Feature Name	Releases	Feature Information
		neighbor (EIGRP), **network (EIGRP)**, **next-hop-self**, **offset-list (EIGRP)** , **passive-interface (EIGRP)**, **router eigrp**, **set metric (EIGRP)**, **show eigrp address-family accounting**, **show eigrp address-family events**, **show eigrp address-family interfaces**, **show eigrp address-family neighbors**, **show eigrp address-family timers**, **show eigrp address-family topology**, **show eigrp address-family traffic**, **show eigrp plugins**, **show eigrp protocols**, **show eigrp tech-support**, **show ip eigrp accounting**, **show ip eigrp events**, **show ip eigrp interfaces**, **show ip eigrp neighbors**, **show ip eigrp topology**, **show ip eigrp traffic**, **show ip eigrp vrf accounting**, **show ip eigrp vrf interfaces**, **show ip eigrp vrf neighbors**, **show ip eigrp vrf topology**, **show ip eigrp vrf traffic**, **shutdown (address-family)**, **split-horizon (EIGRP)**, **summary-address (EIGRP)**, **summary-metric timers active-time**, **timers active-time**, **traffic-share balanced** , **variance (EIGRP)**.

Feature Name	Releases	Feature Information
EIGRP Dual DMVPN Domain Enhancement	Cisco IOS XE Release 3.5S	The EIGRP Dual DMVPN Domain Enhancement feature supports the no next-hop-self functionality on dual DMVPN domains in both IPv4 and IPv6 configurations. In Cisco IOS XE Release 3.9S, support was added for the Cisco CSR 1000V routers. The following commands were introduced or modified by this feature: **ip next-hop-self eigrp**, **ipv6 next-hop self eigrp**,**next-hop-self**, **show ip eigrp interfaces**, **show ipv6 eigrp interfaces**, **show ip eigrp topology**, **show ipv6 eigrp topology**.
Named Mode for EIGRP vNETS IPv4	Cisco IOS XE Release 3.5S Cisco IOS XE Release 3.3SG	The EIGRP vNET feature allows the creation of multiple virtual networks by utilizing a single set of routers and links provided by the physical topology. EIGRP vNET configurations are supported in both classic and named modes. In Cisco IOS Release 15.1(1)SG, EIGRP vNET configurations are supported only in the classic mode. In Cisco IOS XE Release 3.9S, support was added for the Cisco CSR 1000V routers. The following command was modified: **vnet**.



CHAPTER 2

IPv6 Routing: EIGRP Support

Customers can configure Enhanced Interior Gateway Routing Protocol (EIGRP) to route IPv6 prefixes. EIGRP IPv4 runs over an IPv4 transport, communicates only with IPv4 peers, and advertises only IPv4 routes, and EIGRP for IPv6 follows the same model. EIGRP for IPv4 and EIGRP for IPv6 are configured and managed separately. However, the configuration of EIGRP for IPv4 and IPv6 is similar and provides operational familiarity and continuity.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for IPv6 Routing EIGRP Support

This section lists ways in which EIGRP for IPv6 differs from EIGRP IPv4 and lists EIGRP for IPv6 restrictions:

- EIGRP for IPv6 is directly configured on the interfaces over which it runs. This feature allows EIGRP for IPv6 to be configured without the use of a global IPv6 address. There is no network statement in EIGRP for IPv6.

In per-interface configuration at system startup, if EIGRP has been configured on an interface, then the EIGRP protocol may start running before any EIGRP router mode commands have been executed.

- An EIGRP for IPv6 protocol instance requires a router ID before it can start running.
- EIGRP for IPv6 has a shutdown feature. The routing process should be in "no shut" mode in order to start running.
- EIGRP for IPv6 provides route filtering using the **distribute-list prefix-list** command. Use of the **route-map** command is not supported for route filtering with a distribute list.

Information About IPv6 Routing EIGRP Support

Cisco EIGRP for IPv6 Implementation

EIGRP is an enhanced version of the IGRP developed by Cisco. EIGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm called the diffusing update algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize at the same time. Devices that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

EIGRP provides the following features:

- Increased network width--With Routing Information Protocol (RIP), the largest possible width of your network is 15 hops. When EIGRP is enabled, the largest possible width is 224 hops. Because the EIGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this limitation by incrementing the transport control field only when an IPv4 or an IPv6 packet has traversed 15 devices and the next hop to the destination was learned by way of EIGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.
- Fast convergence--The DUAL algorithm allows routing information to converge as quickly as any other routing protocol.
- Partial updates--EIGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for EIGRP packets.
- Neighbor discovery mechanism--This is a simple hello mechanism used to learn about neighboring devices. It is protocol-independent.
- Arbitrary route summarization.
- Scaling--EIGRP scales to large networks.
- Route filtering--EIGRP for IPv6 provides route filtering using the **distribute-list prefix-list**command. Use of the **route-map** command is not supported for route filtering with a distribute list.

EIGRP has the following four basic components:

- Neighbor discovery--Neighbor discovery is the process that devices use to dynamically learn of other devices on their directly attached networks. Devices must also discover when their neighbors become unreachable or inoperative. EIGRP neighbor discovery is achieved with low overhead by periodically sending small hello packets. EIGRP neighbors can also discover a neighbor that has recovered after an outage because the recovered neighbor will send out a hello packet. As long as hello packets are received, the Cisco software can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring devices can exchange routing information.
- Reliable transport protocol--The reliable transport protocol is responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some EIGRP packets must be sent reliably and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities, it is not necessary to send hello packets reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when unacknowledged packets are pending. This provision helps to ensure that convergence time remains low in the presence of varying speed links.
- DUAL finite state machine--The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses several metrics including distance and cost information to select efficient, loop-free paths. When multiple routes to a neighbor exist, DUAL determines which route has the lowest metric (named the feasible distance), and enters this route into the routing table. Other possible routes to this neighbor with larger metrics are received, and DUAL determines the reported distance to this network. The reported distance is defined as the total metric advertised by an upstream neighbor for a path to a destination. DUAL compares the reported distance with the feasible distance, and if the reported distance is less than the feasible distance, DUAL considers the route to be a feasible successor and enters the route into the topology table. The feasible successor route that is reported with the lowest metric becomes the successor route to the current route if the current route fails. To avoid routing loops, DUAL ensures that the reported distance is always less than the feasible distance for a neighbor device to reach the destination network; otherwise, the route to the neighbor may loop back through the local device.
- Protocol-dependent modules--When there are no feasible successors to a route that has failed, but there are neighbors advertising the route, a recomputation must occur. This is the process in which DUAL determines a new successor. The amount of time required to recompute the route affects the convergence time. Recomputation is processor-intensive; it is advantageous to avoid unneeded recomputation. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, DUAL will use them in order to avoid unnecessary recomputation.

The protocol-dependent modules are responsible for network layer protocol-specific tasks. For example, the EIGRP module is responsible for sending and receiving EIGRP packets that are encapsulated in IPv4 or IPv6. It is also responsible for parsing EIGRP packets and informing DUAL of the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IPv4 or IPv6 routing table. Also, EIGRP is responsible for redistributing routes learned by other IPv4 or IPv6 routing protocols.

How to Configure IPv6 Routing EIGRP Support

Enabling EIGRP for IPv6 on an Interface

EIGRP for IPv6 is directly configured on the interfaces over which it runs, which allows EIGRP for IPv6 to be configured without the use of a global IPv6 address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **no shut**
6. **ipv6 enable**
7. **ipv6 eigrp** *as-number*
8. **ipv6 router eigrp** *as-number*
9. **router-id** *ip-address*
10. **exit**
11. **show ipv6 eigrp** [*as-number*] **interfaces** [*type number*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is to be configured.
Step 5	**no shut** **Example:** `Device(config-if)# no shut`	Enables no shut mode so the routing process can start running.
Step 6	**ipv6 enable** **Example:** `Device(config-if)# ipv6 enable`	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.
Step 7	**ipv6 eigrp** *as-number* **Example:** `Device(config-if)# ipv6 eigrp 1`	Enables EIGRP for IPv6 on a specified interface.
Step 8	**ipv6 router eigrp** *as-number* **Example:** `Device(config-if)# ipv6 router eigrp 1`	Enters router configuration mode and creates an EIGRP IPv6 routing process.
Step 9	**router-id** *ip-address* **Example:** `Device(config-router)# router-id 10.1.1.1`	Enables the use of a fixed router ID. Use this command only if an IPv4 address is not defined on the router eligible for router ID.
Step 10	**exit** **Example:** `Device(config-router)# exit`	Enter three times to return to privileged EXEC mode.
Step 11	**show ipv6 eigrp** [*as-number*] **interfaces** [*type number*] [**detail**] **Example:** `Device# show ipv6 eigrp interfaces`	Displays information about interfaces configured for EIGRP for IPv6 .

Configuring the Percentage of Link Bandwidth Used by EIGRP

By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth, as configured with the **bandwidth interface** command. You might want to change that value if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 bandwidth-percent eigrp** *as-number percent*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 bandwidth-percent eigrp** *as-number percent* **Example:** `Device(config-if)# ipv6 bandwidth-percent eigrp 1 75`	Configures the percentage of bandwidth that may be used by EIGRP for IPv6 on an interface

Configuring Summary Addresses

This task configures a summary address for a specified interface. If other specific routes are in the routing table, EIGRP for IPv6 will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 summary-address eigrp** *as-number ipv6-address* [*admin-distance*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shut`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 summary-address eigrp** *as-number ipv6-address* [*admin-distance*] **Example:** `Device(config-if)# ipv6 summary-address eigrp 1 2001:DB8:0:1::/64`	Configures a summary aggregate address for a specified interface.

Configuring EIGRP Route Authentication

EIGRP route authentication provides message digest algorithm 5 (MD5) authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

Each key has its own key identifier, which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and MD5 authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in order from lowest to highest, and uses the first valid key it encounters. Note that the device needs to know the time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 authentication mode eigrp** *as-number* **md5**
6. **ipv6 authentication key-chain eigrp** *as-number key-chain*
7. **exit**
8. **key chain** *name-of-chain*
9. **key** *key-id*
10. **key-string** *text*
11. **accept-lifetime** *start-time* **infinite** | *end-time*| **duration** *seconds*
12. **send-lifetime** *start-time* **infinite** | *end-time* | **duration** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shutdown`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 authentication mode eigrp** *as-number* **md5** **Example:** `Device(config-if)# ipv6 authentication mode eigrp 1 md5`	Specifies the type of authentication used in EIGRP for IPv6 packets.
Step 6	**ipv6 authentication key-chain eigrp** *as-number key-chain* **Example:** `Device(config-if)# ipv6 authentication key-chain eigrp 1 chain1`	Enables authentication of EIGRP for IPv6 packets.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	**key chain** *name-of-chain* **Example:** `Device(config)# key chain chain1`	Identifies a group of authentication keys. • Use the name specified in Step 5.
Step 9	**key** *key-id* **Example:** `Device(config-keychain)# key 1`	Identifies an authentication key on a key chain.
Step 10	**key-string** *text* **Example:** `Device(config-keychain-key)# key-string chain 1`	Specifies the authentication string for a key.

	Command or Action	Purpose
Step 11	**accept-lifetime** *start-time* **infinite** | *end-time*| **duration** *seconds* **Example:** `Device(config-keychain-key)# accept-lifetime 14:30:00 Jan 10 2006 duration 7200`	Sets the time period during which the authentication key on a key chain is received as valid.
Step 12	**send-lifetime** *start-time* **infinite** | *end-time* | **duration** *seconds* **Example:** `Device(config-keychain-key)# send-lifetime 15:00:00 Jan 10 2006 duration 3600`	Sets the time period during which an authentication key on a key chain is valid to be sent.

Overriding the Next Hop in EIGRP

EIGRP will, by default, set the IPv6 next-hop value to be itself for routes that it is advertising, even when advertising those routes back out the same interface where it learned them. Perform this task to change this default and instruct EIGRP to use the received next-hop value when advertising these routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **no ipv6 next-hop-self eigrp** *as-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shutdown`	Enables no shut mode so the routing process can start running.
Step 5	**no ipv6 next-hop-self eigrp** *as-number* **Example:** `Device(config-if)# no ipv6 next-hop-self eigrp 1`	Changes the default IPv6 next-hop value and instructs EIGRP to use the received next-hop value.

Adjusting the Interval Between Hello Packets in EIGRP for IPv6

Routing devices periodically send hello packets to each other to dynamically learn of other devices on their directly attached networks. This information is used to discover neighbors and learn when neighbors become unreachable or inoperative.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 hello-interval eigrp** *as-number seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shutdown`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 hello-interval eigrp** *as-number seconds* **Example:** `Device(config)# ipv6 hello-interval eigrp 1 10`	Configures the hello interval for the EIGRP for IPv6 routing process designated by an autonomous system number.

Adjusting the Hold Time in EIGRP for IPv6

On very congested and large networks, the default hold time might not be sufficient time for all devices to receive hello packets from their neighbors. In this case, you may want to increase the hold time.

This task configures the hold time on a specified interface for a particular EIGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is 3 times the hello interval, or 15 seconds. For slow-speed nonbroadcast multi-access (NBMA) networks, the default hold time is 180 seconds. The hold time should be changed if the hello-interval value is changed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **ipv6 hold-time eigrp** *as-number seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shutdown`	Enables no shut mode so the routing process can start running.
Step 5	**ipv6 hold-time eigrp** *as-number seconds* **Example:** `Device(config)# ipv6 hold-time eigrp 1 40`	Configures the hold time for a particular EIGRP for IPv6 routing process designated by the autonomous system number.

Disabling Split Horizon in EIGRP for IPv6

By default, split horizon is enabled on all interfaces. Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

Split horizon blocks route information from being advertised by a device out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks (such as multipoint GRE), situations can arise for which this behavior is not ideal. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no shut**
5. **no ipv6 split-horizon eigrp** *as-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface FastEthernet 0/0`	Specifies the interface on which EIGRP is configured.
Step 4	**no shut** **Example:** `Device(config)# no shutdown`	Enables no shut mode so the routing process can start running.
Step 5	**no ipv6 split-horizon eigrp** *as-number* **Example:** `Device(config-if)# no ipv6 split-horizon eigrp 101`	Disables EIGRP for IPv6 split horizon on the specified interface.

Configuring EIGRP Stub Routing for Greater Network Stability

The EIGRP stub routing feature can help to provide greater network stability. In the event of network instability, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit devices. Instead, distribution devices to which the stub device is connected answer the query on behalf of the stub

device. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote devices to prevent those remote devices from appearing as transit paths to the hub devices.

Caution EIGRP stub routing should be used only on stub devices. A stub device is defined as a device connected to the network core or distribution layer through which core transit traffic should not flow. A stub device should not have any EIGRP neighbors other than distribution devices.

Configuring a Device for EIGRP Stub Routing

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **eigrp stub receive-only** | **leak-map** | **connected** | **static** | **summary** | **redistributed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**eigrp stub receive-only** | **leak-map** | **connected** | **static** | **summary** | **redistributed** **Example:** `Device(config-router)# eigrp stub`	Configures a device as a stub using EIGRP.

Verifying EIGRP Stub Routing

SUMMARY STEPS

1. **enable**
2. **show ipv6 eigrp neighbors detail** *interface-type* | *as-number* | **static**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 eigrp neighbors detail** *interface-type* | *as-number* | **static** **Example:** `Device# show ipv6 eigrp neighbors detail`	Displays the neighbors discovered by EIGRP for IPv6. This command is performed on the distribution layer device to view the status of the remote device.

Customizing an EIGRP for IPv6 Routing Process

Logging EIGRP Neighbor Adjacency Changes

You can enable the logging of neighbor adjacency changes to monitor the stability of the routing system and to help you detect problems. By default, adjacency changes are logged.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **eigrp log-neighbor-changes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**eigrp log-neighbor-changes** **Example:** `Device(config-router)# eigrp log-neighbor-changes`	Enables the logging of changes in EIGRP for IPv6 neighbor adjacencies.

Configuring Intervals Between Neighbor Warnings

When neighbor warning messages occur, they are logged by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **eigrp log-neighbor-warnings** [*seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router)# eigrp log-neighbor-warnings 300`	Configures the logging intervals of EIGRP neighbor warning messages.

Adjusting EIGRP for IPv6 Metric Weights

EIGRP for IPv6 uses the minimum bandwidth on the path to a destination network and the total delay to compute routing metrics. You can use the **metric weights** command to adjust the default behavior of EIGRP for IPv6 routing and metric computations. EIGRP for IPv6 metric defaults have been carefully selected to provide optimal performance in most networks.

Note Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default values without guidance from an experienced network designer.

By default, the EIGRP composite metric is a 32-bit quantity that is a sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (e.g., GigabitEthernet, FastEthernet, Ethernet), the route with the lowest metric reflects the most desirable path to a destination.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router eigrp** *as-number*
4. **metric weights** *tos k1 k2 k3 k4 k5*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 router eigrp** *as-number* **Example:** `Device(config)# ipv6 router eigrp 1`	Specifies the EIGRP for IPv6 routing process to be configured.
Step 4	**metric weights** *tos k1 k2 k3 k4 k5* **Example:** `Device(config-router)# metric weights 0 2 0 2 0 0`	Tunes EIGRP metric calculations.

Deleting Entries from EIGRP for IPv6 Routing Tables

SUMMARY STEPS

1. **enable**
2. **clear ipv6 eigrp** [*as-number*] [**neighbor** [*ipv6-address* | *interface-type interface-number*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ipv6 eigrp** [*as-number*] [**neighbor** [*ipv6-address* | *interface-type interface-number*]] **Example:** `Device# clear ipv6 eigrp neighbor 3FEE:12E1:2AC1:EA32`	Deletes entries from EIGRP for IPv6 routing tables. The routes that are cleared are the routes that were learned by the specified device.

Configuration Examples for IPv6 Routing EIGRP Support

Example: Configuring EIGRP to Establish Adjacencies on an Interface

EIGRP for IPv6 is configured directly on the interfaces over which it runs. This example shows the minimal configuration required for EIGRP for IPv6 to send hello packets in order to establish adjacencies on Ethernet 0/0:

```
ipv6 unicast-routing
interface ethernet0/0
no shut
  ipv6 enable
  ipv6 eigrp 1
!
ipv6 router eigrp 1
  eigrp router-id 10.1.1.1
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
IPv6 commands	Cisco IOS IPv6 Command Reference
Cisco IOS IPv6 features	Cisco IOS IPv6 Feature Mapping
Configuring EIGRP	*EIGRP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFCs*

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for IPv6 Routing: EIGRP Support

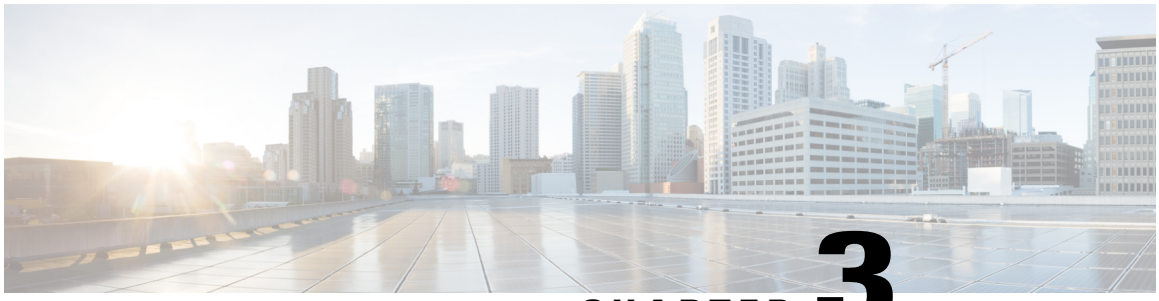
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for IPv6 Routing: EIGRP Support

Feature Name	Releases	Feature Information
IPv6 Routing: EIGRP Support	15.2(1)E	Customers can configure EIGRP to route IPv6 prefixes. There is no linkage between EIGRP for IPv4 and EIGRP for IPv6; they are configured and managed separately. However, the configuration of EIGRP for IPv4 and IPv6 is similar and provides operational familiarity and continuity. The following commands were introduced or modified: **accept-lifetime**, **clear ipv6 eigrp**, **eigrp log-neighbor-changes**, **eigrp log-neighbor-warnings**, **eigrp router-id**, **eigrp stub**, **ipv6 authentication key-chain eigrp**, **ipv6 authentication mode eigrp**, **ipv6 eigrp**, **ipv6 hello-interval eigrp**, **ipv6 hold-time eigrp**, **ipv6 next-hop-self eigrp**, **ipv6 router eigrp**, **ipv6 split-horizon eigrp**, **ipv6 summary-address eigrp**, **ipv6 unicast-routing**, **key**, **key chain**, **key-string**, **metric weights**, **send-lifetime**, **show ipv6 eigrp**, **show ipv6 eigrp neighbors**.

Feature Name	Releases	Feature Information
EIGRP IPv6 VRF Lite	15.1(1)S	The EIGRP IPv6 VRF Lite feature provides EIGRP IPv6 support for multiple VRFs. EIGRP for IPv6 can operate in the context of a VRF. The EIGRP IPv6 VRF Lite feature provides separation between routing and forwarding, providing an additional level of security because no communication between devices belonging to different VRFs is allowed unless it is explicitly configured. The EIGRP IPv6 VRF Lite feature simplifies the management and troubleshooting of traffic belonging to a specific VRF. The EIGRP IPv6 VRF Lite feature is available only in EIGRP named configurations. There are no new or modified commands for this feature.



CHAPTER 3

EIGRP MIB

The EIGRP MIB feature provides complete Enhanced Interior Gateway Routing Protocol (EIGRP) support for GET requests and limited notification (also known as trap) support for neighbor authentication failure, neighbor down, and stuck-in-active (SIA) events. This MIB is accessed through remote Simple Network Management Protocol (SNMP) software clients. The EIGRP IPv6 MIB feature enables IPv6 support for the EIGRP MIB.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for EIGRP MIB

- An Enhanced Interior Gateway Routing Protocol (EIGRP) routing process must be enabled and a Simple Network Management Protocol (SNMP) community string must be configured on at least one device for EIGRP MIB table objects to be visible via SNMP.

- Support for EIGRP notifications (traps) is not activated until a trap destination is configured.

Restrictions for EIGRP MIB

EIGRP MIB support was not implemented for the EIGRP Prefix Limit Support feature.

Information About EIGRP MIB

EIGRP MIB Overview

The EIGRP MIB feature provides MIB support in Cisco software for Enhanced Interior Gateway Routing Protocol (EIGRP) routing processes that run over IPv4 and IPv6. The EIGRP MIB is accessed through remote Simple Network Management Protocol (SNMP) software clients. MIB table objects are accessed as read-only through GETBULK, GETINFO, GETMANY, GETONE, and GETNEXT requests. Counters for MIB table objects are cleared when the EIGRP routing process is reset or when the routing table is refreshed when you enter the **clear ip route** or **clear ip eigrp** command. Managed objects for all EIGRP routing processes are implemented as five table objects—EIGRP Interface, EIGRP Neighbor, EIGRP Topology, EIGRP Traffic Statistics, and EIGRP VPN—on a per-autonomous-system or per-VPN basis.

EIGRP Interface Table

The EIGRP Interface table contains information and statistics for all interfaces on which the Enhanced Interior Gateway Routing Protocol (EIGRP) has been configured. The objects in this table are populated on a per-interface basis. The table below describes EIGRP Interface table objects and the values populated for each object.

Table 5: EIGRP Interface Table Object Descriptions

EIGRP Interface Table Object	Description
cEigrpAcksSuppressed	Total number of individual acknowledgment packets that have been suppressed and combined in an already enqueued outbound reliable packet on an interface.
cEigrpAuthKeyChain	The name of the authentication key chain that is configured on the interface. The key chain is a reference to the set of secret keys that need to be accessed to determine the key string that needs to be used.
cEigrpAuthMode	The authentication mode that is configured for traffic that uses the interface. A value of 0 is displayed when no authentication is enabled. A value of 1 is displayed when message digest algorithm 5 (MD5) authentication is enabled.

EIGRP Interface Table Object	Description
cEigrpCRpkts	Total number conditional receive (CR) packets sent from the interface.
cEigrpHelloInterval	The configured time interval (in seconds) between hello packet transmissions on the interface.
cEigrpPacingReliable	The configured time interval (in milliseconds) between EIGRP packet transmissions on the interface when the reliable transport is used.
cEigrpPacingUnreliable	The configured time interval (in milliseconds) between EIGRP packet transmissions on the interface when the unreliable transport is used.
cEigrpPeerCount	Total number of neighbor adjacencies formed through the interface.
cEigrpPendingRoutes	Total number of routing updates that are queued for transmission on the interface.
cEigrpMcastExcept	Total number of EIGRP multicast exception transmissions that have occurred on the interface.
cEigrpMeanSrtt	The computed smooth round-trip time (SRTT) for packets that were transmitted to and received from all neighbors on the interface.
cEigrpMFlowTimer	The configured multicast flow control timer value (in milliseconds) for the interface.
cEigrpOOSrcvd	Total number of out-of-sequence packets received on the interface.
cEigrpRetranSent	Total number of packet retransmissions sent from the interface.
cEigrpRMcasts	Total number of reliable (acknowledgment required) multicast packets that were transmitted on the interface.
cEigrpRUcasts	Total number of reliable (acknowledgment required) unicast packets that were transmitted on the interface.
cEigrpUMcasts	Total number of unreliable (no acknowledgment required) multicast packets that were transmitted on the interface.

EIGRP Interface Table Object	Description
cEigrpUUcasts	Total number of unreliable (no acknowledgment required) unicast packets that were transmitted on the interface.
cEigrpXmitNextSerial	The serial number of the next packet that is queued for transmission on the interface.
cEigrpXmitReliableQ	Total number of packets waiting in the reliable transport transmission queue (acknowledgment required).
cEigrpXmitUnreliableQ	Total number of packets waiting in the unreliable transport transmission queue (no acknowledgment required).

EIGRP Neighbor Table

The EIGRP Neighbor table contains information about Enhanced Interior Gateway Routing Protocol (EIGRP) neighbors with which adjacencies have been established. EIGRP uses a "Hello" protocol to form neighbor relationships with directly connected EIGRP neighbors. The objects in this table are populated on a per-neighbor basis. The table below describes EIGRP Neighbor table objects and the values populated for each object.

Table 6: EIGRP Neighbor Table Object Descriptions

EIGRP Neighbor Table Object	Description
cEigrpHoldTime	The hold timer value for an adjacency with a neighbor. If this timer expires, the neighbor is declared down and removed from the neighbor table.
cEigrpLastSeq	The number of the last sequence of a packet transmitted to a neighbor. This table object value increases as the sequence number increases.
cEigrpPeerAddr	The source IP address of a neighbor that was used to establish an EIGRP adjacency with the local device. The source IP address can be an IPv4 or IPv6 address.
cEigrpPeerAddrType	The protocol type of the remote source IP address that was used by a neighbor to establish an EIGRP adjacency with the local device. The protocol type can be IPv4 or IPv6.
cEigrpPeerIfIndex	The index of the local interface through which a neighbor can be reached.

EIGRP Neighbor Table Object	Description
cEigrpPeerInterface	The name of the local interface through which a neighbor can be reached.
cEigrpPktsEnqueued	Total number of EIGRP packets (all types) currently queued for transmission to a neighbor.
cEigrpRetrans	Cumulative number of packets retransmitted to a neighbor while the neighbor is in an up state.
cEigrpRetries	Total number of times an unacknowledged packet is sent to a neighbor.
cEigrpRto	The computed retransmission timeout (RTO) for a neighbor. The value for this table object is computed as an aggregate average of the time required for packet delivery.
cEigrpSrtt	The computed smooth round-trip time (SRTT) for packets that are transmitted to and received from a neighbor.
cEigrpUpTime	The period for which the EIGRP adjacency to a neighbor has been in an up state. The time period is displayed in hours:minutes:seconds.
cEigrpVersion	EIGRP version information reported by a remote neighbor.

EIGRP Topology Table

The EIGRP Topology table contains information about Enhanced Interior Gateway Routing Protocol (EIGRP) routes that are received in updates and routes that are locally originated. EIGRP sends routing updates to and receives routing updates from adjacent routers with which adjacencies have been formed. The objects in this table are populated on a per-topology table entry (route) basis. The table below describes EIGRP Topology table objects and the values populated for each object.

Table 7: EIGRP Topology Table Object Descriptions

EIGRP Topology Table Object	Description
cEigrpActive	Status of routes in the topology table. The value for this table object is displayed on a per-route basis. A value of 1 is displayed when a route is in active state. A value of 2 is displayed when a route is in passive state (normal).

EIGRP Topology Table Object	Description
cEigrpDestSuccessors	Total number of successors (a successor is a route that is the next hop to a destination network) for a topology table entry. The topology table will contain a successor for each path to a given destination. This table object value increases each time a successor is added.
cEigrpDistance	The computed distance to the destination network entry from the local router.
cEigrpFdistance	The feasible (best) distance to a destination network. This value is used to calculate a feasible successor for a topology table entry.
cEigrpNextHopAddress	The next-hop IP address for a route in a topology table entry. The next hop can be an IPv4 or IPv6 address.
cEigrpNextHopAddressType	The protocol type of the next-hop IP address for a route in a topology table entry. The protocol type can be IPv4 or IPv6.
cEigrpNextHopInterface	The interface through which the next-hop IP address is reached to forward traffic to the destination.
cEigrpReportDistance	The computed distance to the destination network in the topology entry as reported by the originator of the route.
cEigrpRouteOriginAddr	The IP address of the router that originated the route in the topology table entry. This table is populated only if the topology table entry was not locally originated. The route origin address can be an IPv4 or IPv6 address.
cEigrpRouteOriginType	The protocol type of the IP address defined as the origin of the topology route entry. The protocol type can be IPv4 or IPv6.
cEigrpStuckInActive	Stuck-in-active (SIA) status of a route. The value for this table object is displayed on a per-route basis. A value of 1 is displayed when a route is in SIA state (that is, no reply has been received for queries about alternate paths). SIA queries are transmitted when a route is placed in this state.

EIGRP Traffic Statistics Table

The EIGRP Traffic Statistics table contains counters and statistics for specific types of Enhanced Interior Gateway Routing Protocol (EIGRP) packets that are sent and the related, collective information that is generated. Objects in this table are populated on a per-autonomous-system basis. Objects in this table are populated for adjacencies formed on interfaces that have IP addresses configured under EIGRP network statements. The table below describes EIGRP Traffic Statistics table objects and the values populated for each object.

Table 8: EIGRP Traffic Statistics Table Object Descriptions

EIGRP Traffic Statistics Table Object	Description
cEigrpAcksRcvd	Total number of acknowledgment packets that are received in response to the transmitted update packets. This table object value increases as packets are received.
cEigrpAcksSent	Total number of acknowledgment packets that are transmitted in response to received update packets. This table object value increases as packets are transmitted.
cEigrpAsRouterId	The configured or automatically selected router ID in IP address format. This table object is updated if the router ID is manually reconfigured or if the IP address that was automatically selected is removed.
cEigrpAsRouterIdType	The type of IP address that is used as the router ID. The value for this table object is an IPv4 address.
cEigrpInputQDrops	Total number of packets that are dropped from the input queue because the input queue was full. This table object value increases each time a packet is dropped.
cEigrpInputQHighMark	The highest number of packets that have been in the input queue. This table object value increases only when the previous highest number is exceeded.
cEigrpHeadSerial	Internal sequencing number (serial) that is applied to EIGRP topology table routes. Routes are sequenced starting with 1. A value of 0 is displayed when there are no routes in the topology table. The “Head” serial number is applied to the first route in the sequence.
cEigrpHellosRcvd	Total number of received hello packets. This table object value increases as packets are received.

EIGRP Traffic Statistics Table Object	Description
cEigrpHellosSent	Total number of hello packets transmitted. This table object value increases as packets are transmitted.
cEigrpNbrCount	Total number of live neighbors. This table object value increases or decreases as peering sessions are established or expired.
cEigrpNextSerial	Serial number that is applied to the next route in the sequence.
cEigrpQueriesSent	Total number of alternate route query packets that are transmitted. This table object value increases as packets are transmitted.
cEigrpQueriesRcvd	Total number of alternate route query packets that are received. This table object value increases as packets are received.
cEigrpRepliesSent	Total number of reply packets that are transmitted in response to the received query packets. This table object value increases as packets are transmitted.
cEigrpRepliesRcvd	Total number of reply packets that are received in response to transmitted query packets. This table object value increases as packets are received.
cEigrpSiaQueriesSent	Total number of query packets that are sent in response to a destination that is in a stuck-in-active (SIA) state for a down peer. This table object value increases each time an SIA query packet is sent.
cEigrpSiaQueriesRcvd	Total number of SIA query packets that are received from neighbors searching for an alternate path to a destination. This table object value increases each time an SIA query packet is received.
cEigrpTopoRoutes	Total number of EIGRP-derived routes in the topology table. This table object value increases if a route is added.
cEigrpUpdatesRcvd	Total number of routing update packets that are received. This table object value increases as packets are received.
cEigrpUpdatesSent	Total number of routing update packets that are transmitted. This table object value increases as packets are transmitted.

EIGRP Traffic Statistics Table Object	Description
cEigrpXmitDummies	Total number of temporary entries in the topology table. Dummies are internal entries and not transmitted in routing updates.
cEigrpXmitPendReplies	Total number of replies expected in response to locally transmitted query packets. This table object contains a value of 0 until a route is placed in an active state.

EIGRP VPN Table

The EIGRP VPN table contains information about VPNs that are configured to run an Enhanced Interior Gateway Routing Protocol (EIGRP) process. Devices index VPN routes by using the VPN name and the EIGRP autonomous system number. The table below describes the EIGRP VPN table object and the value populated for that object.

Table 9: EIGRP VPN Table Object Description

EIGRP VPN Table Object	Description
cEigrpVpnName	The VPN routing and forwarding (VRF) name. Only VRFs that are configured to run an EIGRP routing process are populated.

EIGRP Notifications

The EIGRP MIB provides limited notification (trap) support for neighbor authentication failure, neighbor down, and stuck-in-active (SIA) events. Use the **snmp-server enable traps eigrp** command to enable Enhanced Interior Gateway Routing Protocol (EIGRP) notifications or traps on a Cisco device. To activate support for trap events, you must configure a trap destination by using the **snmp-server host** command and define a community string by using the **snmp-server community** command. EIGRP notifications are described in the table below.

Table 10: EIGRP Notifications

EIGRP Notifications	Description
cEigrpAuthFailureEvent	When EIGRP message digest algorithm 5 (MD5) authentication is enabled on any interface and neighbor adjacencies are formed, a notification is sent if any adjacency goes down because of an authentication failure. This notification will be sent once per down event. This notification includes the source IP address of the neighbor from which the authentication failure occurred.
cEigrpNbrDownEvent	This notification is sent when a neighbor goes down for any reason, such as hold time expiry, neighbor shutdown, interface shutdown, SIA events, or authentication failure. If a neighbor is down because of an authentication failure, both cEigrpAuthFailureEvent and cEigrpNbrDownEvent notifications are sent.
cEigrpRouteStuckInActive	During the query phase for a new route to a destination network, the route is placed in active state (during which an alternate path is actively sought) and a query packet is broadcast to the network. If no replies are received for the query, SIA query packets are broadcast. If no replies are received for the SIA queries, the neighbor adjacency is dropped, the route is declared to be in an SIA state, and this notification is sent.

How to Enable EIGRP MIB

Enabling EIGRP MIB Notifications

Perform this task to specify a Simple Network Management Protocol (SNMP) server host, configure an SNMP community access string, and enable Enhanced Interior Gateway Routing Protocol (EIGRP) MIB notifications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** {*hostname* | *ip-address*} [**traps** | **informs** | **version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*]
4. **snmp-server community** *string*
5. **snmp-server enable traps** [*notification-type*]
6. **end**
7. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** {*hostname* | *ip-address*} [**traps** | **informs** | **version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] **Example:** `Device(config)# snmp-server host 10.0.0.1 traps version 2c NETMANAGER`	Specifies the destination server host or destination address for SNMP notifications.
Step 4	**snmp-server community** *string* **Example:** `Device(config)# snmp-server community EIGRP1NET1A`	Configures a community access string to permit SNMP access to the local router by the remote SNMP software client. **Note** Cisco software supports both IPv4 and IPv6.
Step 5	**snmp-server enable traps** [*notification-type*] **Example:** `Device(config)# snmp-server enable traps eigrp`	Enables SNMP support for EIGRP notifications. • Notifications can be configured for only neighbor authentication failure, neighbor down, and stuck-in-active (SIA) events.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config | include snmp`	Displays contents of the current running configuration file. • Use the output modifier "|" to display and verify the SNMP configuration.

Configuration Examples for EIGRP MIB

Example: Enabling EIGRP MIB Notifications

The following example shows how to specify a Simple Network Management Protocol (SNMP) server host, configure an SNMP community string, and enable support for Enhanced Interior Gateway Routing Protocol (EIGRP) notifications:

```
Device(config)# snmp-server host 10.0.0.2 traps version 2c NETMANAGER eigrp
Device(config)# snmp-server community EIGRP1NET1A
Device(config)# snmp-server enable traps eigrp
```

The following sample output from the **show running-config** command displays the EIGRP MIB configuration:

```
Device# show running-config | include snmp

snmp-server community EIGRP1NET1A
snmp-server enable traps eigrp
snmp-server host 10.0.0.2 version 2c NETMANAGER eigrp
```

Additional References for EIGRP MIB

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	EIGRP Command Reference
Basic EIGRP configuration tasks	"Configuring EIGRP" module in the *EIGRP Configuration Guide*

Related Topic	Document Title
SNMP commands	SNMP Support Command Reference
SNMP configuration tasks	"Configuring SNMP Support" module in the *SNMP Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 1213	Management Information Base for Network Management of TCP/IP-based Internet: MIB-II

MIBs

MIB	MIBs Link
CISCO-EIGRP-MIB.my	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

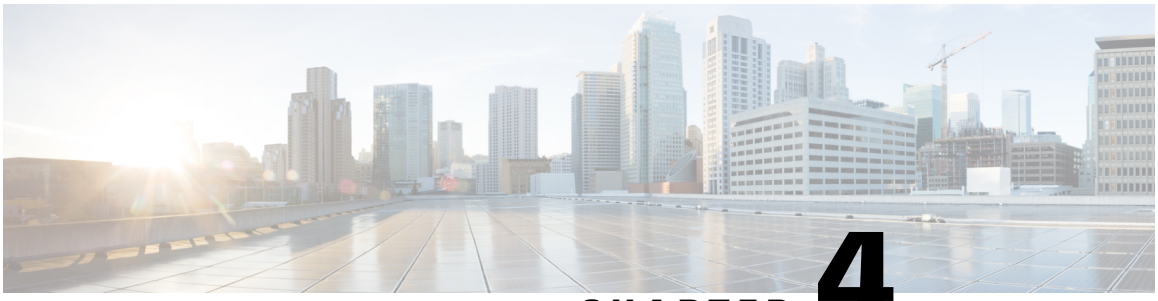
Feature Information for EIGRP MIB

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 11: Feature Information for EIGRP MIB

Feature Name	Releases	Feature Information
EIGRP IPv6 MIB	15.2(4)S 15.3(1)T 15.2(1)E	The EIGRP IPv6 MIB feature enables IPv6 support for the EIGRP MIB. No commands were introduced or modified by this feature.
EIGRP MIB	15.2(1)E	The EIGRP MIB feature provides complete Enhanced Interior Gateway Routing Protocol (EIGRP) support for GET requests and limited notification (trap) support for neighbor authentication failure, neighbor down, and stuck-in-active (SIA) events. This MIB is accessed through remote Simple Network Management Protocol (SNMP) software clients. The following commands were introduced or modified by this feature: **snmp-server enable traps eigrp** and **snmp-server host**.



CHAPTER 4

BFD Support for EIGRP IPv6

The BFD Support for EIGRP IPv6 feature provides Bidirectional Forwarding Detection (BFD) support for Enhanced Interior Gateway Routing Protocol (EIGRP) IPv6 sessions, thereby facilitating rapid fault detection and alternate-path selection in EIGRP IPv6 topologies. BFD is a detection protocol that provides a consistent failure-detection method for network administrators, and network administrators use BFD to detect forwarding path failures at a uniform rate and not at variable rates for different routing protocol 'Hello' mechanisms. This failure-detection methodology ensures easy network profiling and planning and consistent and predictable reconvergence time. This document provides information about BFD support for EIGRP IPv6 networks and explains how to configure BFD support in EIGRP IPv6 networks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for BFD Support for EIGRP IPv6

EIGRP IPv6 sessions have a shutdown option in router, address family, and address-family interface configuration modes. To enable BFD support on EIGRP IPv6 sessions, the routing process should be in no shut mode in the abovementioned modes.

Restrictions for BFD Support for EIGRP IPv6

- The BFD Support for EIGRP IPv6 feature is supported only in EIGRP named mode.
- EIGRP supports only single-hop Bidirectional Forwarding Detection (BFD).
- The BFD Support for EIGRP IPv6 feature is not supported on passive interfaces.

Information About BFD Support for EIGRP IPv6

BFD for EIGRP IPv6

Bidirectional Forwarding Detection (BFD) is a detection protocol that provides fast-forwarding, path-failure detection for all media types, encapsulations, topologies, and routing protocols. The BFD Support for EIGRP IPv6 feature enables BFD to interact with the Enhanced Interior Gateway Routing Protocol (EIGRP) to create BFDv6 sessions between EIGRP neighbors. In a BFD-enabled EIGRP IPv6 session, BFD constantly monitors the forwarding path (from a local device to a neighboring device) and provides consistent failure detection at a uniform rate. Because failure detection happens at a uniform rate and not at variable rates, network profiling and planning is easier, and the reconvergence time remains consistent and predictable.

BFD is implemented in EIGRP at multiple levels; it can be implemented per interface or on all interfaces. When BFD is enabled on a specific interface, all peer relationships formed through the EIGRP “Hello” mechanism on that interface are registered with the BFD process. Subsequently, BFD establishes a session with each of the peers in the EIGRP topology and notifies EIGRP through a callback mechanism of any change in the state of any peer. When a peer is lost, BFD sends a “peer down” notification to EIGRP, and EIGRP unregisters a peer from BFD. BFD does not send a “peer up” notification to EIGRP when the peer is up because BFD now has no knowledge of the state of the peer. This behavior prevents rapid neighbor bouncing and repetitive route computations. The EIGRP “Hello” mechanism will later allow peer rediscovery and reregistration with the BFD process.

How to Configure BFD Support for EIGRP IPv6

Configuring BFD Support on All Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 address** *ipv6-address/prefix-length*
6. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
7. **exit**
8. **router eigrp** *virtual-name*
9. **address-family ipv6 autonomous-system** *as-number*
10. **eigrp router-id** *ip-address*
11. **af-interface default**
12. **bfd**
13. **end**
14. **show eigrp address-family ipv6 neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.

	Command or Action	Purpose
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/1`	Specifies the interface type and number, and enters the interface configuration mode.
Step 5	**ipv6 address** *ipv6-address/prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	Configures an IPv6 address.
Step 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Specifies an EIGRP routing process and enters router configuration mode.
Step 9	**address-family ipv6 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 3`	Enters address family configuration mode for IPv6 and configures an EIGRP routing instance.
Step 10	**eigrp router-id** *ip-address* **Example:** `Device(config-router-af)# eigrp router-id 172.16.1.3`	Sets the device ID used by EIGRP for this address family when EIGRP peers communicate with their neighbors.
Step 11	**af-interface default** **Example:** `Device(config-router-af)# af-interface default`	Configures interface-specific commands on all interfaces that belong to an address family in EIGRP named mode configurations, and enters address-family interface configuration mode.

	Command or Action	Purpose
Step 12	**bfd** **Example:** `Device(config-router-af-interface)# bfd`	Enables BFD on all interfaces.
Step 13	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address-family interface configuration mode and returns to privileged EXEC mode.
Step 14	**show eigrp address-family ipv6 neighbors** **Example:** `Device# show eigrp address-family ipv6 neighbors`	(Optional) Displays neighbors for which BFD has been enabled.

Configuring BFD Support on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 address** *ipv6-address /prefix-length*
6. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*
7. **exit**
8. **router eigrp** *virtual-name*
9. **address-family ipv6 autonomous-system** *as-number*
10. **eigrp router-id** *ip-address*
11. **af-interface** *interface-type interface-number*
12. **bfd**
13. **end**
14. **show eigrp address-family ipv6 neighbors**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/1`	Specifies the interface type and number, and enters the interface configuration mode.
Step 5	**ipv6 address** *ipv6-address /prefix-length* **Example:** `Device(config-if)# ipv6 address 2001:DB8:A:B::1/64`	Configures an IPv6 address.
Step 6	**bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier* **Example:** `Device(config-if)# bfd interval 50 min_rx 50 multiplier 3`	Sets the baseline BFD session parameters on an interface.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 8	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Specifies an EIGRP routing process and enters router configuration mode.

	Command or Action	Purpose
Step 9	**address-family ipv6 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 3`	Enters address family configuration mode for IPv6 and configures an EIGRP routing instance.
Step 10	**eigrp router-id** *ip-address* **Example:** `Device(config-router-af)# eigrp router-id 172.16.1.3`	Sets the device ID used by EIGRP for this address family when EIGRP peers communicate with their neighbors.
Step 11	**af-interface** *interface-type interface-number* **Example:** `Device(config-router-af)# af-interface gigabitethernet0/0/1`	Configures interface-specific commands on an interface that belongs to an address family in an EIGRP named mode configuration, and enters address-family interface configuration mode.
Step 12	**bfd** **Example:** `Device(config-router-af-interface)# bfd`	Enables BFD on the specified interface.
Step 13	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address-family interface configuration mode and returns to privileged EXEC mode.
Step 14	**show eigrp address-family ipv6 neighbors** **Example:** `Device# show eigrp address-family ipv6 neighbors`	(Optional) Displays neighbors for which BFD has been enabled.

Configuration Examples for BFD Support for EIGRP IPv6

Example: Configuring BFD Support on All Interfaces

```
Device(config)# ipv6 unicast-routing
Device(config)# interface GigabitEthernet0/0/1
Device(config-if)# ipv6 address 2001:0DB8:1::12/64
Device(config-if)# bfd interval 50 min_rx 50 multiplier 3
Device(config-if)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 unicast autonomous-system 1
Device(config-router-af)# eigrp router-id 172.16.0.1
```

```
Device(config-router-af)# af-interface default
Device(config-router-af-interface)# bfd
Device(config-router-af-interface)# end
```

Example: Configuring BFD Support on an Interface

```
Device(config)# ipv6 unicast-routing
Device(config)# GigabitEthernet0/0/1
Device(config-if)# ipv6 address 2001:DB8:A:B::1/64
Device(config-if)# bfd interval 50 min_rx 50 multiplier 3
Device(config-if)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous-system 3
Device(config-router-af)# af-interface GigabitEthernet0/0/1
Device(config-router-af-interface)# bfd
Device(config-router-af-interface)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	IP Routing: Protocol-Independent Command Reference
EIGRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples.	IP Routing: EIGRP Command Reference
Configuring EIGRP	"Configuring EIGRP" chapter in *IP Routing: EIGRP Configuration Guide*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

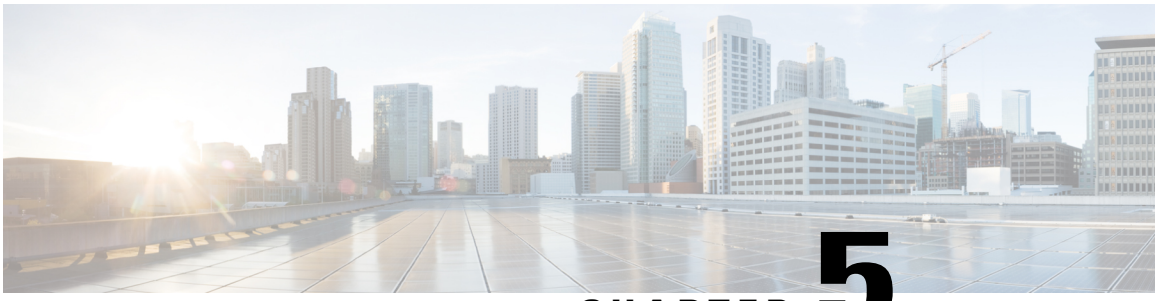
Feature Information for BFD Support for EIGRP IPv6

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 12: Feature Information for BFD Support for EIGRP IPv6

Feature Name	Releases	Feature Information
BFD Support for EIGRP IPv6	15.2(1)E	Bidirectional Forwarding Detection (BFD) is a detection protocol that provides fast-forwarding, path-failure detection for all media types, encapsulations, topologies, and routing protocols. BFD helps network administrators to ensure easier network profiling and planning and consistent and predictable reconvergence time. BFD interacts with Enhanced Interior Gateway Routing Protocol (EIGRP) to create sessions (IPv4 type sessions) between EIGRP neighbors for fast-forwarding, path-failure detections. Each session tests the forwarding path for a single route from a local router to a neighboring router. For any change in state (forwarding path goes down or forwarding path comes up) for any of the sessions, BFD notifies EIGRP of the new state for that route. Support has been added for EIGRP IPv6 neighbors to use BFD as a fall-over mechanism. The following commands were introduced or modified: **bfd**, **show eigrp address-family neighbors**, **show eigrp address-family interfaces**.



CHAPTER 5

EIGRP IPv6 VRF-Lite

The EIGRP IPv6 VRF-Lite feature provides EIGRP IPv6 support for multiple VRFs and simplifies the management and troubleshooting of traffic belonging to a specific VRF.

Note The EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About EIGRP IPv6 VRF-Lite

VRF-Lite for EIGRP IPv6

The EIGRP IPv6 VRF-Lite feature provides separation between routing and forwarding, which supports an additional level of security because communication between devices belonging to different VRFs is not allowed, unless explicitly configured. While the EIGRP IPv6 VRF-Lite feature supports multiple VRFs, the feature also simplifies the management and troubleshooting of traffic belonging to a specific VRF.

Virtual Private Networks (VPNs) provide a secure way for customers to share bandwidth over a service provider backbone network. A VPN is a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, and the service provider associates each interface with a VPN routing table. A VPN routing table is called a VPN routing/forwarding (VRF) table.

VRF-lite allows a service provider to support two or more VPNs with an overlapping IP address using one interface. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.

Note The EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations.

EIGRP Named Configuration

Configuring the **router eigrp** command with the *virtual-instance-name* argument creates an EIGRP configuration referred to as the EIGRP named configuration or EIGRP named mode. An EIGRP named configuration does not create an EIGRP routing instance by itself; it is a base configuration that is required to define address-family configurations that are used for routing.

In EIGRP named configurations, EIGRP VPNs can be configured in IPv4 and IPv6 named configurations. A VRF instance and a route distinguisher must be defined before the address family session can be created.

A single EIGRP routing process can support multiple VRFs. The number of VRFs that can be configured is limited only by the available system resources on the device, which is determined by the number running processes and available memory. However, only a single VRF can be supported by each VPN, and redistribution between different VRFs is not supported.

How to Configure EIGRP IPv6 VRF-Lite

Enabling the EIGRP IPv6 VRF-Lite Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.
Step 4	**address-family ipv6 vrf** *vrf-name* **autonomous-system** *autonomous-system-number* **Example:** `Device(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000`	Enables EIGRP IPv6 VRF-Lite and enters address family configuration mode.
Step 5	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuration Examples for EIGRP IPv6 VRF-Lite

Example: Enabling EIGRP IPv6 VRF-Lite—Named Configuration

The following example shows how to enable the EIGRP IPv6 VRF-lite feature:

```
Device> enable
Device# configure terminal
Device(config)# vrf definition vrf1
Device(config-vrf)# rd 100:1
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
Device(config-vrf)# exit
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000
Device(config-router-af)#
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

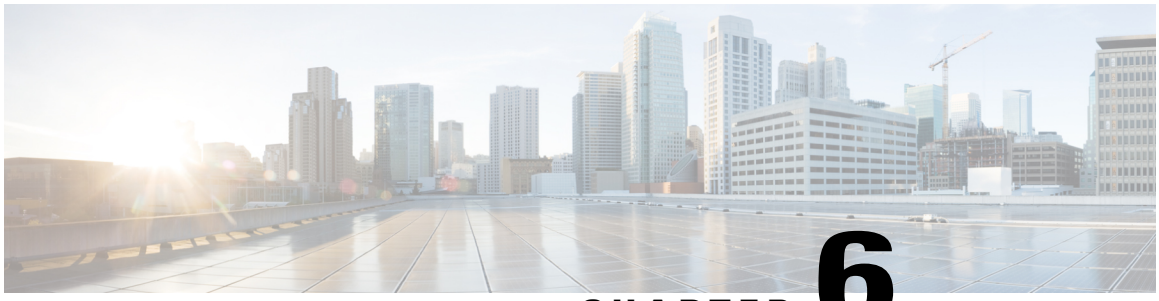
Feature Information for EIGRP IPv6 VRF-Lite

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 13: Feature Information for EIGRP IPv6 VRF-Lite

Feature Name	Releases	Feature Information
EIGRP IPv6 VRF-Lite	15.2(1)E	The EIGRP IPv6 VRF-Lite feature provides EIGRP IPv6 support for multiple VRFs and simplifies the management and troubleshooting of traffic belonging to a specific VRF. **Note** The EIGRP IPv6 VRF-Lite feature is available only in EIGRP named configurations. There are no new or modified commands for this feature.



CHAPTER 6

EIGRP IPv6 NSF/GR

The EIGRP IPv6 NSF/GR feature allows a Nonstop Forwarding (NSF)-aware device that is running Enhanced Interior Gateway Routing Protocol (EIGRP) to forward IPv6 packets while EIGRP restarts after recovering from a failure.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for EIGRP IPv6 NSF/GR

- EIGRP (Enhanced Interior Gateway Routing Protocol) IPv6 must be configured on devices. You need not specify the **network** *network-number* command in EIGRP named mode. By default, EIGRP IPv6 enables EIGRP on all interfaces configured with an IPv6 address.
- Cisco software that supports Nonstop Forwarding (NSF) awareness or NSF capabilities must be installed.

- A redundant facility must be configured to notify EIGRP during a switchover and to notify whether the restart is due to a switchover or a device reboot.
- An NSF-aware device must be up and completely converged with the network before it can assist an NSF-capable device in an NSF restart operation.
- All neighboring devices participating in EIGRP NSF must be NSF-capable or NSF-aware.

Restrictions for EIGRP IPv6 NSF/GR

- Nonstop Forwarding (NSF) is supported on platforms that support high-availability systems.
- An Enhanced Interior Gateway Routing Protocol (EIGRP) NSF-aware network does not allow two neighbors to perform an NSF restart operation at the same time. However, neighbors can re-establish peering sessions after the NSF restart operation is complete.
- NSF for IPv6 is supported only in EIGRP named mode configurations.

Information About EIGRP IPv6 NSF/GR

EIGRP IPv6 NSF/GR

The EIGRP IPv6 NSF/GR feature allows a Nonstop Forwarding (NSF)-aware device that is running Enhanced Interior Gateway Routing Protocol (EIGRP) to forward IPv6 packets along routes that are known to a device that is performing a switchover operation. EIGRP peers retain adjacencies and routes learned from a restarting peer (the device that is undergoing a switchover), and the EIGRP peers continue to forward IPv6 packets to the restarting peer. The high-availability systems on the device retain the forwarding table and continue to forward IPv6 packets until the control plane (EIGRP) has converged on the restarting device.

NSF allows forwarding of IPv6 packets while the device restarts after a failure. Graceful Restart (GR) allows topology databases to resynchronize while maintaining neighbor relationships and forwarding paths.

Note NSF supports IPv4 in EIGRP classic mode and named mode configurations. NSF supports IPv6 in named mode. For more information about the EIGRP IPv4 NSF feature, see the "EIGRP Nonstop Forwarding Awareness" module in the *IP Routing: EIGRP Configuration Guide*.

EIGRP IPv6 NSF Timers

The EIGRP IPv6 NSF/GR feature supports three types of timers: the signal timer, the converge timer, and the graceful-restart purge-time timer.

Configure the signal timer to adjust the maximum time of the initial restart period. The restarting device sends hello packets with the restart-signal (RS) bit set. If the restarting device has not learned about any neighbor or any Nonstop Forwarding (NSF)-aware neighbor or has not received all updates from neighbors when the

timer expires, the Routing Information Base (RIB) is notified for convergence. The default value for the signal timer is 20 seconds. The **timers nsf signal** command is used to configure the signal timer.

Configure the converge timer to adjust the maximum time that a restarting device waits for the end-of-table (EOT) indications from all neighbors. The default value for the converge timer is 120 seconds. The **timers nsf converge** command is used to configure the converge timer.

Configure the graceful-restart purge-time timer to adjust the maximum waiting time to receive the convergent signal from a restarting device. The graceful-restart purge-time timer is used when the NSF-aware peer does not receive the EOT indication from the restarting neighbor. When the graceful-restart purge-time timer expires, the Enhanced Interior Gateway Routing Protocol (EIGRP) peer scans the topology table for stale routes from the restarting neighbor and changes the stale routes to active. This process allows EIGRP peers to find alternate routes instead of waiting during a long switchover operation. The default value for the graceful-restart purge-time timer is 240 seconds. The **timers graceful-restart purge-time** command is used to configure the graceful-restart purge-time timer.

How to Configure EIGRP IPv6 NSF/GR

Enabling EIGRP IPv6 NSF/GR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *name*
4. **address-family ipv6 autonomous-system** *number*
5. **nsf**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**router eigrp** *name* **Example:** `Device(config)# router eigrp e1`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.
Step 4	**address-family ipv6 autonomous-system** *number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv6 routing instance.
Step 5	**nsf** **Example:** `Device(config-router-af)# nsf`	Enables Nonstop Forwarding (NSF) for the specific address family on the device.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Modifying EIGRP IPv6 NSF Timers

Perform this task to modify EIGRP IPv6 NSF timers. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *name*
4. **address-family ipv6 autonomous-system** *number*
5. **timers nsf signal** *seconds*
6. **timers nsf converge** *seconds*
7. **timers graceful-restart purge-time** *seconds*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *name* **Example:** `Device(config)# router eigrp e1`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.
Step 4	**address-family ipv6 autonomous-system** *number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 1`	Enters address family configuration mode to configure an EIGRP IPv6 routing instance.
Step 5	**timers nsf signal** *seconds* **Example:** `Device(config-router-af)# timers nsf signal 15`	Sets the initial restart period, in seconds, for the restarting device to send hello packets with the restart-signal (RS) bit set.
Step 6	**timers nsf converge** *seconds* **Example:** `Device(config-router-af)# timers nsf converge 60`	Sets the maximum time, in seconds, that the restarting device must wait for end-of-table (EOT) indications from all neighbors.
Step 7	**timers graceful-restart purge-time** *seconds* **Example:** `Device(config-router-af)# timers graceful-restart purge-time 150`	Sets the graceful-restart purge-time timer to determine the period, in seconds, for which a Nonstop Forwarding (NSF)-aware device that is running EIGRP must hold routes for an inactive peer.
Step 8	**end** **Example:** `Device(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Verifying the EIGRP IPv6 NSF/GR Configuration

SUMMARY STEPS

1. **enable**
2. **show ipv6 protocols**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ipv6 protocols** **Example:** `Device# show ipv6 protocols`	Displays parameters and the current state of the active IPv6 routing protocol process. • The output of this command can be used to verify the EIGRP IPv6 NSF/GR configuration.

Monitoring EIGRP IPv6 NSF/GR Events

SUMMARY STEPS

1. **enable**
2. **debug eigrp nsf**
3. **debug eigrp address-family ipv6 notifications**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**debug eigrp nsf** **Example:** `Device# debug eigrp nsf`	Displays debugging information about NSF events on the console of the router.
Step 3	**debug eigrp address-family ipv6 notifications** **Example:** `Device# debug eigrp address-family ipv6 notifications`	Displays debugging information about Enhanced Interior Gateway Routing Protocol (EIGRP) address family IPv6 event notifications.

Configuration Examples for EIGRP IPv6 NSF/GR

Example: Configuring an EIGRP NSF Converge Timer

The following example shows how to adjust the maximum time that the restarting router waits for end-of-table (EOT) indications from all neighbors:

```
Device> enable
Device# configure terminal
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous system 1
Device(config-router-af)# timers nsf converge 60
Device(config-router-af)# end
```

Example: Verifying the Configuration of EIGRP IPv6 NSF/GR on an NSF-Aware Device

The following is a sample output from the **show ipv6 protocols** command, which shows that EIGRP NSF is enabled, the graceful-restart purge-time timer is set to 260 seconds, the signal timer is set to 15 seconds, and the converge timer is set to 65 seconds:

```
Device> enable
Device# show ipv6 protocols

IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "bgp 1"
  IGP synchronization is disabled
  Redistribution:
    None
IPv6 Routing Protocol is "bgp multicast"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 1"
EIGRP-IPv6 VR(name) Address-Family Protocol for AS(1)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0 K6=0
  Metric rib-scale 128
  Metric version 64bit
```

```
NSF-aware route hold timer is 260
EIGRP NSF enabled
   NSF signal timer is 15s
   NSF converge timer is 65s
Router-ID: 10.1.1.1
Topology : 0 (base)
  Active Timer: 3 min
  Distance: internal 90 external 170
  Maximum path: 16
  Maximum hopcount 100
  Maximum metric variance 1
  Total Prefix Count: 0
  Total Redist Count: 0

Interfaces:
Redistribution:
  None
```

Additional References for EIGRP IPv6 NSF/GR

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco Express Forwarding (formerly known as CEF) commands	Cisco IOS IP Switching Command Reference
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
Nonstop Forwarding (NSF)	• "Cisco Nonstop Forwarding" module in the Stateful Switchover Deployment Guide • "Cisco Nonstop Forwarding" module in the *High Availability Configuration Guide* • "EIGRP Nonstop Forwarding Awareness" module in the *IP Routing: EIGRP Configuration Guide*
Command Lookup Tool	http://tools.cisco.com/Support/CLILookup

Standards and RFCs

Standard/RFC	Title
RFC 4724	*Graceful Restart Mechanism for BGP*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

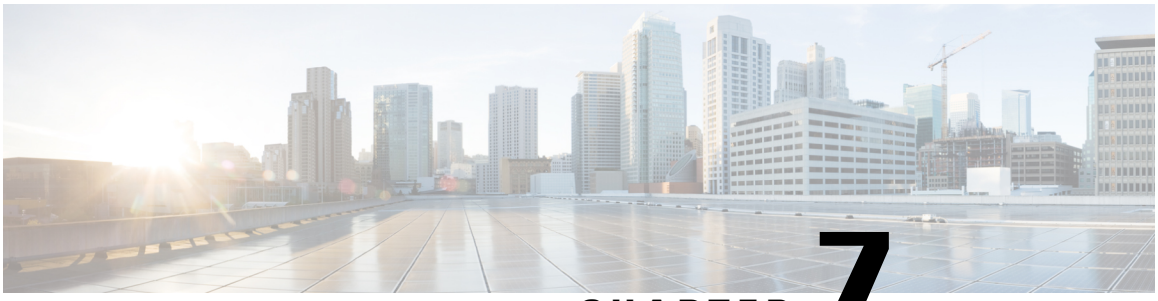
Feature Information for EIGRP IPv6 NSF/GR

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 14: Feature Information for EIGRP IPv6 NSF/GR

Feature Name	Releases	Feature Information
EIGRP IPv6 NSF/GR	15.2(1)E	The EIGRP IPv6 NSF/GR feature allows an NSF-aware router that is running EIGRP to forward IPv6 packets while the control plane restarts after recovering from a failure. The following commands were introduced or modified: **debug eigrp nsf**, **nsf**, **show ipv6 protocols**, **timers graceful-restart purge-time**, **timers nsf converge**, **timers nsf signal**.



CHAPTER 7

EIGRP Wide Metrics

The EIGRP Wide Metrics feature supports 64-bit metric calculations and Routing Information Base (RIB) scaling in Enhanced Interior Gateway Routing Protocol (EIGRP) topologies. The 64-bit calculations work only in EIGRP named mode configurations. EIGRP classic mode configurations use 32-bit calculations. This module provides an overview of the EIGRP Wide Metrics feature.

Information About EIGRP Wide Metrics

EIGRP Composite Cost Metrics

The Enhanced Interior Gateway Routing Protocol (EIGRP) uses bandwidth, delay, reliability, load, and K values (various constants that can be configured by a user to produce varying routing behaviors) to calculate the composite cost metric for local Routing Information Base (RIB) installation and route selections. The EIGRP composite cost metric is calculated using the following formula:

EIGRP composite cost metric = 256*((K1*Scaled Bw) + (K2*Scaled Bw)/(256 – Load) + (K3*Scaled Delay)*(K5/(Reliability + K4)))

EIGRP uses one or more vector metrics to calculate the composite cost metric. The table below lists EIGRP vector metrics and their descriptions.

Table 15: EIGRP Vector Metrics

Vector Metric	Description
bandwidth	The minimum bandwidth (Bw) of the route, in kilobits per second. It can be 0 or any positive integer. The bandwidth for the formula is scaled and inverted by using the following formula: Scaled Bw = (10^7/minimum bandwidth (Bw) in kilobits per second)
delay	Route delay, in tens of microseconds. Scaled Delay = (Delay/10)
load	The effective load of the route, expressed as a number from 0 to 255 (255 is 100 percent loading).
mtu	The minimum maximum transmission unit (MTU) size of the route, in bytes. It can be 0 or any positive integer.
reliability	The likelihood of successful packet transmission, expressed as a number between 0 and 255, where 255 means 100 percent reliability and 0 means no reliability.

EIGRP monitors metric weights, by using K values, on an interface to allow the tuning of EIGRP metric calculations and to indicate the type of service (ToS). K values are integers from 0 to 128; these integers, in conjunction with variables like bandwidth and delay, are used to calculate the overall EIGRP composite cost metric. The table below lists the K values and their defaults.

Table 16: EIGRP K-Value Defaults

Setting	Default Value
K1	1
K2	0
K3	1
K4	0
K5	0

Although you can configure K values to produce varying routing behaviors, most configurations use only the delay and bandwidth metrics by default, with bandwidth taking precedence, to produce a single 32-bit metric.

Use of the default constants effectively reduces the above-mentioned composite cost metric formula to the following default formula: 256*(Scaled Bw + Scaled Delay).

For example, let us consider a link whose bandwidth to a particular destination is 128 kb/s and the delay is 84,000 microseconds. By using the default formula, you can simplify the EIGRP composite cost metric calculation to 256*(Scaled Bw + Scaled Delay), thus resulting in the following value:

Metric = $256*(10^7/128 + 84000/10) = 256*86525 = 22150400$

EIGRP Wide Metrics

The Enhanced Interior Gateway Routing Protocol (EIGRP) composite cost metric (calculated using the bandwidth, delay, reliability, load, and K values) is not scaled correctly for high-bandwidth interfaces or Ethernet channels, resulting in incorrect or inconsistent routing behavior. The lowest delay that can be configured for an interface is 10 microseconds. As a result, high-speed interfaces, such as 10 Gigabit Ethernet (GE) interfaces, or high-speed interfaces channeled together (GE ether channel) will appear to EIGRP as a single GE interface. This may cause undesirable equal-cost load balancing. To resolve this issue, the EIGRP Wide Metrics feature supports 64-bit metric calculations and Routing Information Base (RIB) scaling that provide the ability to support interfaces (either directly or via channeling techniques like port channels or ether channels) up to approximately 4.2 terabits.

Note The 64-bit metric calculations work only in EIGRP named mode configurations. EIGRP classic mode uses 32-bit metric calculations.

To accommodate interfaces with bandwidths above 1 gigabit and up to 4.2 terabits and to allow EIGRP to perform path selections, the EIGRP composite cost metric formula is modified. The paths are selected based on the computed time. The time that information takes to travel through links is measured in picoseconds. The interfaces can be directly capable of these high speeds, or the interfaces can be bundles of links with an aggregate bandwidth greater than 1 gigabit.

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

Default K values are as follows:

- K1 = K3 = 1
- K2 = K4 = K5 = 0
- K6 = 0

The EIGRP Wide Metrics feature also introduces K6 as an additional K value for future use.

By default, the path selection scheme used by EIGRP is a combination of throughput (rate of data transfer) and latency (time taken for data transfer), and the formula for calculating the composite cost metric is as follows:

Composite Cost Metric = (K1*Minimum Throughput) + (K3*Total Latency)

Minimum Throughput = $(10^7* 65536)/Bw)$, where 65536 is the wide-scale constant.

Total Latency for bandwidths below 1 gigabit = (Delay*65536)/10, where 65536 is the wide-scale constant.

Total Latency for bandwidths above 1 gigabit = $(10^7* 65536/10)/ Bw$, 65536 is the wide-scale constant.

With the calculation of larger bandwidths, EIGRP can no longer fit the computed metric into a 4-byte unsigned long value that is needed by the Cisco RIB. To set the RIB scaling factor for EIGRP, use the **metric rib-scale** command. When you configure the **metric rib-scale** command, all EIGRP routes in the RIB are cleared and replaced with the new metric values.

EIGRP Metric Weights

You can use the **metric weights** command to adjust the default behavior of Enhanced Interior Gateway Routing Protocol (EIGRP) routing and metric computations. EIGRP metric defaults (K values) have been carefully selected to provide optimal performance in most networks.

Note Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default K values without guidance from an experienced network designer.

By default, the EIGRP composite cost metric is a 32-bit quantity that is the sum of segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. The formula used to scale and invert the bandwidth value is 10^7/minimum bandwidth in kilobits per second. However, with the EIGRP Wide Metrics feature, the EIGRP composite cost metric is scaled to include 64-bit metric calculations for EIGRP named mode configurations.

For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Gigabit Ethernet (GE), and serial lines running from 9600 bits per second to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Mismatched K Values

EIGRP K values are the metrics that EIGRP uses to calculate routes. Mismatched K values can prevent neighbor relationships from being established and can negatively impact network convergence. The example given below explains this behavior between two EIGRP peers (Device-A and Device-B).

The following configuration is applied to Device-A. The K values are changed using the **metric weights** command. A value of 2 is entered for the *k1* argument to adjust the bandwidth calculation. A value of 1 is entered for the *k3* argument to adjust the delay calculation.

```
Device(config)# hostname Device-A
Device-A(config)# interface serial 0
Device-A(config-if)# ip address 10.1.1.1 255.255.255.0
Device-A(config-if)# exit
Device-A(config)# router eigrp name1
Device-A(config-router)# address-family ipv4 autonomous-system 4533
Device-A(config-router-af)# network 10.1.1.0 0.0.0.255
Device-A(config-router-af)# metric weights 0 2 0 1 0 0 1
```

The following configuration is applied to Device-B, and the default K values are used. The default K values are 1, 0, 1, 0, 0, and 0.

```
Device(config)# hostname Device-B
Device-B(config)# interface serial 0
Device-B(config-if)# ip address 10.1.1.2 255.255.255.0
Device-B(config-if)# exit
Device-B(config)# router eigrp name1
Device-B(config-router)# address-family ipv4 autonomous-system 4533
Device-B(config-router-af)# network 10.1.1.0 0.0.0.255
```

```
Device-B(config-router-af)# metric weights 0 1 0 1 0 0 0
```

The bandwidth calculation is set to 2 on Device-A and set to 1 (by default) on Device-B. This configuration prevents these peers from forming a neighbor relationship.

The following error message is displayed on the console of Device-B because the K values are mismatched:

```
*Apr 26 13:48:41.811: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.1 (Ethernet0/0) is
 down: K-value mismatch
```

The following are two scenarios where the above error message can be displayed:

- Two devices are connected on the same link and configured to establish a neighbor relationship. However, each device is configured with different K values.
- One of two peers has transmitted a “peer-termination” message (a message that is broadcast when an EIGRP routing process is shut down), and the receiving device does not support this message. The receiving device will interpret this message as a K-value mismatch.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

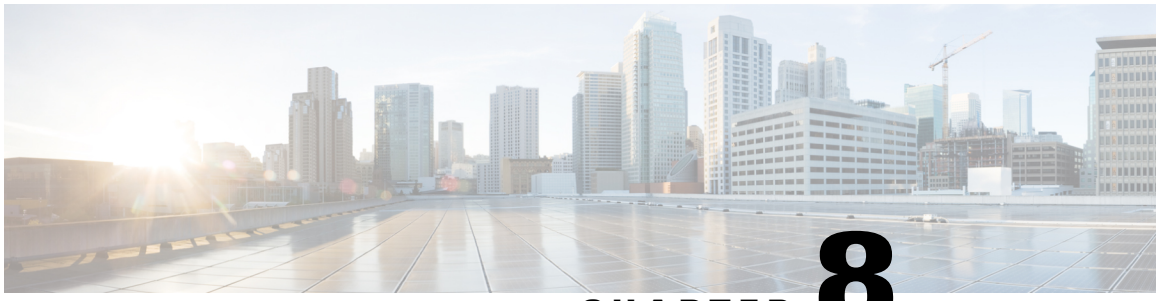
Feature Information for EIGRP Wide Metrics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 17: Feature Information for EIGRP Wide Metrics

Feature Name	Releases	Feature Information
EIGRP Wide Metrics	15.2(1)E	The EIGRP Wide Metrics feature introduces 64-bit metric calculations and RIB scaling in Enhanced Interior Gateway Routing Protocol (EIGRP) topologies. The following commands were introduced or modified by this feature: **metric rib-scale, metric weights**, **show eigrp address-family neighbors**, **show eigrp address-family topology**, **show eigrp plugins**, **show eigrp protocols**, **show eigrp tech-support**, **show ip eigrp neighbors**, and **show ip eigrp topology**.



CHAPTER 8

Add Path Support in EIGRP

The Add Path Support in EIGRP feature enables hubs in a single Dynamic Multipoint VPN (DMVPN) domain to advertise multiple best paths to connected spokes when the Enhanced Interior Gateway Routing Protocol (EIGRP) is the routing protocol between the hubs and the spokes. This module provides information about the Add Path Support in EIGRP feature and explains how to configure it.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Add Path Support in EIGRP

All interfaces in an Enhanced Interior Gateway Routing Protocol (EIGRP) topology are by default configured with the **next-hop-self** command. This command enables EIGRP to set the local outbound interface as the next-hop value while advertising a route to a peer, even when advertising routes out of the interface on which the routes were learned. This default EIGRP behavior may interfere with the **add-paths**command that helps

configure the Add Path Support in EIGRP feature. Therefore, before you configure this feature on a hub device in a Dynamic Multipoint VPN (DMVPN) domain, you must disable the **next-hop-self** command that is configured on the hub interface that connects to spokes in the DMVPN domain.

Restrictions for Add Path Support in EIGRP

- The Add Path Support in EIGRP feature can be enabled only in Enhanced Interior Gateway Routing Protocol (EIGRP) named mode configurations.
- The **variance** command should not be configured when the Add Path Support in EIGRP feature is enabled. The **variance** command alters the metrics of routes in an EIGRP topology, thereby enabling EIGRP to balance traffic among desired paths. Therefore, if you configure the **variance** command on a hub device, the command may interfere with the configuration of this feature.

Information About Add Path Support in EIGRP

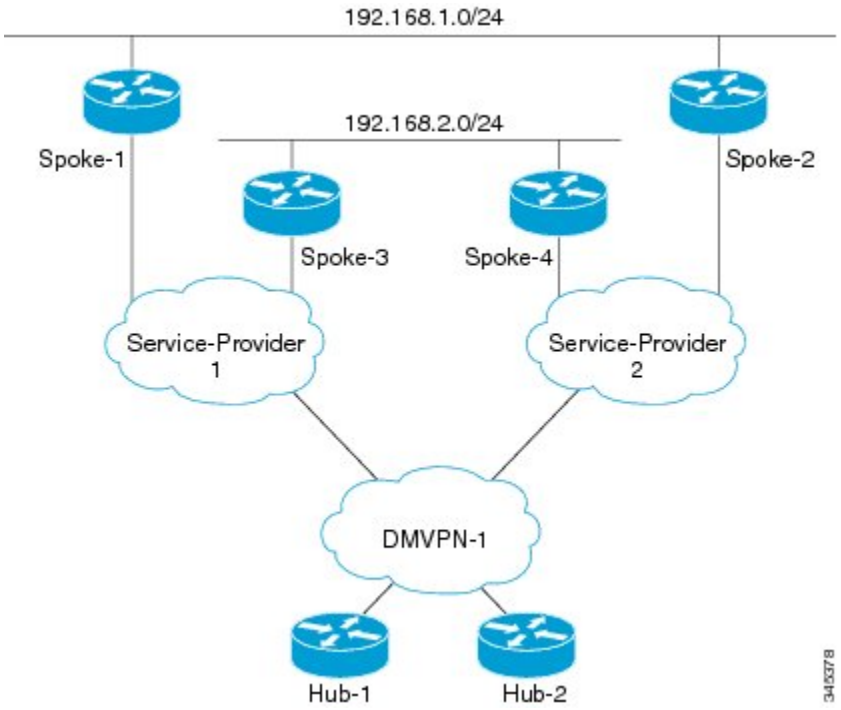
EIGRP Add Path Support Overview

In most Dynamic Multipoint VPN (DMVPN) domains, two or more spokes are connected to the same LAN segment. These spokes connect to more than one hub (for hub redundancy) through different service providers (for service-provider redundancy). In a single DMVPN domain, a hub connects to all spokes through one tunnel interface. In Enhanced Interior Gateway Routing Protocol (EIGRP) topologies, when a hub has more than one path (with the same metric but through different spokes) to reach the same network, both paths are chosen as best paths. However, by default, EIGRP advertises only one path as the best path to connected spokes. With the implementation of the Add Path Support in EIGRP feature, hubs in an EIGRP-DMVPN domain can advertise up to four additional best paths to connected spokes, thereby allowing load balancing and path redundancy. This feature supports both IPv4 and IPv6 configurations.

How Add Path Support in EIGRP Works

A typical single Dynamic Multipoint VPN (DMVPN) domain consists of dual hubs (for hub redundancy) connected to more than one service provider (for service-provider redundancy). In the figure below, two hub devices—Hub-1 and Hub-2—are connected through tunnel interfaces to a DMVPN domain.

Figure 8: Single DMVPN Domain



The DMVPN domain is in turn connected to two service providers—Service-Provider 1 and Service-Provider 2. Four spoke devices in this DMVPN domain—Spoke-1, Spoke-2, Spoke-3, and Spoke-4. Spoke-1 and Spoke-3 are connected to Service-Provider 1, and Spoke-2 and Spoke-4 are connected to Service-Provider 2. The Enhanced Interior Gateway Routing Protocol (EIGRP) is the routing protocol between the hubs and the spokes over the tunnel interfaces.

Spoke-1 and Spoke-2 are connected to a LAN with the network address 192.168.1.0/24. Both these spokes are connected to both the hubs through two different service providers, and hence, these spokes advertise the same LAN network to both hubs. Typically, spokes on the same LAN advertise the same metric; therefore, based on the metric, Hub-1 and Hub-2 have dual Equal-Cost Multipath (ECMP) routes to reach network 192.168.1.0/24. However, because EIGRP is a distance vector protocol, it advertises only one best path to the destination. Therefore, in this EIGRP-DMVPN domain, the hubs advertise only one route (for example, through Spoke-1) to reach network 192.168.1.0/24. When clients in subnet 192.168.2.0/24 communicate with clients in subnet 192.168.1.0/24, all traffic is directed to Spoke-1. Because of this default EIGRP behavior, there is no load balancing on Spoke-3 and Spoke-4. Additionally, if Spoke-1 fails or if the network of Service-Provider 1 goes down, EIGRP must reconverge to provide connectivity to 192.168.1.0/24.

The Add Path Support in EIGRP feature enables EIGRP to advertise up to four additional paths to connected spokes in a single DMVPN domain. If you configure this feature in the example topology discussed above,

both Spoke-1 and Spoke-2 will be advertised to Spoke-3 and Spoke-4 as best paths to network 192.168.1.0, thereby allowing load balancing among all spokes in this DMVPN domain.

How to Configure Add Path Support in EIGRP

Configuring IPv4 Add Path Support on a Hub

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. **address-family ipv4 autonomous-system** *as-number*
5. **af-interface** {**default** | *interface-type interface-number*}
6. **no next-hop-self** [**no-ecmp-mode**]
7. **add-paths** *number*
8. **end**
9. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.

	Command or Action	Purpose
Step 4	**address-family ipv4 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv4 autonomous-system 3`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 5	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface tunnel 0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**no next-hop-self** [**no-ecmp-mode**] **Example:** `Device(config-router-af-interface)# no next-hop-self no-ecmp-mode`	Instructs EIGRP to use the received next hop and not the local outbound interface address as the next hop to be advertised to neighboring devices.
Step 7	**add-paths** *number* **Example:** `Device(config-router-af-interface)# add-paths 4`	Enables EIGRP to advertise multiple paths as best paths to connected spokes in a single Dynamic Multipoint VPN (DMVPN) domain.
Step 8	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.
Step 9	**show running-config** **Example:** `Device# show running-config | section eigrp`	Displays contents of the current running configuration file. • Use the output modifier "|" to display the EIGRP section of the running configuration, and to verify whether the **add-paths** command is enabled in the configuration.

Configuring IPv6 Add Path Support on a Hub

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **router eigrp** *virtual-name*
5. **address-family ipv6 autonomous-system** *as-number*
6. **af-interface** {**default** | *interface-type interface-number*}
7. **no next-hop-self** [**no-ecmp-mode**]
8. **add-paths** *number*
9. **end**
10. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 unicast-routing** **Example:** `Device(config)# ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams.
Step 4	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an Enhanced Interior Gateway Routing Protocol (EIGRP) routing process and enters router configuration mode.

	Command or Action	Purpose
Step 5	**address-family ipv6 autonomous-system** *as-number* **Example:** `Device(config-router)# address-family ipv6 autonomous-system 3`	Enters address family configuration mode and configures an EIGRP routing instance.
Step 6	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface tunnel 0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 7	**no next-hop-self** [**no-ecmp-mode**] **Example:** `Device(config-router-af-interface)# no next-hop-self no-ecmp-mode`	Instructs EIGRP to use the received next-hop address and not the local outbound interface address as the next hop to be advertised to neighboring devices.
Step 8	**add-paths** *number* **Example:** `Device(config-router-af-interface)# add-paths 4`	Enables EIGRP to advertise multiple paths as best paths to connected spokes in a single Dynamic Multipoint VPN (DMVPN) domain.
Step 9	**end** **Example:** `Device(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.
Step 10	**show running-config** **Example:** `Device# show running-config | section eigrp`	Displays contents of the current running configuration file. • Use the output modifier "|" to display the EIGRP section of the running configuration, and to verify whether the **add-paths** command is enabled in the configuration.

Configuration Examples for Add Path Support in EIGRP

Example: Configuring IPv4 Add Path Support on a Hub

```
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 10
Device(config-router-af)# af-interface tunnel 0
```

```
Device(config-router-af-interface)# no next-hop-self no-ecmp-mode
Device(config-router-af-interface)# add-paths 4
Device(config-router-af-interface)# end
```

Example: Configuring IPv6 Add Path Support on a Hub

```
Device(config)# ipv6 unicast-routing
Device(config)# router eigrp name
Device(config-router)# address-family ipv6 autonomous-system 10
Device(config-router-af)# af-interface tunnel 0
Device(config-router-af-interface)# no next-hop-self no-ecmp-mode
Device(config-router-af-interface)# add-paths 4
Device(config-router-af-interface)# end
```

Additional References for Add Path Support in EIGRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP technology white papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

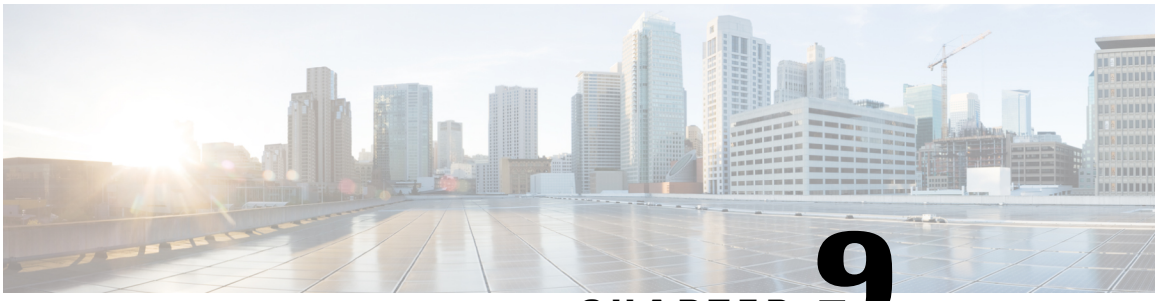
Feature Information for Add Path Support in EIGRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 18: Feature Information for Add Path Support in EIGRP

Feature Name	Releases	Feature Information
Add Path Support in EIGRP	Cisco IOS XE Release 3.8S 15.2(1)E	The Add Path Support in EIGRP feature enables a hub in a single Dynamic Multipoint VPN (DMVPN) domain to advertise multiple paths to connected spokes when the Enhanced Interior Gateway Routing Protocol (EIGRP) is the routing protocol between the hub and the spokes. The following command was introduced or modified: **add-paths**.



CHAPTER 9

EIGRP Stub Routing

The EIGRP stub routing feature improves network stability, reduces resource utilization, and simplifies the stub device configuration.

Stub routing is commonly used in hub-and-spoke network topologies. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote device (the spoke) that is connected to one or more distribution devices (the hub). The remote device is adjacent to one or more distribution devices. The only route for IP traffic to reach the remote device is through a distribution device.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About EIGRP Stub Routing

EIGRP Stub Routing

The EIGRP stub routing feature improves network stability, reduces resource utilization, and simplifies the stub device configuration.

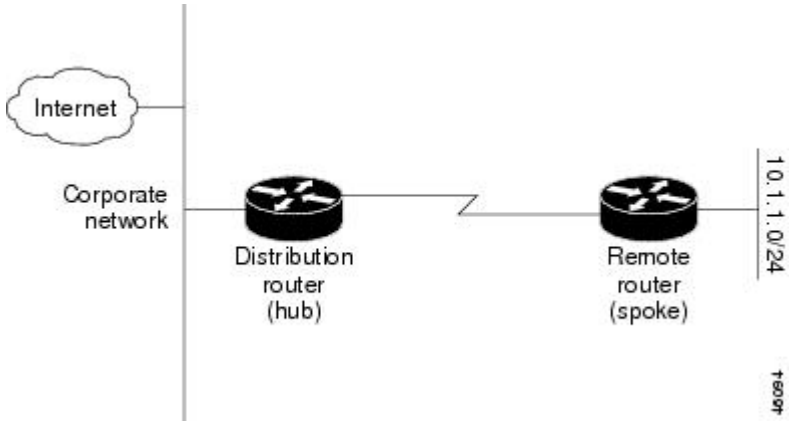
Stub routing is commonly used in hub-and-spoke network topologies. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote device (the spoke) that is connected to one or more distribution devices (the hub). The remote device is adjacent to one or more distribution devices. The only route for IP traffic to reach the remote device is through a distribution device. This type of configuration is commonly used in WAN topologies, where the distribution device is directly connected to a WAN. The distribution device can be connected to many remote devices, which is often the case. In a hub-and-spoke topology, the remote device must forward all nonlocal traffic to a distribution device, so it becomes unnecessary for the remote device to have a complete routing table. Generally, the distribution device need not send anything more than a default route to the remote device.

When using the EIGRP stub routing feature, you need to configure the distribution and remote devices to use EIGRP and configure only the remote device as a stub. Only specified routes are propagated from the remote (stub) device. The stub device responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message "inaccessible." A device that is configured as a stub will send a special peer information packet to all neighboring devices to report its status as a stub device.

Any neighbor that receives a packet informing it of the stub status will not query the stub device for any routes, and a device that has a stub peer will not query that peer. The stub device will depend on the distribution device to send proper updates to all peers.

The figure below shows a simple hub-and-spoke network.

Figure 9: Simple Hub-and-Spoke Network



The stub routing feature by itself does not prevent routes from being advertised to the remote device. In the above example, the remote device can access the corporate network and the Internet only through the distribution device. Having a complete route table on the remote device would serve no functional purpose because the path to the corporate network and the Internet would always be through the distribution device. The large route table would only reduce the amount of memory required by the remote device. Bandwidth and memory can be conserved by summarizing and filtering routes in the distribution device. The remote device need not

receive routes that have been learned from other networks because the remote device must send all nonlocal traffic, regardless of the destination, to the distribution device. If a true stub network is desired, the distribution device should be configured to send only a default route to the remote device. The EIGRP stub routing feature does not automatically enable summarization on distribution devices. In most cases, the network administrator will need to configure summarization on distribution devices.

Note When configuring the distribution device to send only a default route to the remote device, you must use the **ip classless** command on the remote device. By default, the **ip classless** command is enabled in all Cisco images that support the EIGRP stub routing feature.

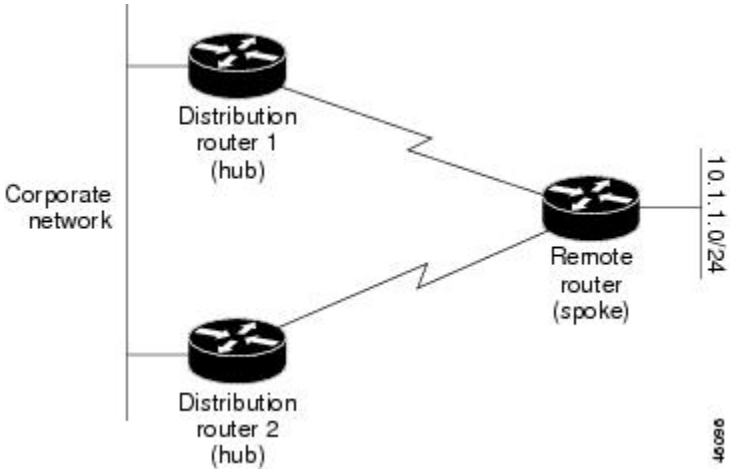
Without the EIGRP stub routing feature, even after routes that are sent from the distribution device to the remote device have been filtered or summarized, a problem might occur. If a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution device, which in turn would send a query to the remote device, even if routes are being summarized. If there is a communication problem (over the WAN link) between the distribution device and the remote device, an EIGRP stuck in active (SIA) condition could occur and cause instability elsewhere in the network. The EIGRP stub routing feature allows a network administrator to prevent queries from being sent to the remote device.

Dual-Homed Remote Topology

In addition to a simple hub-and-spoke network, where a remote device is connected to a single distribution device, the remote device can be dual-homed to two or more distribution devices. This configuration adds redundancy and introduces unique issues, and the stub feature helps to address some of these issues.

A dual-homed remote device will have two or more distribution (hub) devices. However, the principles of stub routing are the same as they are with a hub-and-spoke topology. The figure below shows a common dual-homed remote topology with one remote device: however, 100 or more devices could be connected on the same interfaces on distribution router 1 and distribution router 2. The remote device will use the best route to reach its destination. If distribution router 1 experiences a failure, the remote device can still use distribution router 2 to reach the corporate network.

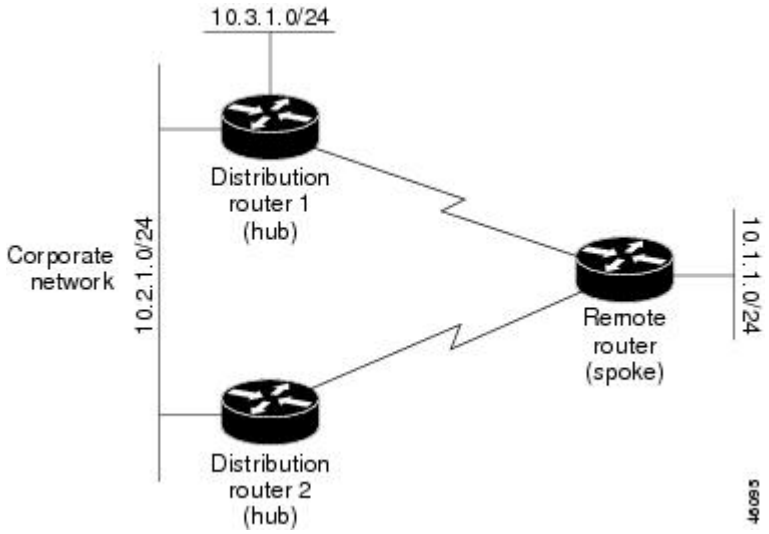
Figure 10: Simple Dual-Homed Remote Topology



The figure above shows a simple dual-homed remote topology with one remote device and two distribution devices. Both distribution devices maintain routes to the corporate network and stub network 10.1.1.0/24.

Dual-homed routing can introduce instability into an EIGRP network. In the figure below, distribution router 1 is directly connected to network 10.3.1.0/24. If summarization or filtering is applied on distribution router 1, the device will advertise network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution router 2 and the remote device).

Figure 11: Dual-Homed Remote Topology with Distribution Router 1 Connected to Two Networks

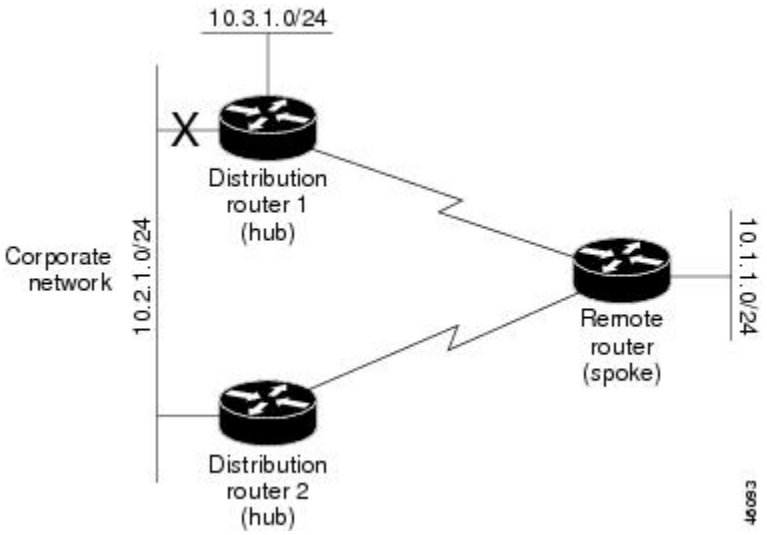


The figure above shows a simple dual-homed remote topology, where distribution router 1 is connected to both network 10.3.1.0/24 and network 10.2.1.0/24.

If the 10.2.1.0/24 link between distribution router 1 and distribution router 2 fails, the lowest cost path to network 10.3.1.0/24 from distribution router 2 will be through the remote device (see the figure below). This route is not desirable because the traffic that was previously traveling across the corporate network 10.2.1.0/24 would now be sent across a much lower bandwidth connection. The overutilization of the lower bandwidth WAN connection can cause many problems that might affect the entire corporate network. The use of the lower bandwidth route that passes through the remote device may cause WAN EIGRP distribution devices

to be dropped. Serial lines on distribution and remote devices may also be dropped, and EIGRP SIA errors on the distribution and core devices can occur.

Figure 12: Dual-Homed Remote Topology with a Failed Route to a Distribution Device



It is not desirable for traffic from distribution router 2 to travel through any remote device to reach network 10.3.1.0/24. Backup routes can be used if links are sized to manage the load. However, most networks, of the type shown in the figure above, have remote devices located at remote offices with relatively slow links. To ensure that traffic from distribution devices are not routed through a remote device, you can configure route summarization on the distribution device and the remote device.

It is typically undesirable for traffic from a distribution device to use a remote device as a transit path. A typical connection from a distribution device to a remote device would have much less bandwidth than a connection at the network core. Attempting to use a remote device with a limited bandwidth connection as a transit path would generally produce excessive congestion at the remote device. The EIGRP stub routing feature can prevent this problem by preventing the remote device from advertising core routes back to the distribution devices. In the above example, routes learned by the remote device from distribution router 1 will not be advertised to distribution router 2. Therefore, distribution router 2 will not use the remote device as a transit for traffic destined to the network core.

The EIGRP stub routing feature provides network stability. If the network is not stable, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit devices. Instead, distribution devices to which the stub device is connected answer queries on behalf of the stub device. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote devices to prevent those devices from appearing as transit paths to hub devices.

Caution The EIGRP stub routing feature should be used only on stub devices. A stub device is defined as a device connected to the network core or distribution layer through which core transit traffic should not flow. A stub device should not have any EIGRP neighbors other than distribution devices. Ignoring this restriction will cause undesirable behavior.

Note Multiaccess interfaces such as ATM, Gigabit Ethernet, Frame Relay, ISDN PRI, and X.25 are supported by the EIGRP stub routing feature only when all devices on that interface, except the hub, are configured as stub devices.

How to Configure EIGRP Stub Routing

Configuring the EIGRP Stub Routing Autonomous System Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **network** *ip-address* [**wildcard-mask**]
5. **eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**]
6. **end**
7. **show ip eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 1`	Configures a remote or distribution device to run an EIGRP process and enters router configuration mode.
Step 4	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.

	Command or Action	Purpose
Step 5	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Device(config-router)# eigrp stub connected static`	Configures a remote device as an EIGRP stub device.
Step 6	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 7	**show ip eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**] **Example:** `Device# show ip eigrp neighbors detail`	(Optional) Verifies that a remote device has been configured as a stub device with EIGRP. • Enter this command on the distribution device. The last line of the output displays the stub status of the remote or spoke device.

Configuring the EIGRP Stub Routing Named Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Enter one of the following:
   - **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
   - **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [**wildcard-mask**]
6. **eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**]
7. **exit-address-family**
8. **end**
9. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] [**neighbors**] [**static**] [**detail**] [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.

Step 4	Enter one of the following:
	
	• **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	Example:
	
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	
	`Device(config-router)# address-family ipv6 autonomous-system 45000`
		Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.

Step 5	**network** *ip-address* [**wildcard-mask**] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution device.
Step 6	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Device(config-router-af) eigrp stub leak-map map1`	Configures a device as a stub using EIGRP.

	Command or Action	Purpose
Step 7	**exit-address-family** **Example:** `Device(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.
Step 9	**show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] [**neighbors**] [**static**] [**detail**] [*interface-type interface-number*] **Example:** `Device# show eigrp address-family ipv4 neighbors detail`	(Optional) Displays neighbors discovered by EIGRP.

Configuration Examples for EIGRP Stub Routing

Example: EIGRP Stub Routing—Autonomous System Configuration

A device that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor devices by default. The following six keywords can be used with the **eigrp stub** command to modify this behavior:

- **connected**
- **leak-map**
- **receive-only**
- **redistributed**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command for an EIGRP autonomous system configuration.

Example: eigrp stub Command

In the following example, the **eigrp stub** command is used to configure the device as a stub that advertises connected and summary routes:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub
```

Example: eigrp stub connected static Command

In the following example, the **eigrp stub** command is used with the **connected** and **static** keywords to configure the device as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub connected static
```

Example: eigrp stub leak-map Command

In the following example, the **eigrp stub** command is issued with the **leak-map** *name* keyword-argument pair to configure the device to reference a leak map that identifies routes that would have been suppressed:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub leak-map map1
```

Example: eigrp stub receive-only Command

In the following example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the device as a receive-only neighbor (connected, summary, and static routes will not be sent):

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub receive-only
```

Example: eigrp stub redistributed Command

In the following example, the **eigrp stub** command is issued with the **redistributed** keyword to configure the device to advertise other protocols and autonomous systems:

```
Device(config)# router eigrp 1
Device(config-router)# network 10.0.0.0
Device(config-router)# eigrp stub redistributed
```

Example: EIGRP Stub Routing—Named Configuration

A device that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor devices by default. The following six keywords can be used with the **eigrp stub** command to modify this behavior:

- **connected**
- **leak-map**
- **receive-only**
- **redistributed**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command for an EIGRP named configuration.

Example: eigrp stub Command

In the following example, the **eigrp stub** command is used to configure the device as a stub that advertises connected and summary routes:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af) eigrp stub
```

Example: eigrp stub connected static Command

In the following named configuration example, the **eigrp stub** command is issued with the **connected** and **static** keywords to configure the device as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub connected static
```

Example: eigrp stub leak-map Command

In the following named configuration example, the **eigrp stub** command is issued with the **leak-map** *name* keyword-argument pair to configure the device to reference a leak map that identifies routes that would normally have been suppressed:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub leak-map map1
```

Example: eigrp stub receive-only Command

In the following named configuration example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the device as a receive-only neighbor (connected, summary, and static routes will not be sent):

```
Device(config)# router eigrp virtual-name1
```

```
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af)# eigrp stub receive-only
```

Example: eigrp stub redistributed Command

In the following named configuration example, the **eigrp stub** command is issued with the **redistributed** keyword to configure the device to advertise other protocols and autonomous systems:

```
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 4453
Device(config-router-af)# network 10.0.0.0
Device(config-router-af) eigrp stub redistributed
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
EIGRP commands	Cisco IOS IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

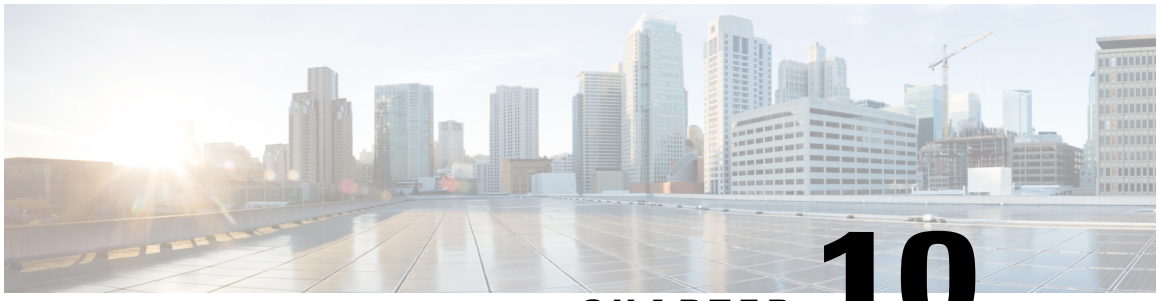
Feature Information for EIGRP Stub Routing

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 19: Feature Information for EIGRP Stub Routing

Feature Name	Releases	Feature Information
EIGRP Stub Routing	15.2(1)E	The EIGRP Stub Routing feature improves network stability, reduces resource utilization, and simplifies stub router configuration. Stub routing is commonly used in a hub-and-spoke network topology. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The following command was introduced or modified: **eigrp stub**.



CHAPTER 10

EIGRP Route Tag Enhancements

The EIGRP Route Tag Enhancements feature enables you to specify and display route tags in dotted-decimal format, filter routes using the route tag value with wildcard mask, and set a default route tag for all internal Enhanced Interior Gateway Routing Protocol (EIGRP) routes.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for EIGRP Route Tag Enhancements

- Default route tags are not supported in EIGRP autonomous system configurations.
- Route tags will not be displayed in dotted-decimal format if the **route-tag notation** global configuration command is not enabled on the device.

Information About EIGRP Route Tag Enhancements

EIGRP Route Tag Enhancements Overview

A route tag is a 32-bit value attached to routes. Route tags are used to filter routes and apply administrative policies, such as redistribution and route summarization, to tagged routes. You can tag routes within a route map by using the **set tag** command. You can match tagged routes and apply administrative policies to tagged routes within a route map by using the **match tag** or **match tag list** command. The **match tag list** command is used to match a list of route tags.

Prior to the EIGRP Route Tag Enhancements feature, EIGRP routes could only be tagged using plain decimals (range: 1 to 4294967295). This feature enables users to specify and display route tag values as dotted decimals (range: 0.0.0.0 to 255.255.255.255), similar to the format used by IPv4 addresses. This enhancement is intended to simplify the use of route tags as users can now filter routes by using the route tag wildcard mask.

This feature also allows you to configure a default route tag for all internal EIGRP routes without using route maps. Use the **eigrp default-route-tag** command in address family configuration mode to configure a default route tag for internal EIGRP routes.

How to Configure EIGRP Route Tag Enhancements

Enabling Dotted-Decimal Notation for Route Tags

Perform this task to enable route tags to be displayed as dotted decimals in **show** commands, irrespective of whether or not the tags were configured as dotted decimals.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-tag notation dotted-decimal**
4. **end**
5. Enter one of the following:
   - **show ip route tag**
   - **show ipv6 route tag**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-tag notation dotted-decimal** **Example:** `Device(config)# route-tag notation dotted-decimal`	Enables the display of route tags in dotted-decimal format.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

Step 5	Enter one of the following:
	• **show ip route tag**
	• **show ipv6 route tag**
	Example:
	`Device# show ip route tag`
	`Device# show ipv6 route tag`
		(Optional) Displays route tag entries for IPv4 or IPv6 routes.

Setting a Route Tag in a Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-name* **[permit | deny]** [*sequence-number*]
4. **set tag** {*tag-value* | *tag-value-dotted-decimal*}
5. **end**
6. **show route-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-name* **[permit | deny]** [*sequence-number*] **Example:** `Device(config)# route-map rip-to-eigrp`	Configures a route map and enters route-map configuration mode.
Step 4	**set tag** {*tag-value* | *tag-value-dotted-decimal*} **Example:** `Device(config-route-map)# set tag 7.7.7.7`	Sets a tag value for a route. **Note** In this example, all routes from Routing Information Protocol (RIP) to EIGRP are given a tag value of 7.7.7.7.
Step 5	**end** **Example:** `Device(config-route-map)# end`	Exits to privileged EXEC mode.
Step 6	**show route-map** **Example:** `Device# show route-map`	(Optional) Displays static and dynamic route maps configured on the router.

Matching a Route Tag in a Route Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-name* **[permit | deny]** [*sequence-number*]
4. **match tag** {*tag-value* | *tag-value-dotted-decimal*} [*...tag-value* | *tag-value-dotted-decimal*]
5. **end**
6. **show route-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-map** *map-name* **[permit | deny]** [*sequence-number*] **Example:** `Device(config)# route-map eigrp-to-rip`	Configures a route map and enters route-map configuration mode.
Step 4	**match tag** {*tag-value* | *tag-value-dotted-decimal*} [*...tag-value* | *tag-value-dotted-decimal*] **Example:** `Device(config-route-map)# match tag 10.10.10.0`	Filters routes that match specific route tags.
Step 5	**end** **Example:** `Device(config-route-map)# end`	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 6	**show route-map** **Example:** `Device# show route-map`	(Optional) Displays static and dynamic route maps configured on the device.

Creating a Route Tag List

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-dotted-decimal mask*
4. **end**
5. **show route-tag list** [*list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-dotted-decimal mask* **Example:** `Device(config)# route-tag list to-rip permit 10.10.10.0 0.0.0.7`	Creates a route tag list. • Route tag lists are used by route maps to match routes based on conditions specified in the route tag lists.
Step 4	**end** **Example:** `Device(config)# end`	Exits to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show route-tag list** [*list-name*] **Example:** `Device(config-router)# show route-tag list to-rip`	(Optional) Displays information about route tag lists configured on the device. • Use the *list-name* argument to display information about a specific route tag list.

Matching a Route Tag List

Route tag lists are used in route maps to match routes based on conditions specified in the route tag lists. Multiple route tag and mask pair sequences can be configured to permit or deny any condition for a list of route tags.

Note You can match either a route tag or a route tag list within a single route map sequence.

Perform this task to match routes based on conditions specified in the route tag list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-value-dotted-decimal mask*
4. **route-map** *map-name* **[permit | deny]** [*sequence-number*]
5. **match tag list** *list-name* [*...list-name*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**route-tag list** *list-name* **{deny | permit | sequence** *number* **{deny | permit}}** *tag-value-dotted-decimal mask* **Example:** `Device(config)# route-tag list list1 permit 10.10.10.0 0.0.0.7`	Configures a route tag list.
Step 4	**route-map** *map-name* **[permit | deny]** [*sequence-number*] **Example:** `Device(config)# route-map to-ospf`	Configures a route map and enters route-map configuration mode.
Step 5	**match tag list** *list-name* [*...list-name*] **Example:** `Device(config-route-map)# match tag list list1`	Filters routes that match a specified route tag list.
Step 6	**end** **Example:** `Device(config-route-map)# end`	Exits to privileged EXEC mode.

Setting a Default Route Tag for EIGRP Internal Routes

Perform this task to set a default route tag for all internal EIGRP routes without using a route map. Default route tags are supported only in EIGRP named mode configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-name*
4. Enter one of the following:
   - **address-family ipv4 unicast autonomous-system** *autonomous-system-number*
   - **address-family ipv6 unicast autonomous-system** *autonomous-system-number*
5. **eigrp default-route-tag** {*route-tag-plain-decimal* | *route-tag-dotted-decimal*}
6. **end**
7. Enter one of the following:
   - **show eigrp address-family ipv4 topology**
   - **show eigrp address-family ipv6 topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-name* **Example:** `Device(config)# router eigrp name`	Configures an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4 unicast autonomous-system** *autonomous-system-number* • **address-family ipv6 unicast autonomous-system** *autonomous-system-number*	Enters IPv4 or IPv6 address family configuration mode and configures an EIGRP routing instance.

	Command or Action	Purpose

	Example:
	`Device(config-router)# address-family ipv4 unicast autonomous-system 1`
	`Device(config-router)# address-family ipv6 unicast autonomous-system 1`	

Step 5	**eigrp default-route-tag** {*route-tag-plain-decimal* | *route-tag-dotted-decimal*} **Example:** `Device(config-router-af)# eigrp default-route-tag 10`	Sets a default route tag for all internal EIGRP routes.
Step 6	**end** **Example:** `Device(config-router-af)# end`	Exits to privileged EXEC mode.

Step 7	Enter one of the following:
	• **show eigrp address-family ipv4 topology**
	• **show eigrp address-family ipv6 topology**
	Example:
	`Device(config-router-af)# show eigrp address-family ipv4 topology`
	`Device(config-router-af)# show eigrp address-family ipv6 topology`
		(Optional) Displays entries of EIGRP address-family IPv4 or IPv6 topology tables.

Configuration Examples for EIGRP Route Tag Enhancements

Example: Enabling Dotted-Decimal Notation for Route Tags

The following example shows how to enable the display of route tags in dotted-decimal format by using the **route-tag notation** command. If you do not configure the **route-tag notation** command, route tags will be displayed as plain decimals in **show** commands even if the route tags were configured as dotted decimals. When you configure the **route-tag notation** command, route tags will be displayed as dotted decimals even if the route tags were configured as plain decimals.

```
Device# configure terminal
```

```
Device(config)# route-tag notation dotted-decimal
```

Example: Setting a Route Tag

The following example shows how to redistribute EIGRP routes into RIP and RIP routes into EIGRP by setting tags for routes within route maps:

```
Device(config)# route-map eigrp-to-rip
Device(config-route-map)# set tag 10.10.10.10
Device(config-route-map)# exit
Device(config)# route-map rip-to-eigrp
Device(config-route-map)# set tag 20.20.20.20
Device(config-route-map)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 7 route-map eigrp-to-rip metric 5
Device(config-router)# exit
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 autonomous-system 10
Device(config-router-af)# topology base
Device(config-router-af-topology)# redistribute rip route-map rip-to-eigrp 2 2 2 2 2
Device(config-router-af-topology)# end
```

Example: Matching a Route Tag

The following example shows how to redistribute EIGRP routes with a route tag value of 10.10.10.10 into a RIP domain:

```
Device(config)# route-map eigrp-to-rip
Device(config-route-map)# match tag 10.10.10.10
Device(config-route-map)# exit
Device(config)# router rip
Device(config-router)# redistribute eigrp 7 route-map eigrp-to-rip 5
Device(config-router)# end
```

Example: Configuring a Route Tag List

The following example shows how to configure a route tag list named TAG with various criteria for filtering routes. Route maps will use this list to match routes based on the criteria specified in the list. Route tag lists can accept route tags and wild card masks.

```
Device(config)# route-tag list TAG permit 1.1.1.1 0.0.0.1
Device(config)# route-tag list TAG seq 3 permit 2.2.2.2 0.0.0.3
Device(config)# route-tag list TAG seq 10 permit 3.3.3.3 0.0.0.7
Device(config)# route-tag list TAG seq 15 5.5.5.5 0.0.0.31
Device(config)# route-tag list TAG seq 20 deny 4.4.4.4 0.0.0.4
```

Example: Matching a Route Tag List

The following example shows how to use a route map to filter routes that match a specific route tag list. A single list can have multiple match criteria. All criteria must match before the route can be filtered. This

example shows how to configure a route tag list named List1 in a route map and use the **match tag list** command to filter routes that match the criteria listed in the route tag list.

```
Device(config)# route-tag list List1 permit 10.10.10.0 0.0.0.7
Device(config)# route-map to-ospf
Device(config-route-map)# match tag list List1
Device(config-route-map)# exit
Device(config)# router ospf 10
Device(config-router)# redistribute eigrp 7 route-map to-ospf metric 20
Device(config-router)# end
```

Example: Setting a Default Route Tag

The following example shows how to set a default route tag for all internal EIGRP routes without using a route map. Default route tags are supported only in EIGRP named configurations.

```
Device(config)# router eigrp name
Device(config-router)# address-family ipv4 unicast autonomous-system 1
Device(config-router-af)# eigrp default-route-tag 10.10.10.10
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
EIGRP commands	EIGRP Command Reference

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for EIGRP Route Tag Enhancements

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 20: Feature Information for Route Tag Enhancement

Feature Name	Releases	Feature Information
EIGRP Route Tag Enhancements	15.2(1)E	The EIGRP Route Tag Enhancements feature enables you to specify and display route tags in dotted-decimal format, filter routes using the route tag wildcard mask, and set a default route tag for all internal Enhanced Interior Gateway Routing Protocol (EIGRP) routes. The following commands were introduced or modified: **eigrp default-route-tag**, **match tag**, **match tag list**, **route-tag list**, **route-tag notation**, **set tag (IP)**, **show eigrp address-family topology**, **show ip eigrp topology**, **show ipv6 eigrp topology**, **show ip eigrp vrf topology, show ip route**, **show ip route tag**, **show ipv6 route tag**, **show ip route vrf**, **show ipv6 route vrf**, **show route map**, and **show route-tag list**.