IP Routing EIGRP Configuration Guide, Cisco IOS Release 12.2SR

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

Configuring EIGRP

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 is based on research conducted at SRI International and employs an algorithm referred to as 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. Routers 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 feature information and caveats, see 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 document.

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

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 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 routers. It is protocol-independent.
- 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 referred to as an autonomous system configuration. EIGRP autonomous system configuration creates an EIGRP routing instance that can be used for exchanging routing information.

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

It is recommended that you configure an autonomous system number when the address family is configured, either by entering the *autonomous-system-number*argument with the **address-family** command or separately 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 an EIGRP named configuration. An EIGRP named configuration does not create an EIGRP routing instance by itself. EIGRP named configuration is a base configuration that is required to define address family configurations that are used for routing.

In EIGRP named configuration, EIGRP VPNs can be configured in IPv4 and IPv6 named configurations. A VRF instance and a route distinguisher may or may not be used to create the address family.

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

EIGRP VPNs can be configured under EIGRP named configurations. A VRF and 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 router, which is determined by the number of VRFs, 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. 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.

EIGRP Neighbor Relationship Maintenance

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

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, while others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities (such as Ethernet) 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.

• Neighbor Authentication, page 3

Neighbor Authentication

The authentication to packets being sent between neighbors ensures that a router accepts packets only from other routers that have the same preshared key. Without this authentication configured you can purposely or accidentally add another router to the network, or send packets with different or conflicting route information on to the network, resulting in topology corruption and denial of service.

EIGRP authentication is configurable on a per-interface basis. Packets exchanged between neighbors connected through an interface are authenticated. Message digest algorithm 5 (MD5) authentication is supported to prevent the introduction of unauthorized information from unapproved sources. MD5 authentication is defined in RFC 1321. The Hash-based Message Authentication Code (HMAC)-Secure Hash Algorithms (SHA)-256 authentication method is also supported. When you are using the HMAC-SHA-256 authentication method, a shared secret key is configured in all routers attached to a common network. For each packet, the key is used to generate and verify a message digest that gets added to the packet. The message digest is a one-way function of the packet and the secret key. For more information on HMAC-SHA-256 authentication see *FIPS PUB 180-2, SECURE HASH STANDARD (SHS)* for the SHA-256 algorithm, and RFC 2104 for the HMAC algorithm.

If HMAC-SHA-256 authentication is set, the EIGRP packets will be authenticated using HMAC-SHA-256 message authentication codes. The HMAC algorithm takes as inputs the data to authenticate (that is, the EIGRP packet) and a shared secret key that is known to both the sender and the receiver, and outputs a 256-bit hash that is used for authentication. If the hash value provided by the sender matches the hash value calculated by the receiver, the packet is accepted by the receiver; otherwise it is discarded.

Typically, the shared secret key is configured to be identical between the sender and the receiver. To protect against packet replay attacks with a spoofed source address, the shared secret key to be used for a packet is defined as the concatenation of the user-configured shared secret (identical across all routers participating in the authenticated domain) with the IPv4 or IPv6 address (which is unique for each router) from which this particular packet is sent.

The router sending a packet calculates the hash to be sent based on:

- key part 1--the configured shared secret.
- key part 2--the local interface address from which the packet will be sent.
- data--the EIGRP packet to be sent (prior to addition of the IP header).

The router receiving the packet calculates the hash for verification based on:

- key part 1--the configured shared secret.
- key part 2--the IPv4 or IPv6 source address in the IPv4 or IPv6 packet header.
- data--the EIGRP packet received (after removing the IP header).

Therefore, for successful authentication on receipt, all of the following must be true:

- The sender and receiver must have the same shared secret.
- The source address chosen by the sender must match the source address in the IP header that the receiver receives.
- The EIGRP packet data that the sender transmitted must match the EIGRP packet data that the receiver receives.

Authentication cannot succeed if:

- The sender does not know the shared secret expected by the receiver.
- The IP source address in the IP header is modified in transit.
- Any of the EIGRP packet data is modified in transit.

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 a 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 router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but there are neighbors advertising the destination, a recomputation must occur. This process 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 any feasible successors it finds in order to avoid unnecessary recomputation.

Protocol-Dependent Modules

The 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 IP. It 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 by other IP routing protocols.

EIGRP Metric Weights

EIGRP 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**(EIGRP) command to adjust the default behavior of EIGRP routing and metric computations. For example, this adjustment allows you to tune the system behavior to allow for satellite transmission. EIGRP 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. The formula used to scale and invert the bandwidth value is 10^7/minimum Bw in kilobits per second.

For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media FDDI, Gigabit Ethernet, 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, page 5

Mismatched K Values

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

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

```
Router(config)# hostname Router-A
Router-A(config)# interface serial 0
Router-A(config-if)# ip address 10.1.1.1 255.255.255.0
Router-A(config-if)# exit
Router-A(config)# router eigrp virtual-name1
Router-A(config-router)# address-family ipv4 autonomous-system 4533
```

```
Router-A(config-router-af)# network 10.1.1.0 0.0.0.255
Router-A(config-router-af)# metric weights 0 2 0 1 0 0
```

The following configuration is applied to ROUTER-B. However, the **metric weights** command is not applied and the default K values are used. The default K values are 1, 0, 1, 0, and 0.

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

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

The following error message is displayed in the console of ROUTER-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
```

There are two scenarios where this error message can be displayed:

- The two routers are connected on the same link and configured to establish a neighbor relationship. However, each router is configured with different K values.
- The K-value mismatch error message can also be displayed if one of the two peers has transmitted a “goodbye” message, and the receiving router does not support this message. In this case, the receiving router will interpret this message as a K-value mismatch.

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 the 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 routers 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 router 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 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 will terminate the session when the hold timer expires. The sending and receiving routers will reconverge normally after the sender reloads.

Routing Metric Offset Lists

An offset list is the mechanism for increasing incoming and outgoing metrics to routes learned via EIGRP. An offset list provides a local mechanism for increasing the value of routing metrics. 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

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

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

EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4)))

The table below lists the EIGRP vector metrics and their descriptions.

Table 1: EIGRP Vector Metrics

Vector Metric	Description
bandwidth	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 Bw in kilobits per second)
delay	Route delay in tens of microseconds.
delay reliability	Likelihood of successful packet transmission expressed as a number between 0 and 255. The value 255 means 100 percent reliability; 0 means no reliability.
load	Effective load of the route expressed as a number from 0 to 255 (255 is 100 percent loading).
mtu	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 for the tuning of EIGRP metric calculations and indicate type of service (ToS). The table below lists the K values and their default.

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 the routers.

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

Using the cut-down formula, the EIGRP metric calculation would simplify to 256*(Bw + Delay), 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 tenths of microseconds.

When EIGRP calculates the delay for Mobile Ad Hoc Networks (MANET) and the delay is obtained from a router 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 subnets of 192.168.7.0 configured. Automatic summarization is performed when two or more **network** (EIGRP) router configuration or address family configuration commands are configured for the EIGRP process. By default, this feature is enabled.

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 in which additional summarization can be performed. 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 any more specific routes are in the routing table, EIGRP will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

Floating Summary Routes

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

The figure below shows a network with three routers, 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 configuration for an autonomous system configuration:

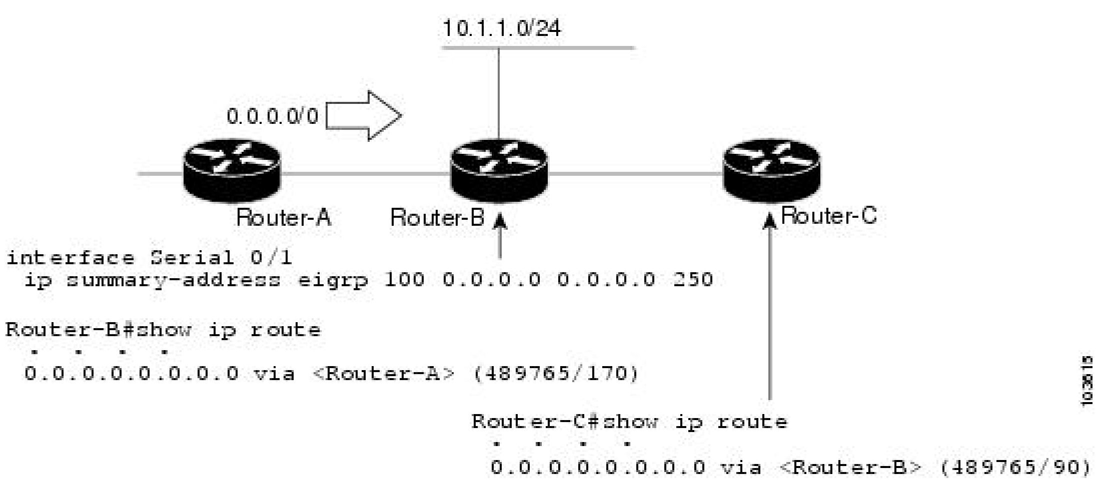
```
Router(config)# interface Serial 0/1

Router(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 configuration for a named configuration:

```
Router(config-router-af)# af-interface serial0/1
Router(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 on Router-B, a route for 0.0.0.0/0 to the null 0 interface with an administrative distance of 5. 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 0.0.0.0.0/0 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(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:

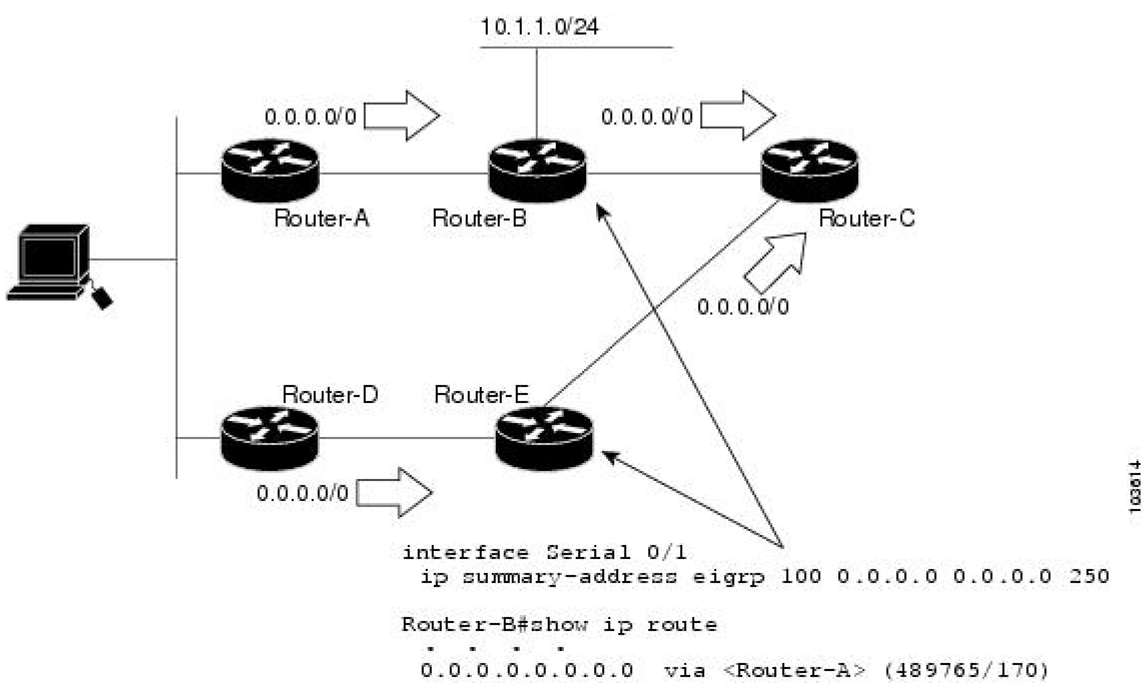
```
Router(config-router-af-interface)# summary-address eigrp 100 0.0.0.0 0.0.0.0 250
```

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 to Router-A or behind Router-A will be dropped when it 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 the 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 on only single-homed remote routers or areas where there is 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 cause a black hole to be formed.

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 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 the order from lowest to highest, and uses the first valid key it encounters. Note that the router needs to know the time to configure keys with lifetimes. Refer to the Network Time Protocol (NTP) and calendar commands in the Performing Basic System Management module of the*Cisco IOS Network Management Configuration Guide*.

For autonomous system and named configuration examples of route authentication, see the Example EIGRP Route Authentication--Autonomous System Configuration, page 62 and the Example EIGRP Route Authentication--Named Configuration, page 63.

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 routers 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 routers to receive hello packets from their neighbors. In this case, you may want to increase the hold time.

Note Do not adjust the hold time without advising 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 for 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 router 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. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

Link Bandwidth Percentage

By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth, as 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 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). 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 stub router configuration.

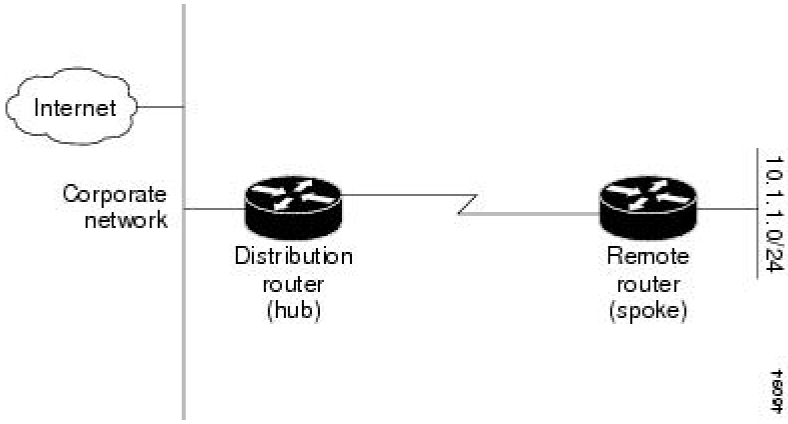
Stub routing is commonly used in a hub-and-spoke network topology. In a hub-and-spoke network, one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers.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

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

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

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

Figure 3: Simple Hub-and-Spoke Network



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

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

Without the EIGRP Stub Routing feature, even after the routes that are sent from the distribution router to the remote router 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 router, which in turn would send a query to the remote router even if routes are being summarized. If there is a problem communicating over the WAN link between the distribution router and the remote router, 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 router.

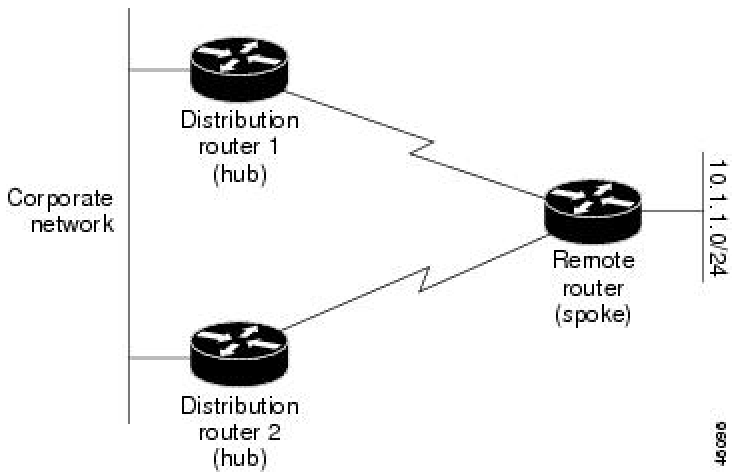
- Dual-Homed Remote Topology, page 13

Dual-Homed Remote Topology

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

A dual-homed remote router will have two or more distribution (hub) routers. 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 router, but 100 or more routers could be connected on the same interfaces on distribution router 1 and distribution router 2. The remote router will use the best route to reach its destination. If distribution router 1 experiences a failure, the remote router can still use distribution router 2 to reach the corporate network.

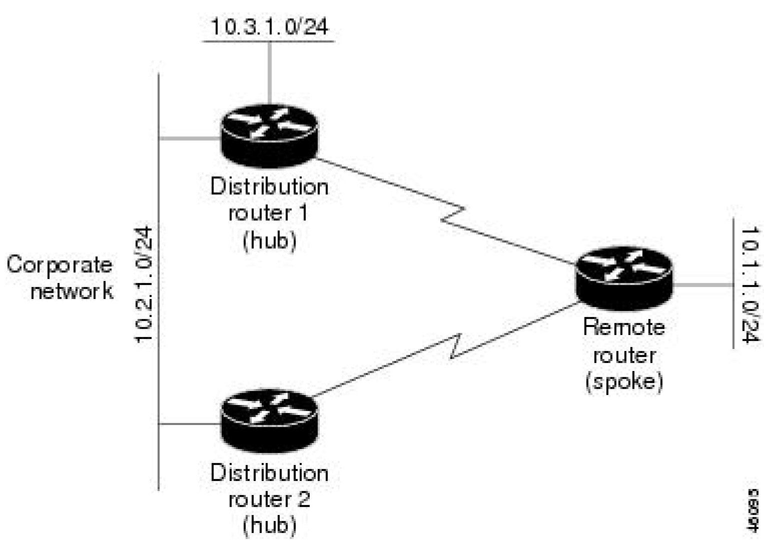
Figure 4: Simple Dual-Homed Remote Topology



The figure above shows a simple dual-homed remote with one remote router and two distribution routers. Both distribution routers 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 router will advertise network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution router 2 and the remote router).

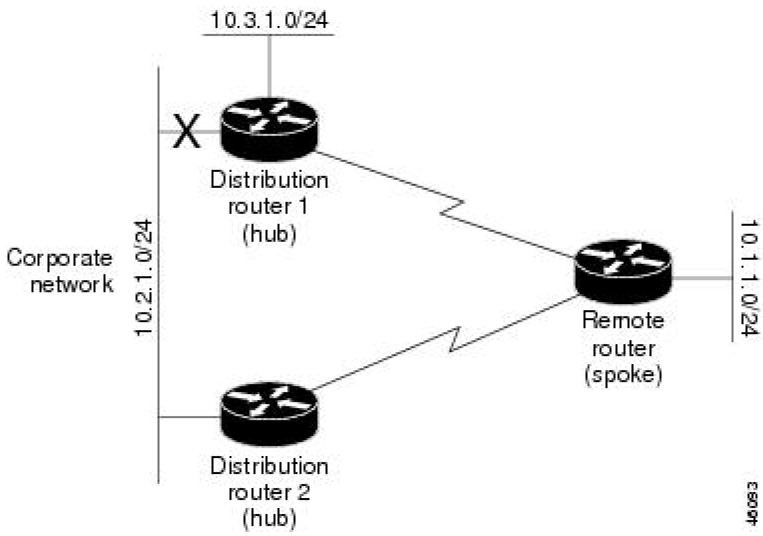
Figure 5: Dual-Homed Remote Topology with Distribution Router 1 Connected to Two Networks



The figure above shows a simple dual-homed remote router 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 has failed, the lowest cost path to network 10.3.1.0/24 from distribution router 2 is through the remote router (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 over utilization of the lower bandwidth WAN connection can cause a number of problems that might affect the entire corporate network. The use of the lower bandwidth route that passes through the remote router might cause WAN EIGRP distribution routers to be dropped. Serial lines on distribution and remote routers could also be dropped, and EIGRP SIA errors on the distribution and core routers could occur.

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



It is not desirable for traffic from distribution router 2 to travel through any remote router in order to reach network 10.3.1.0/24. If the links are sized to manage the load, it would be acceptable to use one of the backup routes. However, most networks of this type have remote routers located at remote offices with relatively slow links. This problem can be prevented if proper summarization is configured on the distribution router and remote router.

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

The EIGRP Stub Routing feature provides greater network stability. If the network is not stable, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit routers. Instead, distribution routers to which the stub router is connected answer the query on behalf of the stub router. 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 routers to prevent those remote routers from appearing as transit paths to the hub routers.

Caution

The EIGRP Stub Routing feature should be used only on stub routers. A stub router is defined as a router connected to the network core or distribution layer through which core transit traffic should not flow. A stub router should not have any EIGRP neighbors other than distribution routers. 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 routers on that interface, except the hub, are configured as stub routers.

EIGRP Stub Routing Leak Map Support

In EIGRP stub routing configurations where there is a remote site with more than one router, only one of the remote routers can be configured as the stub router. If you have two distribution layer routers, and two routers at a remote site, there is no way to declare both remote routers as stub routers. If one remote router is configured as a stub router, the other remote router cannot learn routes toward the network core if the link between the stub router and the distribution layer router fails and cannot route around the failed link.

The stub router cannot readvertise routes it has learned from any neighboring EIGRP router. To resolve this issue, a leak map configuration can be added to the EIGRP stub routing feature that allows a selected set of learned routes to be readvertised to other peers. The set of routes allowed through the stub router 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 the routes permitted through the stub from being readvertised into the core of the network.

Configure the **eigrp stub** command with the **leak-map** keyword 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 router 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 the interfaces in the 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** command with the *autonomous-system-number*argument creates an EIGRP configuration referred to as an autonomous system configuration. EIGRP autonomous system configuration creates an EIGRP routing instance that 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Router(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*	Associates networks with an EIGRP routing process.

	Command or Action	Purpose
	Example: `Router(config-router)# network 172.16.0.0`	
Step 5	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Enabling EIGRP Named Configuration

Perform this task to enable EIGRP and to create an EIGRP routing process. EIGRP sends updates to the interfaces in the 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** command with the *virtual-instance-name* argument creates an EIGRP configuration referred to as an EIGRP named configuration. EIGRP named configuration does not create an EIGRP routing instance by itself. An EIGRP named configuration is a base configuration that is required to define address family configurations under it that are used for routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Do 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
	Example: `Router# configure terminal`	
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.

Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	Example:
	`or`
	Example:
	`Router(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:** `Router(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Enabling EIGRP IPv6 VRF-Lite Named Configuration

Perform this task to enable IPv6 VRF-Lite in an EIGRP 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(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:** `Router(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:** `Router(config-router-af)# end`	Exits address family configuration mode and returns to privileged EXEC mode.

Configuring Optional EIGRP Parameters Autonomous System Configuration

Perform this task to configure optional EIGRP parameters including 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. **exit**

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.
Step 3	**router eigrp** *autonomous-system* **Example:** `Router(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:** `Router(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**passive-interface** [**default**] [*interface-type interface-number*] **Example:** `Router(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*]	(Optional) Applies an offset to routing metrics.

	Command or Action	Purpose
	Example: `Router(config-router)# offset-list 21 in 10 gigabitethernet 0/0/1`	
Step 7	**metric weights** *tos k1 k2 k3 k4 k5* **Example:** `Router(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** **Example:** `Router(config-router)# no auto-summary`	(Optional) Disables automatic summarization. **Note** Automatic summarization is enabled by default.
Step 9	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode.

Configuring Optional EIGRP Parameters Named Configuration

Perform this task to configure optional EIGRP named configuration parameters including applying offsets to routing metrics, adjusting EIGRP metrics, and disabling automatic summarization.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Do 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. **metric weights** *tos k1 k2 k3 k4 k5*
7. **af-interface** {**default** | *interface-type interface-number*}
8. **passive-interface** [**default**] [*interface-type interface-number*]
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. **exit-af-topology**

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.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Do one of the following: • **address-family ipv4** [**multicast**] [**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

	•
	• **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
	
	Example:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	
	Example:
	
	Example:
	`Router(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Router(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* **Example:** `Router(config-router-af)# 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 7	**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 8	**passive-interface** [**default**] [*interface-type interface-number*] **Example:** `Router(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:** `Router(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**	Exits address family interface configuration mode.

	Command or Action	Purpose
	Example: `Router(config-router-af-interface)# exit-af-interface`	
Step 11	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(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:** `Router(config-router-af-topology)# offset-list 21 in 10 ethernet 0`	(Optional) Applies an offset to routing metrics.
Step 13	**no auto-summary** **Example:** `Router(config-router-af-topology)# no auto-summary`	(Optional) Disables automatic summarization. **Note**Automatic summarization is enabled by default.
Step 14	**exit-af-topology** **Example:** `Router(config-router-af-topology)# exit-af-topology`	Exits address family topology configuration mode.

Configuring 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.

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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Router(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:** `Router(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:** `Router(config-router)# redistribute rip`	Redistributes routes from one routing domain into another routing domain.
Step 6	**distance eigrp** *internal-distance external-distance*	Allows the use of two administrative distances--internal and external--that could be a better route to a node.

	Command or Action	Purpose
	Example: `Router(config-router)# distance eigrp 80 130`	
Step 7	**default-metric** *bandwidth delay reliability loading mtu* **Example:** `Router(config-router)# default-metric 1000 100 250 100 1500`	Sets metrics for EIGRP.
Step 8	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring 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:** `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	**router eigrp** *autonomous-system* **Example:** `Router(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:** `Router(config-router)# exit`	Exits router configuration mode.
Step 5	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet0/0/1`	Enters interface configuration mode.
Step 6	**ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*] **Example:** `Router(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0`	(Optional) Configures a summary aggregate address.
Step 7	**ip bandwidth-percent eigrp** *as-number percent* **Example:** `Router(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:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring 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. Do 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** *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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	Example:
	Example:
	`Router(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**af-interface** *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 6	**summary-address** *ip-address mask* [*administrative-distance* [**leak-map** *leak-map-name*]] **Example:** `Router(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:** `Router(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 8	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(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:** `Router(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:** `Router(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring EIGRP Event Logging Autonomous System Configuration

Perform this task to configure event logging in an EIGRP 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Router(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:** `Router(config-router)# eigrp event-log-size 5000010`	(Optional) Sets the size of the EIGRP event log.
Step 5	**eigrp log-neighbor-changes** **Example:** `Router(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.
Step 6	**eigrp log-neighbor-warnings** [*seconds*]	(Optional) Enables the logging of EIGRP neighbor warning messages.

	Command or Action	Purpose
	Example: `Router(config-router)# eigrp log-neighbor-warnings 300`	
Step 7	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring EIGRP Event Logging Named Configuration

Perform this task to configure event logging in an EIGRP named configuration:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Do 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.

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

Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	Example:
	Example:
	`Router(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:** `Router(config-router-af)# eigrp log-neighbor-warnings 300`	(Optional) Enables the logging of EIGRP neighbor warning messages.
Step 6	**eigrp log-neighbor-changes** **Example:** `Router(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*}	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.

	Command or Action	Purpose
	Example: `Router(config-router-af)# topology base`	
Step 8	**eigrp event-log-size** *size* **Example:** `Router(config-router-af-topology)# eigrp event-log-size 10000`	(Optional) Sets the size of the EIGRP event log.
Step 9	**end** **Example:** `Router(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

Perform the following task to configure equal and unequal cost load balancing in an EIGRP 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:** `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	**router eigrp** *autonomous-system* **Example:** `Router(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:** `Router(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:** `Router(config-router)# maximum-paths 5`	Controls the maximum number of parallel routes that an IP routing protocol can support.
Step 6	**variance** *multiplier* **Example:** `Router(config-router)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 7	**end** **Example:** `Router(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Equal and Unequal Cost Load Balancing Named Configuration

Perform the following task to configure equal and unequal cost load balancing in an EIGRP named configuration:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Do 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	Example:
	`or`
	Example:
	`Router(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(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:** `Router(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:** `Router(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:** `Router(config-router-af-topology)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 9	**end** **Example:** `Router(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring EIGRP Route Authentication Autonomous System Configuration

Perform the following task to configure route authentication in an EIGRP autonomous system configuration:

Before you can configure EIGRP route authentication, you must enable EIGRP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot*
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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* **Example:** `Router(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:** `Router(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* **Example:** `Router(config-if)# ip authentication key-chain eigrp 1 keychain1`	Enables authentication of EIGRP packets.
Step 6	**exit**	Exits to global configuration mode.

	Command or Action	Purpose
	Example: `Router(config-if)# exit`	
Step 7	**key chain** *name-of-chain* **Example:** `Router(config)# key chain keychain1`	Identifies a key chain and enters key chain configuration mode.
Step 8	**key** *key-id* **Example:** `Router(config-keychain)# key 1`	Identifies the key number and enters key chain key configuration mode.
Step 9	**key-string** *text* **Example:** `Router(config-keychain-key)# key-string 0987654321`	Identifies the key string.
Step 10	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Router(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:** `Router(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:** `Router(config-keychain-key)# end`	Exits key chain key configuration mode and returns to privileged EXEC mode.

Configuring EIGRP Route Authentication Named Configuration

Perform the following task to configure route authentication in an EIGRP named configuration:

Before you can configure EIGRP route authentication, you must enable EIGRP.

SUMMARY STEPS

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

	Command or Action	Purpose

Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	
	Example:
	
	Example:
	`Router(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:** `Router(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:** `Router(config-router-af)# af-interface ethernet0/0`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 7	**authentication key-chain** *name-of-chain* **Example:** `Router(config-router-af-interface)# authentication key-chain SITE1`	Specifies an authentication key chain for EIGRP.
Step 8	**authentication mode** {**hmac-sha-256** *encryption-type password* | **md5**} **Example:** `Router(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**	Exits address family interface configuration mode.

	Command or Action	Purpose
	Example: `Router(config-router-af-interface)# exit-af- interface`	
Step 10	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 11	**exit** **Example:** `Router(config-router)# exit`	Exits router configuration mode and returns to global configuration mode.
Step 12	**key chain** *name-of-chain* **Example:** `Router(config)# key chain keychain1`	Identifies a key chain and enters key chain configuration mode.
Step 13	**key** *key-id* **Example:** `Router(config-keychain)# key 1`	Identifies the key number and enters key chain key configuration mode.
Step 14	**key-string** *text* **Example:** `Router(config-keychain-key)# key-string 0987654321`	Identifies the key string.
Step 15	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Router(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 16	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} **Example:** `Router(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:** `Router(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 Autonomous System Configuration

Perform the following task to adjust the interval between hello packets and the hold time in an EIGRP 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** *slot* / *port*
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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Router(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:** `Router(config-router)# exit`	Exits to global configuration mode.
Step 5	**interface** *slot* / *port*	Enters interface configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# interface gigabitethernet 0/1/1`	
Step 6	**ip hello-interval eigrp** *autonomous-system-number seconds* **Example:** `Router(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:** `Router(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 the advice from your technical support personnel.
Step 8	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

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

Perform the following task to adjust the interval between hello packets and the hold time in an EIGRP named configuration:

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Do 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	Example:
	Example:
	`Router(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**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 6	**hello-interval** *seconds* **Example:** `Router(config-router-af-interface)# hello-interval 10`	Configures the hello interval for an EIGRP address family named configuration.
Step 7	**hold-time** *seconds* **Example:** `Router(config-router-af-interface)# hold-time 50`	Configures the hold time for an EIGRP address family named configuration.
Step 8	**end** **Example:** `Router(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Disabling Split Horizon Autonomous System Configuration

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.

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *slot* / *port*
4. **no ip split-horizon eigrp** *autonomous-system-number*
5. **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.
Step 3	**interface** *slot* / *port* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 4	**no ip split-horizon eigrp** *autonomous-system-number* **Example:** `Router(config-if)# no ip split-horizon eigrp 101`	Disables split horizon.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling Split Horizon and Next-Hop-Self Named Configuration

EIGRP will, by default, set 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 where it learned them. Perform this task to change this default to instruct EIGRP to use the received next hop value when advertising these routes. Disabling next-hop-self is primarily useful in Dynamic Multipoint VPN (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. Do 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**
8. **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.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Do 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:
	`Router(config-router)# address-family ipv4 autonomous-system 45000`
	Example:
	`or`
	Example:
	`Router(config-router)# address-family ipv6 autonomous-system 45000`	

Step 5	**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 6	**no split-horizon** **Example:** `Router(config-router-af-interface)# no split-horizon`	Disables EIGRP split horizon.
Step 7	**no next-hop-self** **Example:** `Router(config-router-af-interface)# no next-hop-self`	(Optional) Instructs an EIGRP router to use the received next hop rather than the local outbound interface address as the next hop.
Step 8	**end** **Example:** `Router(config-router-af-interface)# end`	Exits address family interface configuration mode and returns to privileged EXEC mode.

Configuring EIGRP Stub Routing Autonomous System Configuration

Perform the following task to configure stub routing in an EIGRP 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Router(config)# router eigrp 1`	Configures a remote or distribution router to run an EIGRP process and enters router configuration mode.
Step 4	**network** *ip-address* [**wildcard-mask**] **Example:** `Router(config-router)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution router.
Step 5	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Router(config-router)# eigrp stub connected static`	Configures a remote router as an EIGRP stub router.
Step 6	**end** **Example:** `Router(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:** `Router# show ip eigrp neighbors detail`	(Optional) Verifies that a remote router has been configured as a stub router with EIGRP. • Enter this command from the distribution router. The last line of the output displays the stub status of the remote or spoke router.

Configuring EIGRP Stub Routing Named Configuration

Perform the following task to configure stub routing in an EIGRP named configuration:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. Do 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:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enables an EIGRP routing process in global configuration mode and enters router configuration mode.
Step 4	Do 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: `Router(config-router)# address-family ipv4 autonomous-system 45000` **Example:** `or` **Example:** `Router(config-router)# address-family ipv6 autonomous-system 45000`	
Step 5	**network** *ip-address* [**wildcard-mask**] **Example:** `Router(config-router-af)# network 172.16.0.0`	Specifies the network address of the EIGRP distribution router.
Step 6	**eigrp stub** [**receive-only**] [**leak-map** *name*] [**connected**] [**static**] [**summary**] [**redistributed**] **Example:** `Router(config-router-af) eigrp stub leak-map map1`	Configures a router as a stub using EIGRP.
Step 7	**exit-address-family** **Example:** `Router(config-router-af)# exit-address-family`	Exits address family configuration mode.
Step 8	**end** **Example:** `Router(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:** `Router# show eigrp address-family ipv4 neighbors detail`	(Optional) Displays the neighbors that are discovered by EIGRP.

Monitoring and Maintaining EIGRP Autonomous System Configuration

Perform the following steps to monitor and maintain an 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*] **neighbors** [*interface-type* | **static** | **detail**]
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:

```
Router# enable
```

Step 2 **show ip eigrp** [**vrf** {*vrf-name* | *****}] **[** *autonomous-system-number*] **accounting**

This command displays prefix accounting information for EIGRP processes. The following is sample output from the command:

Example:

```
Router# show ip eigrp vrf VRF1 accounting
EIGRP-IPv4 Accounting for AS(100)/ID(10.0.2.1) VRF(VRF1)
Total Prefix Count: 4  States: A-Adjacency, P-Pending, D-Down
State Address/Source   Interface        Prefix   Restart  Restart/
                                        Count     Count   Reset(s)
 P    Redistributed     ----              0         3         211
 A    10.0.1.2          Gi0/0             2         0          84
 P    10.0.2.4          Se2/0             0         2         114
 D    10.0.1.3          Gi0/0             0         3           0
```

Step 3 **show ip eigrp events** [*starting-event-number ending-event-number*] [**type**]

This command displays the EIGRP event log. The following is sample output from the command:

Example:

```
Router# show ip eigrp events
1    02:37:58.171 NSF stale rt scan, peer: 10.0.0.0
2    02:37:58.167 Metric set: 10.0.0.1/24 284700416
3    02:37:58.167 FC sat rdbmet/succmet: 284700416 0
4    02:37:58.167 FC sat nh/ndbmet: 10.0.0.2 284700416
5    02:37:58.167 Find FS: 10.0.0.0/24 284700416
6    02:37:58.167 Rcv update met/succmet: 284956416 284700416
7    02:37:58.167 Rcv update dest/nh: 10.0.0.0/24 10.0.0.1
8    02:37:58.167 Peer nsf restarted: 10.0.0.1 Tunnel0
9    02:36:38.383 Metric set: 10.0.0.0/24 284700416
10   02:36:38.383 RDB delete: 10.0.0.0/24 10.0.0.1
11   02:36:38.383 FC sat rdbmet/succmet: 284700416 0
12   02:36:38.383 FC sat nh/ndbmet: 0.0.0.0 284700416
```

Step 4 **show ip eigrp interfaces** [**vrf** {*vrf-name*| *****}] [*autonomous-system-number*] [*type number*] [**detail**]

This command displays information about interfaces that are configured for EIGRP. The following is sample output from the command:

Example:

```
Router# show ip eigrp interfaces
EIGRP-IPv4 Interfaces for AS(60)
                    Xmit Queue    Mean   Pacing Time   Multicast   Pending
Interface   Peers   Un/Reliable   SRTT   Un/Reliable   Flow Timer  Routes
Gi0           0        0/0          0      11/434          0          0
Gi0           1        0/0        337       0/10           0          0
SE0:1.16      1        0/0         10       1/63         103          0
Tu0           1        0/0        330       0/16           0          0
```

Step 5 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **neighbors** [*interface-type* | **static** | **detail**]

This command displays neighbors discovered by EIGRP. The following is sample output from this command:

Example:

```
Router# show ip eigrp neighbors
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.1.2                 Gi0/0             13 00:00:03 1996  5000  0  5
2   10.1.1.9                 Gi0/0             14 00:02:24   206  5000  0  5
1   10.1.2.3                 Gi0/1             11 00:20:39   2202  5000  0  5
```

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**]

This command displays entries in the EIGRP topology table. The following is sample output from this command:

Example:

```
Router#
show ip eigrp topology

EIGRP-IPv4 Topology Table for AS(1)/ID(10.0.0.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status, s - sia status
P 10.0.0.0/8, 1 successors, FD is 409600
        via 10.0.0.1 (409600/128256), GigabirEthernet0/0
P 172.16.1.0/24, 1 successors, FD is 409600
        via 10.0.0.1 (409600/128256), GigabitEthernet0/0
P 10.0.0.0/8, 1 successors, FD is 281600
        via Summary (281600/0), Null0
P 10.0.1.0/24, 1 successors, FD is 281600
        via Connected, GigabitEthernet0/0
```

Step 7 **show ip eigrp** [**vrf** {*vrf-name* | *****}] [*autonomous-system-number*] **traffic**

This command displays the number of EIGRP packets sent and received. The following is sample output from the command:

Example:

```
Router# show ip eigrp traffic
EIGRP-IPv4 Traffic Statistics for AS(60)
Hellos sent/received: 21429/2809
Updates sent/received: 22/17
Queries sent/received: 0/0
Replies sent/received: 0/0
Acks sent/received: 16/13
SIA-Queries sent/received: 0/0
SIA-Replies sent/received: 0/0
Hello Process ID: 204
PDM Process ID: 203
Socket Queue: 0/2000/2/0 (current/max/highest/drops)
Input Queue: 0/2000/2/0 (current/max/highest/drops)
```

Monitoring and Maintaining EIGRP Named Configuration

Perform the following steps to monitor and maintain an 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:

```
Router# enable
```

Step 2 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**

This command displays prefix accounting information for EIGRP processes. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 22 accounting
EIGRP-IPv4 VR(saf) Accounting for AS(22)/ID(10.0.0.1)
Total Prefix Count: 3  States: A-Adjacency, P-Pending, D-Down
State Address/Source    Interface       Prefix   Restart  Restart/
                                        Count     Count   Reset(s)
 A    10.0.0.2          Gi0/0              2         0        0
```

```
 P    10.0.2.4          Se2/0               0         2        114
 D    10.0.1.3          Gi0/0               0         3        0
```

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**]

This command displays information about EIGRP address-family events. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 3 events
Event information for AS 3:
1 15:37:47.015 Change queue emptied, entries: 1
2 15:37:47.015 Metric set: 10.0.0.0/24 307200
3 15:37:47.015 Update reason, delay: new if 4294967295
4 15:37:47.015 Update sent, RD: 10.0.0.0/24 4294967295
5 15:37:47.015 Update reason, delay: metric chg 4294967295
6 15:37:47.015 Update sent, RD: 10.0.0.0/24 4294967295
7 15:37:47.015 Route installed: 10.0.0.0/24 10.0.1.2
8 15:37:47.015 Route installing: 10.0.0.0/24 10.0.1.2
```

Step 4 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **interfaces** [**detail**] [*interface-type interface-number*]

This command displays information about interfaces that are configured for EIGRP. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 4453 interfaces
EIGRP-IPv4 VR(Virtual-name) Address-family Neighbors for AS(4453)
                    Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Services
Se0           1          0/0        28      0/15          127           0
Se1           1          0/0        44      0/15          211           0
```

Step 5 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **neighbors**[**static**] [**detail**] [*interface-type interface-number*]

This command displays the neighbors that are discovered by EIGRP. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 4453 neighbors
EIGRP-IPv4 VR(Virtual-name) Address-family Neighbors for AS(4453)
Address                         Interface              Hold Uptime  SRTT  RTO    Q     Seq
                                     (sec)         (ms)  (ms)   Cnt   Num
172.16.81.28            GigabitEthernet1/1/1     13   0:00:41  0      11   4     20
172.16.80.28            GigabitEthernet0/0/1     14   0:02:01  0      10   12    24
172.16.80.31            GigabitEthernet0/1/1     12   0:02:02  0      4    5     20
```

Step 6 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **timers**

This command displays information about EIGRP timers and expiration times. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 4453 timers
EIGRP-IPv4 VR(Virtual-name) Address-family Timers for AS(4453)
Hello Process
Expiration Type
| 1.022 (parent)
| 1.022 Hello (Et0/0)
Update Process
```

```
Expiration Type
| 14.984 (parent)
| 14.984 (parent)
| 14.984 Peer holding
SIA Process
Expiration Type for Topo(base)
| 0.000 (parent)
```

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**}]

This command displays entries in the EIGRP topology table. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 4453 topology
EIGRP-IPv4 VR(Virtual-name) Topology Table for AS(4453)/ID(10.0.0.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status, s - sia Status
P  10.17.17.0/24, 1 successors, FD is 409600
          via 10.10.10.2 (409600/128256), GigabitEthernet3/0/1
P  172.16.19.0/24, 1 successors, FD is 409600
          via 10.10.10.2 (409600/128256), GigabitEthernet3/0/1
P  192.168.10.0/24, 1 successors, FD is 281600
          via Connected, GigabitEthernet3/0/1
P  10.10.10.0/24, 1 successors, FD is 281600
          via Redistributed (281600/0)
```

Step 8 **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **traffic**

This command displays the number of EIGRP packets that are sent and received. The following is sample output from the command:

Example:

```
Router# show eigrp address-family ipv4 4453 traffic
EIGRP-IPv4 VR(virtual-name) Address-family Traffic Statistics for AS(4453)
  Hellos sent/received: 122/122
  Updates sent/received: 3/1
  Queries sent/received: 0/0
  Replies sent/received: 0/0
  Acks sent/received: 0/3
  SIA-Queries sent/received: 0/0
  SIA-Replies sent/received: 0/0
  Hello Process ID: 128
  PDM Process ID: 191
  Socket Queue: 0/2000/1/0 (current/max/highest/drops)
  Input Queue: 0/2000/1/0 (current/max/highest/drops
```

Step 9 **show eigrp plugins** [*plugin-name*] [**detailed**]

This command displays general information including the versions of the EIGRP protocol features that are currently running. The following is sample output from the command:

Example:

```
Router# show eigrp plugins
EIGRP feature plugins:::
    eigrp-release      :   5.00.00 : Portable EIGRP Release
                       :  19.00.00 : Source Component Release(rel5)
    igrp2              :   3.00.00 : Reliable Transport/Dual Database
    bfd                :   1.01.00 : BFD Platform Support
    mtr                :   1.00.01 : Multi-Topology Routing(MTR)
    eigrp-pfr          :   1.00.01 : Performance Routing Support
    ipv4-af            :   2.01.01 : Routing Protocol Support
    ipv4-sf            :   1.01.00 : Service Distribution Support
    external-client    :   1.02.00 : Service Distribution Client Support
    ipv6-af            :   2.01.01 : Routing Protocol Support
```

```
    ipv6-sf            :   1.01.00 : Service Distribution Support
    snmp-agent         :   1.01.01 : SNMP/SNMPv2 Agent Support
```

Step 10 **show eigrp protocols** [**vrf** *vrf-name*]

This command displays general information about EIGRP protocols that are currently running. The following is sample output from the command:

Example:

```
Router# show eigrp protocols
EIGRP-IPv4 Protocol for AS(10)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 10.0.1.1
Topology : 0 (base)
Active Timer: 3 min
Distance: internal 90 external 170
Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1
EIGRP-IPv4 Protocol for AS(5) VRF(VRF1)
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 10.1.2.1
Topology : 0 (base)
Active Timer: 3 min
Distance: internal 90 external 170
Maximum path: 4
Maximum hopcount 100
Maximum metric variance 1
Total Prefix Count: 0
Total Redist Count: 0
```

Configuration Examples for EIGRP

Example Enabling EIGRP--Autonomous System Configuration

The following example shows how to configure EIGRP:

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

Example Enabling EIGRP--Named Configuration

The following example shows how to enable EIGRP named configuration:

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

Example Enabling EIGRP IPv6 VRF-Lite--Named Configuration

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

```
Router> enable
Router# configure terminal
Router(config)# vrf definition vrf1
Router(config-vrf)# rd 100:1
Router(config-vrf)# address-family ipv6
Router(config-vrf-af)# exit
Router(config-vrf)# exit
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv6 vrf vrf1 autonomous-system 45000
Router(config-router-af)# network 172.16.0.0
```

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:

```
Router> enable
Router# configure terminal
Router(config)# router eigrp 1
Router(config-router)# network 172.16.0.0
Router(config-router)# passive-interface
Router(config-router)# offset-list 21 in 10 ethernet 0

Router(config-router)# metric weights 0 2 0 2 0 0
Router(config-router)# no auto-summary
Router(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, and disabling automatic summarization:

```
Router> enable
Router# configure terminal
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# network 172.16.0.0
Router(config-router-af)# metric weights 0 2 0 2 0 0
Router(config-router-af)# af-interface ethernet0/0
Router(config-router-af-interface)# passive-interface
Router(config-router-af-interface)# bandwidth-percent 75
Router(config-router-af-interface)# exit-af-interface
Router(config-router-af-interface)# topology base
Router(config-router-af-topology)# offset-list 21 in 10 ethernet 0

Router(config-router-af-topology)# no auto-summary
Router(config-router-af-topology)# exit-af-topology
```

Example EIGRPRedistribution--AutonomousSystemConfiguration

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:

```
Router> enable
Router# configure terminal
Router(config)# router eigrp 1
Router(config-router)# network 172.16.0.0
Router(config-router)# redistribute rip
Router(config-router)# distance eigrp 80 130
Router(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 also configures the automatic summary feature for an EIGRP autonomous system configuration. This configuration causes EIGRP to summarize network 10.0.0.0 from the Ethernet interface 0 only.

```
Router> enable
Router# configure terminal
Router(config)# router eigrp 101
Router(config-router)# exit
Router(config)# interface ethernet0
Router(config-if)# ip summary-address eigrp 100 0.0.0.0 0.0.0.0
Router(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. This causes the creation of 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 from 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, this 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 a **distribute-list**command. You can configure this command to filter all outbound route advertisements sent out 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 from the Ethernet interface 0/0 only.

```
Router> enable
Router# configure terminal
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# af-interface ethernet 0/0
Router(config-router-af-interface)# summary-address 192.168.0.0 255.255.0.0
Router(config-router-af-interface)# exit-af-interface
Router(config-router-af)# topology base
Router(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:

```
Router> enable
Router# configure terminal
Router(config)# router eigrp 1
Router(config-router)# eigrp event-log-size 5000
Router(config-router)# eigrp log-neighbor-changes
Router(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:

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

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

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

Example EIGRP Route Authentication--Autonomous System Configuration

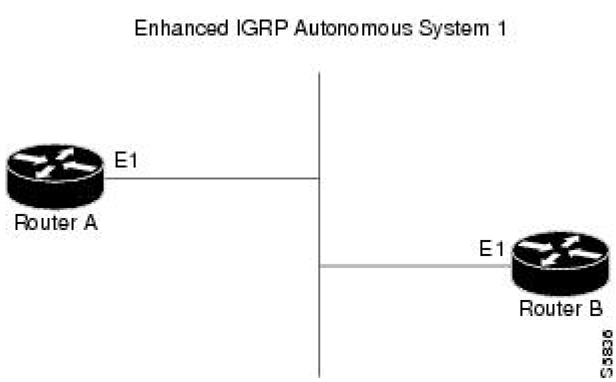
The following example shows how to enable MD5 authentication on EIGRP packets in autonomous system 1.

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 off the key chain. Key 1 will no longer be valid to be used for sending after December 4, 2006. After this date, key 2 would be used to send MD5 authentication, because it is valid until January 4, 2007.

The figure below shows the scenario.

Figure 7: EIGRP Route Authentication Scenario



Router A Configuration

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

Router B Configuration

```
Router> enable
Router(config)# configure terminal
Router(config)# router eigrp 1
Router(config-router)# exit
Router(config)# interface ethernet 1
Router(config-if)# ip authentication mode eigrp 1 md5
Router(config-if)# ip authentication key-chain eigrp 1 key2
Router(config-if)# exit
Router(config)# key chain key2
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string 0987654321
Router(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Router(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
Router(config-keychain-key)# exit
Router(config-keychain)# key 2
Router(config-keychain-key)# key-string 1234567890
Router(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Router(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 off the key chain. Key 1 will no longer be valid to be used for sending after December 4, 2006. After this date key 2 would be used to send MD5 authentication, because it is valid until January 4, 2007.

Example EIGRP Route Authentication--Named Configuration, page 63 shows the scenario.

Router A Configuration

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

```
Router(config-router-af)# af-interface ethernet0/0
Router(config-router-af-interface)# authentication key-chain SITE1
Router(config-router-af-interface)# authentication mode md5
Router(config-router-af-interface)# exit-af-interface
Router(config-router-af)# exit-address-family
Router(config-router)# exit
Router(config)# key chain SITE1
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string 0987654321
Router(config-keychain-key)# accept-lifetime 04:00:00 Dec 4 2006 infinite
Router(config-keychain-key)# send-lifetime 04:00:00 Dec 4 2006 infinite
Router(config-keychain-key)# exit
Router(config-keychain)# key 2
Router(config-keychain-key)# key-string 1234567890
Router(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Router(config-keychain-key)# send-lifetime 04:45:00 Jan 4 2007 infinite
```

Router B Configuration

```
Router> enable
Router# configure terminal
Router(config)# router eigrp virtual-name2
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# network 172.16.0.0
Router(config-router-af)# af-interface ethernet0/0
Router(config-router-af-interface)# authentication key-chain SITE2
Router(config-router-af-interface)# authentication mode md5
Router(config-router-af-interface)# exit-af-interface
Router(config-router-af)# exit-address-family
Router(config-router)# exit
Router(config)# key chain SITE2
Router(config-keychain)# key 1
Router(config-keychain-key)# key-string 0987654321
Router(config-keychain-key)# accept-lifetime 04:00:00 Jan 4 2007 infinite
Router(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

```
Router> enable
Router# configure terminal
Router(config)# router eigrp 1
Router(config-router)# exit
```

```
Router(config)# interface Ethernet0/1
Router(config-if)# ip hello-interval eigrp 109 10
Router(config-if)# ip hold-time eigrp 109 40
```

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

The following example shows how to adjust the interval between hello packets and the hold time in an EIGRP named configuration:

```
Router> enable
Router# configure terminal
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# af-interface ethernet0/0
Router(config-router-af-interface)# hello-interval 10
Router(config-router-af-interface)# hold-time 50
```

Example Disabling 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:

```
Router> enable
Router# configure terminal
Router(config)# router eigrp 1
Router(config-router)# exit
Router(config)# interface Ethernet0/1
Router(config-if)# no ip split-horizon eigrp 101
```

Example Disabling 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 will, by default, set the next-hop value to the local outbound interface address for routes that it is advertising, even when advertising those routes back out of the same interface where it learned them. 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 next-hop-self is primarily useful in DMVPN spoke-to-spoke topologies.

```
Router> enable
Router# configure terminal
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# af-interface ethernet0/0
Router(config-router-af-interface)# no split-horizon
Router(config-router-af-interface)# no next-hop-self
```

Example EIGRPStubRouting--AutonomousSystemConfiguration

A router that is configured as a stub with the **eigrp stub**command shares connected and summary routing information with all neighbor routers by default. 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.

eigrp stub Command: Example

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

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

eigrp stub connected static Command: Example

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

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

eigrp stub leak-map Command: Example

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

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

eigrp stub receive-only Command: Example

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

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

eigrp stub redistributed Command: Example

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

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

Example EIGRP Stub Routing--Named Configuration

A router that is configured as a stub with the **eigrp stub**command shares connected and summary routing information with all neighbor routers by default. 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.

eigrp stub Command: Example

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

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

eigrp stub connected static Command: Example

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

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

eigrp stub leak-map Command: Example

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

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

eigrp stub receive-only Command: Example

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

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 autonomous-system 4453
```

```
Router(config-router-af)# network 10.0.0.0
Router(config-router-af)# eigrp stub receive-only
```

eigrp stub redistributed Command: Example

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

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 autonomous-system 4453
Router(config-router-af)# network 10.0.0.0
Router(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 L2/L3 API and Tunable Metric for Mobile Adhoc Networks feature	Mobile Ad Hoc Networks for Router-to-Radio Communications module of the *Cisco IOS IP Mobility Configuration Guide*
EIGRP message authentication	EIGRP Message Authentication Configuration Example
EIGRP Technology Support	Enhanced Interior Gateway Routing Protocol
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol
IPv6 and EIGRP	Implementing EIGRP for IPv6 module of the *Cisco IOS IPv6 Configuration Guide*
Protocol-independent features that work with EIGRP	Configuring IP Routing Protocol-Independent Features module of the *Cisco IOS IP Routing : Protocol-Independent Configuration Guide*
Service Advertisement Framework	*Cisco IOS Service Advertisement Framework Configuration Guide*
Service Advertisement Framework commands	*Cisco IOS Service Advertisement Framework Command Reference*

Standards

Standard	Title
FIPS PUB 180-2	*SECURE HASH STANDARD (SHS)*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing standards 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

RFC	Title
RFC 1321	*The MD5 Message-Digest Algorithm*
RFC 2104	*HMAC: Keyed-Hashing for Message Authentication*

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 EIGRP Features

Feature Name	Releases	Feature Information
EIGRP IPv6 VRF-Lite	15.1(1)S 15.1(4)M	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.
EIGRP/SAF HMAC-SHA-256 Authentication	15.1(2)S	EIGRP packets will be authenticated using HMAC-SHA-256 message authentication codes. The HMAC algorithm takes as inputs the data to authenticate (that is, the EIGRP packet) and a shared secret key that is known to both the sender and the receiver, and outputs a 256-bit hash that will be used for authentication. If the hash value provided by the sender matches the hash value calculated by the receiver, the packet will be accepted by the receiver; otherwise it will be discarded. The following command was introduced or modified by this feature: **authentication mode (EIGRP)** .

Feature Name	Releases	Feature Information
Enhanced Interior Gateway Routing Protocol	11.2(1) 12.2(33)SRA 15.0(1)M 12.2(33)SRE 12.2(33)XNE 15.0(1)S	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, and IGRP is obsolete. The following commands were introduced or modified by this feature: **auto-summary (EIGRP)** , **clear ip eigrp neighbors**, **default-information**, **default-metric (EIGRP)**, **distance (EIGRP)**, **eigrp log-neighbor-changes**, **eigrp log-neighbor-warnings**, **eigrp router-id**, **ip bandwidth-percent eigrp**, **ip hello-interval eigrp**, **ip hold-time eigrp**, **ip next-hop-self eigrp**, **ip split-horizon eigrp**, **ip summary-address eigrp**, **metric maximum-hops**, **metric weights (EIGRP)**, **neighbor (EIGRP)**, **network (EIGRP)**, **offset-list (EIGRP)**, **router eigrp**, **set metric (EIGRP)**, **show ip eigrp accounting**, **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**, **summary-metric, timers active-time**, **traffic-share balanced**, **variance (EIGRP)**.
		In Cisco IOS Release 15.0(1)M, 12.2(33)SRE, and 12.2(33)XNE, the following commands were introduced or modified: **address-family (EIGRP)**, **af-interface**,

Feature Name	Releases	Feature Information
		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, match extcommunity, metric maximum-hops, metric weights, next-hop-self, offset-list (EIGRP), passive-interface (EIGRP), router 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, shutdown (address-family), split-horizon (EIGRP), summary-address (EIGRP), timers active-time, traffic-share balanced, variance (EIGRP)** In Cisco IOS Release 15.0(1)M, 12.2(33)SRE, and 12.2(33)XNE, the following commands were replaced: **clear ip eigrp vrf neighbors, eigrp interface, log-neighbor-warnings, show ip**

Feature Name	Releases	Feature Information
		eigrp vrf accounting, show ip eigrp vrf interfaces, show ip eigrp vrf neighbors, show ip eigrp vrf topology, show ip eigrp vrf traffic.
EIGRP Stub Routing	12.0(7)T 12.0(15)S 12.2(33)SRE 12.2(33)XNE 15.0(1)M 15.0(1)S	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 by this feature: **eigrp stub**.
IP Enhanced IGRP Route Authentication	11.3(1) 12.2(33)SRA 12.2(33)SRE 12.2(33)XNE 15.0(1)M 15.0(1)S	The IP Enhanced IGRP route authentication feature provides MD5 authentication of routing updates from the EIGRP routing protocol. The following commands were introduced or modified by this feature: **accept-lifetim e, ip authentication key-chain eigrp, ip authentication mode eigrp, key chain, key, key-string, se nd-lifetime**. In Cisco IOS Release 15.0(1)M, 12.2(33)SRE, and 12.2(33)XNE, the following commands were introduced or modified: **authentication mode (EIGRP)**, **authentication key-chain (EIGRP)**.

EIGRP MIB

The EIGRP MIB feature introduces an Enhanced Interior Gateway Routing Protocol (EIGRP) MIB in Cisco IOS software. This MIB is accessed through remote Simple Network Management Support (SNMP) software clients. This MIB provides full EIGRP support for GET requests and limited notification (TRAP) support for stuck-in-active (SIA) and neighbor authentication failure events.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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

- EIGRP MIB table objects are not visible via SNMP until an EIGRP routing process is enabled and an SNMP community string is configured on at least one router.
- Support for EIGRP notifications (TRAP) is not activated until a trap destination is configured.

Restrictions for EIGRP MIB

- EIGRP MIB support has not been implemented for the EIGRP Prefix Limit Support feature.

- EIGRP MIB support is available for IPv4 only.

Information About EIGRP MIB

EIGRP MIB Overview

The EIGRP MIB feature introduces EIGRP MIB support in Cisco IOS software. EIGRP routing processes that run over IPv4 are supported. The EIGRP MIB is accessed through remote SNMP software clients. MIB table objects are accessed as read-only through GET, GETINFO, GETMANY, GETNEXT, GETBULK, and SET requests. Counters for MIB table objects are cleared when the EIGRP routing process is reset or when the routing table is refreshed by entering the **clear ip route** or **clear ip eigrp**commands,or by entering **clear eigrp address-family**commands. Managed objects for all EIGRP routing processes are implemented as five table objects on a per-autonomous-system or per-Virtual-Private-Network (VPN) basis.

EIGRP VPN Table

The EIGRP VPN Table contains information regarding which VPNs are configured to run an EIGRP routing process. VPN routes are indexed by the VPN name and the EIGRP autonomous system number. The EIGRP VPN table object and the value populated for that object are described in the table below.

Table 4: VPN Table Object Description

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

EIGRP Traffic Statistics Table

The EIGRP Traffic Statistics Table contains counters and statistics for the specific types of EIGRP packets that are sent and the related collective information that is generated. The objects in this table are populated on a per-autonomous-system basis. Objects in this table are populated for adjacencies formed on all interfaces with an IP address that is configured under an EIGRP network statement. Traffic statistics table objects and the values populated for each object are described in the table below.

Table 5: EIGRP Traffic Statistics Options

Object	Description
cEigrpNbrCount	Total number of live neighbors. This table object is incremented or decremented as peering sessions are established or expired.
cEigrpHellosSent	Total number of transmitted hello packets. This table object is incremented as packets are transmitted.
cEigrpHellosRcvd	Total number of received hello packets. This table object is incremented as packets are received.
cEigrpUpdatesSent	Total number of transmitted routing update packets. This table object is incremented as packets are transmitted.
cEigrpUpdatesRcvd	Total number of received routing update packets. This table object is incremented as packets are received.
cEigrpQueriesSent	Total number of alternate route query packets transmitted. This table object is incremented as packets are transmitted.
cEigrpQueriesRcvd	Total number of alternate route query packets received. This table object is incremented as packets are received.
cEigrpRepliesSent	Total number of reply packets that are transmitted in response to received query packets. This table object is incremented as packets are transmitted.
cEigrpRepliesRcvd	Total number of reply packets that are received in response to transmitted query packets. This table object is incremented as packets are transmitted.
cEigrpAcksSent	Total number of acknowledgment packets that are transmitted in response to received update packets. This table object is incremented as packets are transmitted.
cEigrpAcksRcvd	Total number of acknowledgment packets that are received in response to transmitted update packets. This table object is incremented as packets are received.
cEigrpInputQHighMark	The highest number of packets that have been in the input queue. This table object is incremented only when the previous highest number is exceeded.

Object	Description
cEigrpInputQDrops	Total number of packets dropped from the input queue because the input queue was full. This tabled object is incremented each time a packet is dropped.
cEigrpSiaQueriesSent	Total number of query packets sent in response to a destination that is in a SIA state for a down peer. This table object is incremented each time an SIA query packet is sent.
cEigrpSiaQueriesRcvd	Total number of SIA query packets received from neighbors searching for an alternate path to a destination. This table object is incremented each time an SIA query packet is received.
cEigrpAsRouterIdType	The type of IP address that is used as the router ID. The value for this table object can be an IPv4 address.
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.
cEigrpTopoRoutes	Total number of EIGRP-derived routes in the topology table. This table object is incremented if a route is added or removed.
cEigrpHeadSerial	Internal sequencing number (serial) 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.
cEigrpNextSerial	The serial number applied to the next route in the sequence.
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.
cEigrpXmitDummies	Total number of temporary entries in the topology table. Dummies are internal entries and not transmitted in routing updates.

EIGRP Topology Table

The EIGRP Topology Table contains information regarding EIGRP routes received in updates and routes that are locally originated. EIGRP sends routing updates to and receives routing updates from adjacent routers to which peering relationships (adjacencies) have been formed. The objects in this table are populated on a per-topology-table-entry (route) basis. Topology table objects and the values populated for each object are described in the table below.

Table 6: Topology Table Object Descriptions

Object	Description
cEigrpActive	Displays the active status for 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 has gone into an active state. A value of 2 is displayed when a route is in a passive state (normal).
cEigrpStuckInActive	Displays the 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 an SIA state (no reply has been received for queries for alternate paths). SIA queries are transmitted when a route is placed in this state.
cEigrpDestSuccessors	Total number successors (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 is incremented each time a successor is added or removed.
cEigrpFdistance	The feasible (best) distance to a destination network. This value is used to calculate the feasible successor for a topology table entry.
cEigrpRouteOriginAddr	The protocol type of an IP address defined in the origin of the topology table entry.
cEigrpRouteOriginType	Displays 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.
cEigrpNextHopAddress Type	Displays the protocol type for the next-hop IP address for the route in a topology table entry.
cEigrpNextHopAddress	The next-hop IP address for a route in a topology table entry.
cEigrpNextHopInterface	The interface through which the next-hop IP address is reached to send traffic to the destination.
cEigrpDistance	The computed distance to the destination network entry from the local router.

Object	Description
cEigrpReportDistance	The computed distance to the destination network in the topology entry as reported by the originator of the route.

EIGRP Neighbor Table

The EIGRP Neighbor Table contains information about EIGRP neighbors to 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. Neighbor table objects and the values populated for each object are described in the table below.

Table 7: Neighbor Table Object Descriptions

Object	Description
cEigrpPeerAddrType	The protocol type of the remote source IP address used by the neighbor to establish the EIGRP adjacency with the local router.
cEigrpPeerAddr	The source IP address of the neighbor that was used to establish EIGRP adjacency with the local router.
cEigrpPeerInterface	The name of the local interface, through which the neighbor can be reached. This table object is populated on a per-neighbor basis.
cEigrpPeerIfIndex	The index of the local interface, through which this neighbor can be reached.
cEigrpHoldTime	The hold timer value for the adjacency with the neighbor. If this timer expires, the neighbor is declared down and removed from the neighbor table.
cEigrpUpTime	The length of time for which the EIGRP adjacency to the neighbor has been in an up state. The time period is displayed in hours:minutes:seconds.
cEigrpSrtt	The computed smooth round trip time (SRTT) for packets transmitted to and received from the neighbor.
cEigrpRto	The computed retransmission timeout (RTO) for the neighbor. The value for this table object is computed as an aggregate average of the time required for packet delivery. This table object is populated on a per-neighbor basis.
cEigrpPktsEnqueued	Total number of EIGRP packets (all types) currently queued for transmission to a neighbor.

Object	Description
	This table object is populated on a per-neighbor basis.
cEigrpLastSeq	The number of the last sequence number of a packet transmitted to a neighbor. This table object is incremented as the sequence number increases.
cEigrpVersion	The EIGRP version information reported by the remote neighbor. This table object is populated on a per-neighbor basis.
cEigrpRetrans	Cumulative number of packets retransmitted to the neighbor, while the neighbor is in an up state. This table object is populated on a per-neighbor basis.
cEigrpRetries	Total number of times an unacknowledged packet has been sent to a neighbor. This table object is populated on a per-neighbor basis.

EIGRP Interface Table

The EIGRP Interface Table contains information and statistics for each interface that EIGRP has been configured to run over. The objects in this table are populated on a per-interface basis. Interface table objects and the values populated for each object are described in the table below.

Table 8: EIGRP Interface Table Object Descriptions

Object	Description
cEigrpPeerCount	Total number of neighbor adjacencies formed through this interface.
cEigrpXmitReliableQ	Total number of packets waiting in the reliable transport transmission queue (acknowledgment is required) to be sent to a neighbor.
cEigrpXmitUnreliableQ	Total number of packets waiting in the unreliable transmission queue (no acknowledgment required).
cEigrpMeanSrtt	The computed SRTT for packets transmitted to and received from all neighbors on the interface.
cEigrpPacingReliable	The configured time interval (in milliseconds) between EIGRP packet transmissions on this interface when the reliable transport is used.
cEigrpPacingUnreliable	The configured time interval (in milliseconds) between EIGRP packet transmissions on this interface when the unreliable transport is used.

Object	Description
cEigrpMFlowTimer	The configured multicast flow control timer value (in milliseconds) for this interface.
cEigrpPendingRoutes	Total number of routing updates queued for transmission on this interface.
cEigrpHelloInterval	The configured time interval (in seconds) between Hello packet transmissions for this interface.
cEigrpXmitNextSerial	The serial number of the next packet that is queued for transmission on this interface.
cEigrpUMcasts	Total number of unreliable (no acknowledgment required) multicast packets transmitted on this interface.
cEigrpRMcasts	Total number of reliable (acknowledgment required) multicast packets transmitted on this interface.
cEigrpUUcasts	Total number of unreliable (no acknowledgment required) unicast packets transmitted on this interface.
cEigrpRUcasts	Total number of reliable (acknowledgment required) unicast packets transmitted on this interface.
cEigrpMcastExcept	The total number of EIGRP multicast exception transmissions that have occurred on this interface.
cEigrpCRpkts	Total number conditional-receive packets sent on this interface.
cEigrpAcksSuppressed	Total number of individual acknowledgment packets that have been suppressed and combined in an already enqueued outbound reliable packet on this interface.
cEigrpRetranSent	Total number of packet retransmissions sent on this interface.
cEigrpOOSrvcd	Total number of out-of-sequence packets received on this interface.
cEigrpAuthMode	The authentication mode configured for traffic that uses this interface. The value of 0 is displayed when no authentication is enabled. The value of 1 is displayed when message digest algorithm 5 (MD5) authentication is enabled.

Object	Description
cEigrpAuthKeyChain	The name of the authentication key chain configured on this interface. The key chain is a reference to which set of secret keys is to be accessed to determine which key string to use. The key-chain name is not the key string (password).

EIGRP Notifications

The EIGRP MIB provides limited notification (TRAP) support for SIA and neighbor authentication failure events. The **snmp-server enable traps eigrp** command is used to enable EIGRP notifications on a Cisco router. Support for TRAP events is not activated until a trap destination is configured with the **snmp-server host** command and a community string is defined with the **snmp-server community** command. EIGRP notifications are described in the table below.

Table 9: EIGRP Notifications

EIGRP Traps (Notifications)	Description
cEigrpAuthFailureEvent	When EIGRP MD5 authentication is enabled on any interface and neighbor adjacencies are formed, a notification is sent if any adjacency goes down as a result 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.
cEigrpRouteStuckInActive	During the query phase for a new route to a destination network, the route is placed in the active state (an alternate path is actively being sought) and a query packet is broadcast to the network. If no replies are received to the query, an SIA query packets are broadcast. If a reply is not received for the SIA queries, the neighbor adjacency is dropped, the route is declared SIA, and this notification is sent.

How to Enable EIGRP MIB

Enabling EIGRP MIB

Perform this task to enable an EIGRP MIB. This task specifies an SNMP server host, configures an SNMP community access string, and enables EIGRP notifications.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrrp**]
4. **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*]
5. **snmp-server enable traps eigrp**
6. **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.
Step 3	**snmp-server host** {*hostname* | *ip-address*} [**vrf** *vrf-name*] [**traps** | **informs**] [**version** {**1** | **2c** | **3** [**auth** | **noauth** | **priv**]}] *community-string* [**udp-port** *port*] [*notification-type*] [**vrrp**] **Example:** `Router(config)# snmp-server host 10.0.0.1 traps version 2c NETMANAGER eigrp`	Specifies the destination host or address for SNMP notifications.
Step 4	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number*] **Example:** `Router(config)# snmp-server community EIGRP1NET1A`	Configures a community access string to permit SNMP access to the local router by the remote SNMP software client. • Only IPv4 is supported in Cisco IOS Releases 12.3(14)T and 12.2(33)SRB.
Step 5	**snmp-server enable traps eigrp** **Example:** `Router(config)# snmp-server enable traps eigrp`	Enables SNMP support for EIGRP notifications. • Notifications can be configured for only SIA and neighbor authentication failure events.
Step 6	**end** **Example:** `Router(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Enabling EIGRP MIB

Example EIGRP MIB Configuration

In the following example, an SNMP server host is specified, a community string is configured, and support for EIGRP notifications is enabled:

```
Router(config)# snmp-server host 10.0.0.1 traps version 2c NETMANAGER eigrp
Router(config)# snmp-server community EIGRP1NET1A
Router(config)# snmp-server enable traps eigrp
```

Example EIGRP MIB Verification

In the following example, the local SNMP configuration is verified by entering the **show running-config**command:

```
Router# show running-config | include snmp
snmp-server community EIGRP1NET1A
snmp-server enable traps eigrp
snmp-server host 10.0.0.1 version 2c NETMANAGER
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
Basic EIGRP configuration tasks	Configuring EIGRP module
Troubleshooting SIA events	What Does the EIGRP DUAL-3-SIA Error Message Mean?
SNMP commands	*Cisco IOS Network Management Command Reference*
SNMP configuration tasks	Configuring SNMP Support module of the *Cisco IOS Network Management Configuration Guide*

Standards

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

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

RFCs

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

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 10: Feature Information for EIGRP MIB

Feature Name	Releases	Feature Information
EIGRP MIB	12.2(33)SRB 12.2(33)SXI4 12.2(33)XNE 12.3(14)T 15.0(1)M	The EIGRP MIB feature introduces an EIGRP MIB in Cisco IOS software. This MIB is accessed through remote Simple Network Management Support (SNMP) software clients. This MIB provides full EIGRP support for GET requests and limited notification (TRAP) support for stuck-in-active (SIA) and neighbor authentication failure events. The following commands were new or modified for this release: **snmp-server enable traps eigrp**, **snmp-server host**.

EIGRP MPLS VPN PE-CE Site of Origin

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces the capability to filter Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) traffic on a per-site basis for Enhanced Interior Gateway Routing Protocol (EIGRP) networks. Site of Origin (SoO) filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent transient routing loops from occurring in complex and mixed network topologies. This feature is designed to support the MPLS VPN Support for EIGRP Between Provider Edge (PE) and Customer Edge (CE) feature. Support for backdoor links is provided by this feature when a Cisco IOS release is implemented on PE routers that support EIGRP MPLS VPNs.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 MPLS VPN PE-CE Site of Origin

This document assumes that Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone). The following tasks will also need to be completed before you can configure this feature:

- This feature was introduced to support the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature and should be configured after the EIGRP MPLS VPN is created.

- All PE routers that are configured to support the EIGRP MPLS VPN must run a Cisco IOS release that provides support for the SoO extended community.

Restrictions for EIGRP MPLS VPN PE-CE Site of Origin

- If a VPN site is partitioned and the SoO extended community attribute is configured on a backdoor router interface, the backdoor link cannot be used as an alternate path to reach prefixes originated in other partitions of the same site.
- A unique SoO value must be configured for each individual VPN site. The same value must be configured on all provider edge and customer edge interfaces (if SoO is configured on the CE routers) that support the same VPN site.

Information About EIGRP MPLS VPN PE-CE Site of Origin

EIGRP MPLS VPN PE-CE Site of Origin Support Overview

The EIGRP MPLS VPN PE-CE Site of Origin feature introduces SoO support for EIGRP-to-BGP and BGP-to-EIGRP redistribution. The SoO extended community is a BGP extended community attribute that is used to identify routes that have originated from a site so that the readvertisement of that prefix back to the source site can be prevented. The SoO extended community uniquely identifies the site from which a PE router has learned a route. SoO support provides the capability to filter MPLS VPN traffic on a per-EIGRP-site basis. SoO filtering is configured at the interface level and is used to manage MPLS VPN traffic and to prevent routing loops from occurring in complex and mixed network topologies, such as EIGRP VPN sites that contain both VPN and backdoor links.

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

Site of Origin Support for Backdoor Links

The EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature introduces support for backdoor links. A backdoor link or a route is a connection that is configured outside of the VPN between a remote and main site; for example, a WAN leased line that connects a remote site to the corporate network. Backdoor links are typically used as back-up routes between EIGRP sites if the VPN link is down or not available. A metric is set on the backdoor link so that the route though the backdoor router is not selected unless there is a VPN link failure.

The SoO extended community is defined on the interface of the backdoor router. It identifies the local site ID, which should match the value that is used on the PE routers that support the same site. When the backdoor router receives an EIGRP update (or reply) from a neighbor across the backdoor link, the router checks the update for an SoO value. If the SoO value in the EIGRP update matches the SoO value on the local backdoor interface, the route is rejected and not added to the EIGRP topology table. This typically occurs when the route with the local SoO valued in the received EIGRP update was learned by the other VPN site and then advertised through the backdoor link by the backdoor router in the other VPN site. SoO filtering on the backdoor link prevents transient routing loops from occurring by filtering out EIGRP updates that contain routes that carry the local site ID.

Note If a VPN site is partitioned and the SoO extended community attribute is configured on a backdoor router interface, the backdoor link cannot be used as an alternate path to reach prefixes originated in other partitions of the same site.

If this feature is enabled on the PE routers and the backdoor routers in the customer sites, and SoO values are defined on both the PE and backdoor routers, both the PE and backdoor routers will support convergence between the VPN sites. The other routers in the customer sites need only propagate the SoO values carried by the routes, because the routes are forwarded to neighbors. These routers do not otherwise affect or support convergence beyond normal Diffusing Update Algorithm (DUAL) computations.

Router Interoperation with a Site of Origin Extended Community

The configuration of an SoO extended community allows routers that support the EIGRP MPLS VPN PE-CE Site of Origin feature to identify the site from which each route originated. When this feature is enabled, the EIGRP routing process on the PE or CE router checks each received route for the SoO extended community and filters based on the following conditions:

- A received route from BGP or a CE router contains a SoO value that matches the SoO value on the receiving interface.

If a route is received with an associated SoO value that matches the SoO value that is configured on the receiving interface, the route is filtered because it was learned from another PE router or from a backdoor link. This behavior is designed to prevent routing loops.

- A received route from a CE router is configured with an SoO value that does not match.

If a route is received with an associated SoO value that does not match the SoO value that is configured on the receiving interface, the route is added to the EIGRP topology table so that it can be redistributed into BGP.

If the route is already installed to the EIGRP topology table but is associated with a different SoO value, the SoO value from the topology table will be used when the route is redistributed into BGP.

- A received route from a CE router does not contain an SoO value.

If a route is received without a SoO value, the route is accepted into the EIGRP topology table, and the SoO value from the interface that is used to reach the next hop CE router is appended to the route before it is redistributed into BGP.

When BGP and EIGRP peers that support the SoO extended community receive these routes, they will also receive the associated SoO values and pass them to other BGP and EIGRP peers that support the SoO extended community. This filtering is designed to prevent transient routes from being relearned from the originating site, which prevents transient routing loops from occurring.

Redistribution of BGP VPN Routes That Carry the Site of Origin into EIGRP

When an EIGRP routing process on a PE router redistributes BGP VPN routes into an EIGRP topology table, EIGRP extracts the SoO value (if one is present) from the appended BGP extended community attributes and appends the SoO value to the route before adding it to the EIGRP topology table. EIGRP tests the SoO value for each route before sending updates to CE routers. Routes that are associated with SoO values that match the SoO value configured on the interface are filtered out before they are passed to the CE routers. When an EIGRP routing process receives routes that are associated with different SoO values, the SoO value is passed to the CE router and carried through the CE site.

BGP Cost Community Support for EIGRP MPLS VPN PE-CE Network Topologies

The BGP cost community is a nontransitive extended community attribute that is passed to internal BGP (iBGP) and confederation peers but not external BGP (eBGP) peers. The cost community feature allows you to customize the local route preference and influence the BGP best path selection process.

Before BGP cost community support for EIGRP MPLS VPN PE-CE network topologies was introduced, BGP preferred locally sourced routes over routes learned from BGP peers. Backdoor links in an EIGRP MPLS VPN topology were preferred by BGP when the backdoor link was learned first. (A backdoor link or a route is a connection that is configured outside of the VPN between a remote and main site; for example, a WAN leased line that connects a remote site to the corporate network).

The “prebest path” point of insertion (POI) was introduced in the BGP Cost Community feature to support mixed EIGRP VPN network topologies that contain VPN and backdoor links. This POI is applied automatically to EIGRP routes that are redistributed into BGP. The “prebest path” POI carries the EIGRP route type and metric. This POI influences the best path calculation process by influencing BGP to consider this POI before any other comparison step. No configuration is required. This feature is enabled automatically for EIGRP VPN sites when a Cisco IOS release that supports this feature is installed on the PE routers or the CE and backdoor router at the customer sites.

For more information about the BGP Cost Community feature, see to the BGP Cost Community module in the *Cisco IOS IP Routing: BGP Configuration Guide*.

Benefits of the EIGRP MPLS VPN PE-CE Site of Origin Support Feature

The configuration of the EIGRP MPLS VPN PE-CE Site of Origin Support feature introduces per-site VPN filtering, which improves support for complex topologies, such as MPLS VPNs with backdoor links, CE routers that are dual-homed to different PE routers, and PE routers that support CE routers from different sites within the same virtual routing and forwarding (VRF) instance.

How to Configure EIGRP MPLS VPN PE-CE Site of Origin Support

- Configuring the Site of Origin Extended Community, page 93
- Verifying the Configuration of the Site of Origin Extended Community, page 95

Configuring the Site of Origin Extended Community

The configuration of the SoO extended community allows MPLS VPN traffic to be filtered on a per-site basis. The SoO extended community is configured in an inbound BGP route map on the PE router and is applied to the interface. The SoO extended community can be applied to all exit points at the customer site for more specific filtering but must be configured on all interfaces of PE routers that provide VPN services to CE routers.

- Border Gateway Protocol (BGP) is configured in the network core (or the service provider backbone).
- Configure an EIGRP MPLS VPN before configuring this feature.
- All PE routers that are configured to support the EIGRP MPLS VPN must support the SoO extended community.
- A unique SoO value must be configured for each VPN site. The same value must be used on the interface of the PE router that connects to the CE router for each VPN site.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-name* {**permit** | **deny**} [*sequence-number*]
4. **set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*}
5. **exit**
6. **interface** *type number*
7. **ip vrf forwarding** *vrf-name*
8. **ip vrf sitemap** *route-map-name*
9. **ip address** *ip-address subnet-mask*
10. **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.
Step 3	**route-map** *map-name* {**permit** | **deny**} [*sequence-number*] **Example:** `Router(config)# route-map Site-of-Origin permit 10`	Enters route-map configuration mode and creates a route map. • The route map is created in this step so that SoO extended community can be applied.

	Command or Action	Purpose
Step 4	**set extcommunity** {**rt** *extended-community-value* [**additive**] | **soo** *extended-community-value*} **Example:** `Router(config-route-map)# set extcommunity soo 100:1` **Example:**	Sets BGP extended community attributes. • The **rt** keyword specifies the route target extended community attribute. • The **soo** keyword specifies the site of origin extended community attribute. • The *extended-community-value*argument specifies the value to be set. The value can be one of the following formats: ◦ autonomous-system-number: network-number ◦ ip-address: network-number The colon is used to separate the autonomous system number and network number or IP address and network number. • The **additive** keyword adds a route target to the existing route target list without replacing any existing route targets.
Step 5	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and enters global configuration mode.
Step 6	**interface** *type number* **Example:** `Router(config)# interface FastEthernet 0/0`	Enters interface configuration mode to configure the specified interface.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Router(config-if)# ip vrf forwarding VRF1`	Associates the VRF with an interface or subinterface. • The VRF name configured in this step should match the VRF name created for the EIGRP MPLS VPN with the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature.
Step 8	**ip vrf sitemap** *route-map-name* **Example:** `Router(config-if)# ip vrf sitemap Site-of-Origin`	Associates the VRF with an interface or subinterface. • The route map name configured in this step should match the route map name created to apply the SoO extended community in Step 3.
Step 9	**ip address** *ip-address subnet-mask* **Example:** `Router(config-if)# ip address 10.0.0.1 255.255.255.255`	Configures the IP address for the interface. • The IP address needs to be reconfigured after enabling VRF forwarding.
Step 10	**end**	Exits interface configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router(config-if)# end`	

- What to Do Next, page 95

What to Do Next

- For mixed EIGRP MPLS VPN network topologies that contain backdoor routes, the next task is to configure the “prebest path” cost community for backdoor routes.

Verifying the Configuration of the Site of Origin Extended Community

Use the following steps to verify the configuration of the SoO extended community attribute.

SUMMARY STEPS

1. **enable**
2. **show ip bgp vpnv4** {**all** | **rd** *route-distinguisher*| **vrf** *vrf-name*} [*ip-prefix* / *length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**]
3. **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**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip bgp vpnv4** {**all** | **rd** *route-distinguisher*| **vrf** *vrf-name*} [*ip-prefix* / *length* [**longer-prefixes**] [*output-modifiers*]] [*network-address* [*mask*] [**longer-prefixes**] [*output-modifiers*]] [**cidr-only**] [**community**] [**community-list**] [**dampened-paths**] [**filter-list**] [**flap-statistics**] [**inconsistent-as**] [**neighbors**] [**paths** [*line*]] [**peer-group**] [**quote-regexp**] [**regexp**] [**summary**] [**tags**] **Example:** `Router# show ip bgp vpnv4 all 10.0.0.1`	Displays VPN address information from the BGP table. • Use the **show ip bgp vpnv4** command with the **all** keyword to verify that the specified route has been configured with the SoO extended community attribute.
Step 3	**show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **topology** [*topology-name*] [*ip-address*] [**active**]	Displays entries in the EIGRP topology table.

	Command or Action	Purpose
	[**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**route-type**{**connected** | **external** | **internal** | **local** | **redistributed** | **summary** | **vpn**}] **Example:** `Router# show eigrp address-family ipv4 4453 topology 10.10.10.0/24`	

Configuration Examples for EIGRP MPLS VPN PE-CE SoO

Example Configuring the Site of Origin Extended Community

The following example, beginning in global configuration mode, configures the SoO extended community on an interface:

```
Router(config)# route-map Site-of-Origin permit 10
Router(config-route-map)# set extcommunity soo 100:1
Router(config-route-map)# exit

Router(config)# interface FastEthernet 0/0
Router(config-if)# ip vrf forwarding VRF1
Router(config-if)# ip vrf sitemap Site-of-Origin
Router(config-if)# ip address 10.0.0.1 255.255.255.255

Router(config-if)# end
```

Example Verifying the Site of Origin Extended Community

The following example shows VPN address information from the BGP table and verifies the configuration of the SoO extended community:

```
Router# show ip bgp vpnv4 all 10.0.0.1
BGP routing table entry for 100:1:10.0.0.1/32, version 6
Paths: (1 available, best #1, no table)
  Advertised to update-groups:
     1
  100 300
    192.168.0.2 from 192.168.0.2 (172.16.13.13)
      Origin incomplete, localpref 100, valid, external, best
      Extended Community: SOO:100:1
```

The following example shows how to display EIGRP metrics for specified internal services and external services:

```
Router# show eigrp address-family ipv4 4453 topology 10.10.10.0/24
EIGRP-IPv4 VR(virtual-name) Topology Entry for AS(4453)/ID(10.0.0.1) for 10.10.10.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 128256
  Descriptor Blocks:
  0.0.0.0 (Null0), from Connected, Send flag is 0x0
```

```
Composite metric is (128256/0), service is Internal
Vector metric:
  Minimum bandwidth is 10000000 Kbit
  Total delay is 5000 microseconds
  Reliability is 255/255
  Load is 1/255
  Minimum MTU is 1514
  Hop count is 0
  Originating router is 10.0.0.1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
BGP cost community feature and the "prebest path" point of insertion	BGP Cost Community module of the *Cisco IOS IP Routing: BGP Configuration Guide*
Cisco Express Forwarding (CEF) commands	*Cisco IOS IP Switching Command Reference*
CEF configuration tasks	Cisco Express Forwarding Overview module of the *Cisco IOS IP Switching Configuration Guide*
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
EIGRP configuration tasks	Configuring EIGRP
MPLS VPNs	*Cisco IOS Multiprotocol Label Switching Configuration Guide*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	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

RFC	Title
None	--

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 MPLS VPN PE-CE Site of Origin (SoO)

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 MPLS VPN PE-CE Site of Origin (SoO)

Feature Name	Releases	Feature Information
EIGRP MPLS VPN PE-CE Site of Origin (SoO)	12.0(27)S 12.2(18)SXE 12.2(28)SB 12.2(30)S 12.2(33)SRE 12.2(33)XNE 12.3(8)T 15.0(1)M	The EIGRP MPLS VPN PE-CE Site of Origin (SoO) feature introduces the capability to filter MPLS VPN traffic on a per-site basis for EIGRP networks. The following command was introduced or modified by this feature: **ip vrf sitemap**

Glossary

AFI--Address Family Identifier. Carries the identity of the network layer protocol that is associated with the network address.

backdoor link --A link connecting two backdoor routers.

backdoor router --A router that connects two or more sites, that are also connected to each other through an MPLS VPN EIGRP PE to CE links.

BGP --Border Gateway Protocol. An interdomain routing protocol that exchanges reachability information with other BGP systems. It is defined by RFC 1163, A Border Gateway Protocol (BGP). BGP supports CIDR and uses route aggregation mechanisms to reduce the size of routing tables.

Cost Community --An extended community attribute that can be inserted anywhere into the best path calculation.

customer edge (CE) router --A router that belongs to a customer network, that connects to a provider edge (PE) router to utilize MPLS VPN network services.

MBGP --multiprotocol BGP. An enhanced version of BGP that carries routing information for multiple network-layer protocols and IP multicast routes. It is defined in RFC 2858, Multiprotocol Extensions for BGP-4.

provider edge (PE) router --The PE router is the entry point into the service provider network. The PE router is typically deployed on the edge of the network and is administered by the service provider. The PE router is the redistribution point between EIGRP and BGP in PE to CE networking.

site --A collection of routers that have well-defined exit points to other “sites.”

site of origin (SoO) --A special purpose tag or attribute that identifies the site that injects a route into the network. This attribute is used for intersite filtering in MPLS VPN PE-to-CE topologies.

VPN --Virtual Private Network. Allows IP traffic to travel securely over public TCP/IP networks and the Internet by encapsulating and encrypting all IP packets. VPN uses a tunnel to encrypt all information at the IP level.

EIGRP Nonstop Forwarding (NSF) Awareness

Nonstop Forwarding (NSF) awareness allows an NSF-aware router to assist NSF-capable and NSF-aware neighbors to continue forwarding packets during a switchover operation or during a well-known failure condition. The EIGRP Nonstop Forwarding Awareness feature allows an NSF-aware router that is running Enhanced Interior Gateway Routing Protocol (EIGRP) to forward packets along routes known to a router performing a switchover operation or in a well-known failure condition. This capability allows the EIGRP peers of the failing router to retain the routing information that it has advertised and to continue using this information until the failed router resumes normal operation and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Nonstop Forwarding Awareness

- Your network is configured to run EIGRP.
- An NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in an NSF restart operation.
- A version of Cisco IOS that supports NSF awareness or NSF capabilities must be installed.

Restrictions for EIGRP Nonstop Forwarding Awareness

- All neighboring devices participating in EIGRP NSF must be NSF-capable or NSF-aware.
- EIGRP NSF awareness does not support two neighbors performing an NSF restart operation at the same time. However, both neighbors can reestablish peering sessions after the NSF restart operation is completed.

Information About EIGRP Nonstop Forwarding Awareness

Cisco NSF Routing and Forwarding Operation

Cisco NSF is supported by the BGP, EIGRP, OSPF, and IS-IS protocols for routing and by Cisco Express Forwarding (CEF) for forwarding. Of the routing protocols, BGP, OSPF, and IS-IS have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices. The IS-IS protocol can be configured to use state information that has been synchronized between the active and the standby route processor (RP) to recover route information following a switchover instead of information received from peer devices.

In this document, a networking device that is NSF-aware is running NSF-compatible software. A device that is NSF-capable has been configured to support NSF; therefore, the device rebuilds routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the routing information base (RIB) tables. After the routing protocols have converged, CEF updates the forwarding information base (FIB) table and removes stale route entries. CEF, in turn, updates the line cards with the new FIB information.

Cisco Express Forwarding

In a Cisco networking device, CEF provides packet forwarding, a key element of NSF. CEF maintains the FIB and uses the FIB information that was current at the time of a switchover to continue forwarding packets during the switchover. NSF helps to reduce traffic interruption during the switchover.

During normal NSF operation, CEF on the active RP synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the standby RP. Upon switchover of the active RP, the standby RP initially has FIB and adjacency databases that are mirror images of those that were current on the active RP. For platforms with intelligent line cards, the line cards will maintain the current forwarding information over a switchover; for platforms with forwarding engines, CEF will keep the forwarding engine on the standby RP current with changes that are sent to it by CEF on the active RP. In this way, the line cards or forwarding engines will be able to continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates in turn cause prefix-by-prefix updates for CEF, which it uses to update the FIB and adjacency databases. Existing and new entries will receive the new version ("epoch") number, indicating that they have been refreshed. The forwarding information is updated on the line cards or forwarding engine during convergence. The RP signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information

The routing protocols run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables.

Note For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

EIGRP Nonstop Forwarding Awareness

NSF awareness allows a router that is running EIGRP to assist NSF-capable neighbors to continue forwarding packets during a switchover operation or well-known failure condition. The EIGRP Nonstop Forwarding Awareness feature provides EIGRP with the capability to detect a neighbor that is undergoing an NSF restart event (RP switchover operation) or well-known failure condition, maintain the peering session with this neighbor, retain known routes, and continue to forward packets for these routes. The deployment of EIGRP NSF awareness can minimize the effects of the following:

- Well-known failure conditions (for example, a stuck-in-active event)
- Unexpected events (for example, an RP switchover operation)
- Scheduled events (for example, a hitless software upgrade)

EIGRP NSF awareness is enabled by default and is transparent to the network operator and EIGRP peers that do not support NSF capabilities.

Note An NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in an NSF restart operation.

EIGRP NSF Capable and NSF Aware Interoperation

EIGRP NSF capabilities are exchanged by EIGRP peers in hello packets. An NSF-capable router notifies its neighbors that an NSF restart operation has started by setting the restart (RS) bit in a hello packet. When an NSF-aware router receives notification from an NSF-capable neighbor that an NSF-restart operation is in progress, both routers immediately exchange their topology tables. The NSF-aware router sends an end-of-table (EOT) update packet when the transmission of its topology table is complete. The NSF-aware router then performs the following actions to assist the NSF-capable router:

- Expires the EIGRP hello hold timer to reduce the time interval set for hello packet generation and transmission. This allows the NSF-aware router to reply to the NSF-capable router more quickly and reduces the amount of time required for the NSF-capable router to rediscover neighbors and rebuild the topology table.

- Starts the route-hold timer. This timer is used to set the period of time that the NSF-aware router will hold known routes for the NSF-capable neighbor. This timer is configured with the **timers graceful-restart purge-time**command. The default time period is 240 seconds.
- Notes in the peer list that the NSF-capable neighbor is restarting, maintains adjacency, and holds known routes for the NSF-capable neighbor until the neighbor signals that it is ready for the NSF-aware router to send its topology table or the route-hold timer expires. If the route-hold timer expires on the NSF-aware router, it discards held routes and treats the NSF-capable router as a new router joining the network and reestablishing adjacency accordingly.

When the switchover operation is complete, the NSF-capable router notifies its neighbors that it has reconverged and has received all of their topology tables by sending an EOT update packet to the assisting routers. The NSF-capable router then returns to normal operation. The NSF-aware router looks for alternate paths (go active) for any routes that are not refreshed by the NSF-capable (restarting) router. The NSF-aware router returns to normal operation. If all paths are refreshed by the NSF-capable router, the NSF-aware router immediately returns to normal operation.

Non-NSF Aware EIGRP Neighbors

NSF-aware routers are completely compatible with non-NSF aware or non-NSF capable neighbors in an EIGRP network. A non-NSF aware neighbor ignores NSF capabilities and resets the adjacency when they are received.

The NSF-capable router drops any queries that are received while converging to minimize the number of transient routes that are sent to neighbors. The NSF-capable router, however, still acknowledges these queries to prevent these neighbors from resetting adjacency.

Note An NSF-aware router continues to send queries to an NSF-capable router that is converging after a switchover, effectively extending the time before a stuck-in-active (SIA) condition can occur.

EIGRP NSF Route-Hold Timers

The route-hold timer is configurable, which allows you to tune network performance and avoid undesired conditions such as “black holing” routes if the switchover operation is lengthy. When the timer expires, the NSF-aware router scans the topology table and discards stale routes, allowing EIGRP peers to find alternate routes instead of waiting during a long switchover operation.

The route-hold timer is configured with the **timers graceful-restart purge-time** router configuration command. The default time period for the route-hold timer is 240 seconds. The configurable range is from 10 to 300 seconds.

How to Modify and Maintain EIGRP Nonstop Forwarding Awareness

Adjusting NSF Route-Hold Timers

Perform the following steps to configure NSF route-hold timers on an NSF-aware router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** {*autonomous-system-number* | *virtual-instance-name*}
4. **address-family ipv4** [**multicast**][**unicast**][**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **timers graceful-restart purge-time** *seconds*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** {*autonomous-system-number* | *virtual-instance-name*} **Example:** `Router(config)# router eigrp 101`	Enters router configuration mode and creates an EIGRP routing process.
Step 4	**address-family ipv4** [**multicast**][**unicast**][**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf vrf1`	(Optional) Enters address family configuration mode and creates a session for the VRF.
Step 5	**timers graceful-restart purge-time** *seconds* **Example:** `Router(config-router)# timers graceful-restart purge-time 120`	Sets the route-hold timer to determine how long an NSF-aware router that is running EIGRP will hold routes for an inactive peer. **Note** The **timers nsf route-hold** command was replaced with the **timers graceful-restart purge-time** command in Cisco IOS Release 15.0(1)M and 12.2(33)SRE.
Step 6	**exit**	Exits router configuration mode and enters global configuration mode.

	Command or Action	Purpose

	Example:
	`Router(config-router)# exit`
	Example:
	`Router(config)#`	

• Troubleshooting Tips, page 106

Troubleshooting Tips

Neighbor adjacencies are maintained during NSF switchover operations. If adjacencies between NSF-capable and NSF-aware neighbors are being reset too often, the route-hold timers may need to be adjusted. The **show ip eigrp neighbors detail** command can be used to help determine if the route-hold timer value should be set to a longer time period. The time that adjacency is established with specific neighbors is displayed in the output. This time indicates if adjacencies are being maintained or reset and when the last time that specific neighbors were restarted.

Monitoring EIGRP NSF Debug Events and Notifications

Perform the following steps to monitor EIGRP NSF debug events and notifications on an NSF-aware router.

The **debug eigrp nsf** and **debug ip eigrp notifications** commands are provided together for example purposes only. You do not have to issue these commands together or in the same session as there are differences in the information that is provided.

Debugging processes are heavy users of CPU resources. Debug commands should not be used in a production network unless you are troubleshooting a problem.

SUMMARY STEPS

1. **enable**
2. **debug eigrp nsf**
3. **debug ip eigrp notifications**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug eigrp nsf**	Displays NSF notifications and information about NSF events in an EIGRP network on the console of the router.

	Command or Action	Purpose
	Example: `Router# debug eigrp nsf`	
Step 3	**debug ip eigrp notifications** **Example:** `Router# debug ip eigrp notifications`	Displays EIGRP events and notifications in the console of the router. The output from this command also includes NSF notifications and information about NSF events.

Verifying the Local Configuration of EIGRP NSF Awareness

Perform the following steps to verify NSF-awareness configuration on a router that is running EIGRP.

SUMMARY STEPS

1. **enable**
2. **show ip protocols**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables higher privilege levels, such as privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip protocols** **Example:** `Router# show ip protocols`	Displays the parameters and current state of the active routing protocol process. The output of this command can be used to verify EIGRP NSF-awareness.

Configuration Examples for EIGRP Nonstop Forwarding Awareness

- Example EIGRP Graceful-Restart Purge-Time Timer Configuration, page 107
- Example Monitoring EIGRP NSF Debug Events and Notifications Configuration, page 108
- Example Verifying Local Configuration of EIGRP NSF Awareness, page 108

Example EIGRP Graceful-Restart Purge-Time Timer Configuration

The **timers graceful-restart purge-time** command is used to set the route-hold timer that determines how long an NSF-aware router that is running EIGRP will hold routes for an inactive peer. The following example shows how to set the route-hold timer to two minutes:

```
Router(config-router)# timers graceful-restart purge-time 120
```

Example Monitoring EIGRP NSF Debug Events and Notifications Configuration

The following example output shows that an NSF-aware router has received a restart notification. The NSF-aware router waits for EOT to be sent from the restarting (NSF-capable) neighbor.

```
Router# debug ip eigrp notifications
*Oct  4 11:39:18.092:EIGRP:NSF:AS2. Rec RS update from 135.100.10.1,
00:00:00. Wait for EOT.
*Oct  4 11:39:18.092:%DUAL-5-NBRCHANGE:IP-EIGRP(0) 2:Neighbor
135.100.10.1 (POS3/0) is up:peer NSF restarted
*Sep 23 18:49:07.578: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 1.1.2.1 (Ethernet1/0)
is resync: peer graceful-restart
```

Example Verifying Local Configuration of EIGRP NSF Awareness

The following is example output from the **show ip protocols** command. The output from this command can be used to verify the local configuration of EIGRP NSF awareness. The output shows that the router is NSF-aware and that the route-hold timer is set to 240 seconds, which is the default value.

```
Router# show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is “eigrp 101”
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 101
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.4.9.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

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*

Related Topic	Document Title
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 Nonstop Forwarding Awareness

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 EIGRP Nonstop Forwarding Awareness

Feature Name	Releases	Feature Information
EIGRP Nonstop Forwarding (NSF) Awareness	12.2(15)T 12.2(33)SRE 15.0(1)M 15.0(1)S	The EIGRP Nonstop Forwarding Awareness feature allows an NSF-aware router running EIGRP to forward packets along routes known to a router performing a switchover operation or in a well-known failure condition. The following commands were introduced or modified: **debug eigrp nsf**, **debug ip eigrp notifications**, **show ip eigrp neighbors**, **show ip protocols**, **timers graceful-restart purge-time**, **timers nsf route-hold**.

EIGRP Prefix Limit Support

The EIGRP Prefix Limit Support feature introduces the capability to limit the number of prefixes per VRF that are accepted from a specific peer or to limit all prefixes that are accepted by an Enhanced Interior Gateway Routing Protocol (EIGRP) process through peering and redistribution. This feature is designed to protect the local router from external misconfiguration that can negatively impact local system resources; for example, a peer that is misconfigured to redistribute full Border Gateway Protocol (BGP) routing tables into EIGRP. This feature is enabled under the IPv4 VRF address family and can be configured to support the MPLS VPN Support for EIGRP Between Provider Edge and Customer Edge feature.

For more information about EIGRP MPLS VPN configuration, refer to the EIGRP MPLS VPN PE-CE Site of Origin module.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Prefix Limit Support

- Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) services have been configured between the Provider Edge (PE) routers and the customer edge (CE) routers at the customer sites.

Restrictions for EIGRP Prefix Limit Support

- This feature is supported only under the IPv4 VRF address family and can be used only to limit the number of prefixes that are accepted through a VRF.
- The EIGRP Prefix Limiting Support feature is enabled only under the IPv4 VRF address-family. A peer that is configured to send too many prefixes or a peer that rapidly advertises and then withdraws prefixes can cause instability in the network. This feature can be configured to automatically reestablish a disabled peering session at the default or user-defined time interval or when the maximum-prefix limit is not exceeded. However, the configuration of this feature alone cannot change or correct a peer that is sending an excessive number of prefixes. If the maximum-prefix limit is exceeded, you will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer.

Information About EIGRP Prefix Limit Support

- Misconfigured VPN Peers, page 112
- EIGRP Prefix Limit Support Overview, page 112
- Warning-Only Mode, page 113
- Restart Reset and Dampening Timers and Counters, page 113

Misconfigured VPN Peers

In MPLS VPNs, the number of routes that are permitted in the VPN routing and forwarding instance (VRF) is configured with the **maximum routes** VRF configuration command. However, limiting the number routes permitted in the VPN does not protect the local router from a misconfigured peer that sends an excessive number of routes or prefixes. This type of external misconfiguration can have a negative effect on the local router by consuming all available system resources (CPU and memory) in processing prefix updates. This type of misconfiguration can occur on a peer that is not within the control of the local administrator.

EIGRP Prefix Limit Support Overview

The EIGRP Prefix Limit Support feature provides the ability to configure a limit on the number of prefixes that are accepted from EIGRP peers or learned through redistribution. This feature can be configured on per-peer or per-process basis and can be configured for all peers and processes. This feature is designed to protect the local router from misconfigured external peers by limiting the amount of system resources that can be consumed to process prefix updates.

- Protecting the Router from External Peers, page 112
- Limiting the Number of Redistributed Prefixes, page 113
- Protecting the Router at the EIGRP Process Level, page 113

Protecting the Router from External Peers

This feature can be configured to protect an individual peering session or protect all peering sessions. When this feature is enabled and the maximum-prefix limit has been exceeded, the router will tear down the peering session, clear all routes that were learned from the peer, and then place the peer in a penalty state for the default or user-defined time period. After the penalty time period expires, normal peering will be reestablished.

Limiting the Number of Redistributed Prefixes

This feature can be configured to limit the number of prefixes that are accepted into the EIGRP topology table through redistribution from the Routing Information Base (RIB). All sources of redistribution are processed cumulatively. When the maximum-prefix limit is exceeded, all routes learned through redistribution are discarded and redistribution is suspended for the default or user-defined time period. After the penalty time period expires, normal redistribution will occur.

Protecting the Router at the EIGRP Process Level

This feature can be configured to protect the router at the EIGRP process level. When this feature is configured at the EIGRP process level, the maximum-prefix limit is applied to all peering sessions and to route redistribution. When the maximum-prefix limit is exceeded, all sessions with the remote peers are torn down, all routes learned from remote peers are removed from the topology and routing tables, all routes learned through redistribution are discarded, and redistribution and peering are suspended for the default or user-defined time period.

Warning-Only Mode

The EIGRP Prefix Limit Support feature has two modes of operation. This feature can control peering and redistribution per default and user-defined values or this feature can operate in warning-only mode. In warning-only mode the router will monitor the number of prefixes learned through peering and/or redistribution but will not take any action when the maximum-prefix limit is exceeded. Warning-only mode is activated only when the **warning-only** keyword is configured for any of the maximum-prefix limit commands. Only syslog messages are generated when this mode of operation is enabled. Syslog messages can be sent to a syslog server or printed in the console. These messages can be buffered or rate limited per standard Cisco IOS system logging configuration options. For more information about system logging in Cisco IOS software, refer to the Troubleshooting and Fault Managemen of the *Cisco IOS Network Management Configuration Guide*.

Restart Reset and Dampening Timers and Counters

The EIGRP Prefix Limit Support feature provides two user-configurable timers, a restart counter, and a dampening mechanism. When the maximum-prefix limit is exceeded, peering and/or redistribution is suspended for a default or user-defined time period. If the maximum-prefix limit is exceeded too often, redistribution and/or peering will be suspended until manual intervention is taken.

- Restart Timer, page 113
- Restart Counter, page 114
- Reset Timer, page 114
- Dampening Mechanism, page 114

Restart Timer

The restart timer determines how long the router will wait to form an adjacency or accept redistributed routes from the RIB after the maximum-prefix limit has been exceeded. The default restart-time period is 5 minutes.

Restart Counter

The restart counter determines the number of times a peering session can be automatically reestablished after the peering session has been torn down or after the a redistributed routes have been cleared and relearned because the maximum-prefix limit has been exceeded. The default restart-count limit is three.

Caution After the restart count limit has been crossed, you will need to enter the **clear ip route ***, **clear ip eigrp neighbor**, or **clear eigrp address-family neighbor**command to restore normal peering and redistribution.

Reset Timer

The reset timer is used to configure the router to reset the restart count to 0 after the default or configured reset-time period has expired. This timer is designed to provide administrator with control over long-and medium-term accumulated penalties. The default reset-time period is 15 minutes.

Dampening Mechanism

The dampening mechanism is used to apply an exponential decay penalty to the restart-time period each time the maximum-prefix limit is exceeded. The half-life for the decay penalty is 150 percent of the default or user-defined restart-time value in minutes. This mechanism is designed to identify and suppress unstable peers. It is disabled by default.

How to Configure the Maximum-Prefix Limit

Configuring the Maximum Number of Prefix Accepted from Peering Sessions Autonomous System Configuration

The maximum-prefix limit can be configured for all peering sessions or individual peering sessions with the **neighbor maximum-prefix**(EIGRP) command. When the maximum-prefix limit is exceeded, the session with the remote peer is torn down and all routes learned from the remote peer are removed from the topology and routing tables. The maximum-prefix limit that can be configured is limited only by the available system resources on the router.

Note

In EIGRP, **neighbor** commands have been used traditionally to configure static neighbors. In the context of this feature, however, the **neighbor maximum-prefix** command can be used to configure the maximum-prefix limit for both statically configured and dynamically discovered neighbors.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note

- This task can be configured only in IPv4 VRF address family configuration mode.
- When you configure the **neighbor maximum-prefix** command to protect a single peering session, only the maximum-prefix limit, the percentage threshold, the warning-only configuration options can be configured. Session dampening, restart, and reset timers are configured on a global basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**multicast**][**unicast**][**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **description** *text*
6. **neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*] [**warning-only**]
7. **neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
8. **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	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**multicast**][**unicast**][**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf vrf1`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **description** *text* **Example:** `Router(config-router-af)# neighbor 172.16.2.3 description peer with example.com`	(Optional) Associates a description with a neighbor.
Step 6	**neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*] [**warning-only**] **Example:** `Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from the specified EIGRP neighbor.
Step 7	**neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# neighbor maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from all EIGRP neighbors.
Step 8	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

• Troubleshooting Tips, page 116

Troubleshooting Tips

If an individual peer or all peers have exceeded the maximum-prefix limit the same number of times as the default or user-defined restart-count value, the individual session or all sessions will need to be manually reset with the **clear ip route*** or **clear ip eigrp neighbor** command before normal peering can be reestablished.

Configuring the Maximum Number of Prefixes Accepted from Peering Sessions Named Configuration

The maximum-prefix limit can be configured for all peering sessions or individual peering sessions with the **neighbor maximum-prefix** (EIGRP) command. When the maximum-prefix limit is exceeded, the session with the remote peer is torn down and all routes learned from the remote peer are removed from the topology and routing tables. The maximum-prefix limit that can be configured is limited only by the available system resources on the router.

Note In EIGRP, **neighbor** commands have been used traditionally to configure static neighbors. In the context of this feature, however, the **neighbor maximum-prefix** command can be used to configure the maximum-prefix limit for both statically configured and dynamically discovered neighbors.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note

- This task can be configured only in IPv4 VRF address family configuration mode.
- When you configure the **neighbor maximum-prefix** command to protect a single peering session, only the maximum-prefix limit, the percentage threshold, and the warning-only configuration options can be configured. Session dampening, restart, and reset timers are configured on a global basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **neighbor** {*ip-address* | *peer-group-name*} **description** *text*
6. **neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*][**warning-only**]
7. **neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
8. **exit-address-family**

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.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf RED autonomous-system 45000`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**neighbor** {*ip-address* | *peer-group-name*} **description** *text* **Example:** `Router(config-router-af)# neighbor 172.16.2.3 description peer with example.com`	(Optional) Associates a description with a neighbor.
Step 6	**neighbor** *ip-address* **maximum-prefix** *maximum* [*threshold*][**warning-only**] **Example:** `Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from the specified EIGRP neighbor.
Step 7	**neighbor maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# neighbor maximum-prefix 10000 80 warning-only`	Limits the number of prefixes that are accepted from all EIGRP neighbors.
Step 8	**exit-address-family**	Exits address family configuration mode.

	Command or Action	Purpose
	Example: `Router(config-router-af)# exit-address-family`	

• Troubleshooting Tips, page 119

Troubleshooting Tips

If an individual peer or all peers have exceeded the maximum-prefix limit the same number of times as the default or user-defined restart-count value, the individual session or all sessions will need to be manually reset with the **clear ip route*** or **clear eigrp address-family neighbors** command before normal peering can be reestablished.

Configuring the Maximum Number of Prefixes Learned Through Redistribution Autonomous System Configuration

The maximum-prefix limit can be configured for prefixes learned through redistribution with the **redistribute maximum-prefix** (EIGRP) command. When the maximum-prefix limit is exceeded, all routes learned from the RIB will be discarded and redistribution will be suspended for the default or user-defined time period. The maximum-prefix limit that can be configured for redistributed prefixes is limited only by the available system resources on the router.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
5. **redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
6. **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.
Step 3	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**unicast**] **vrf** *vrf-name* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# redistribute maximum-prefix 10000 80 reset-time 10 restart 2`	Limits the number of prefixes redistributed into an EIGRP process.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address family configuration mode and enters privileged EXEC mode.

• Troubleshooting Tips, page 120

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear ip eigrp neighbors** command will need to be entered before normal redistribution will occur.

Configuring the Maximum Number of Prefixes Learned Through Redistribution Named Configuration

The maximum-prefix limit can be configured for prefixes learned through redistribution with the **redistribute maximum-prefix**(EIGRP) command. When the maximum-prefix limit is exceeded, all routes learned from the RIB will be discarded and redistribution will be suspended for the default or user-defined time period. The maximum-prefix limit that can be configured for redistributed prefixes is limited only by the available system resources on the router.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

VRFs have been created and configured. EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family topology configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **network** *ip-address* [*wildcard-mask*]
6. **topology** {**base** | *topology-name* **tid** *number*}
7. **redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
8. **exit-af-topology**

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	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Router(config-router-af)# network 172.16.0.0`	Specifies the network for an EIGRP address family routing process.
Step 6	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(config-router-af)# topology base`	Configures an EIGRP process to route traffic under the specified topology instance and enters address-family topology configuration mode.
Step 7	**redistribute maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af-topology)# redistribute maximum-prefix 10000 80 reset-time 10 restart 2`	Limits the number of prefixes redistributed into an EIGRP process.
Step 8	**exit-af-topology** **Example:** `Router(config-router-af-topology)# exit-af-topology`	Exits address family topology configuration mode.

• Troubleshooting Tips, page 122

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear eigrp address-family neighbors** command will need to be entered before normal redistribution will occur.

Configuring the Maximum-Prefix Limit for an EIGRP Process Autonomous System Configuration

The maximum-prefix limit can be configured for an EIGRP process to limit the number prefixes that are accepted from all sources. This task is configured with the **maximum-prefix**command. When the maximum-prefix limit is exceeded, sessions with the remote peers are brought down and all routes learned from remote peers are removed from the topology and routing tables. Also, all routes learned from the RIB are discarded and redistribution is suspended for the default or user-defined time period.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

Note This task can be configured only in IPv4 VRF address family configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *as-number*
4. **address-family ipv4** [**unicast**] **vrf** *vrf-name*
5. **maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
6. **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.
Step 3	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Enters router configuration mode and creates an EIGRP routing process. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**unicast**] **vrf** *vrf-name*	Enters address family configuration mode and creates a session for the VRF.

	Command or Action	Purpose
	Example: `Router(config-router)# address-family ipv4 vrf RED`	
Step 5	**maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af)# maximum-prefix 10000 80 reset-time 10 restart 2 warning-only`	Limits the number of prefixes that are accepted under an address family by an EIGRP process. • The example configures a maximum-prefix limit of 10,000 prefixes, a reset time period of 10 minutes, a warning message to be displayed at 80 percent of the maximum-prefix limit, and a restart time period of 2 minutes.
Step 6	**end** **Example:** `Router(config-router-af)# end`	Exits address-family configuration mode and enters privileged EXEC mode.

• Troubleshooting Tips, page 124

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear ip eigrp neighbors** command will need to be entered before normal redistribution will occur.

Configuring the Maximum-Prefix Limit for an EIGRP Process Named Configuration

The maximum-prefix limit can be configured for an EIGRP process to limit the number prefixes that are accepted from all sources. This task is configured with the **maximum-prefix**command. When the maximum-prefix limit is exceeded, sessions with the remote peers are brought down and all routes learned from remote peers are removed from the topology and routing tables. Also, all routes learned from the RIB are discarded and redistribution is suspended for the default or user-defined time period.

Default or user-defined restart, restart-count, and reset-time values for the process-level configuration of this feature, configured with the **maximum-prefix** command, are inherited by the **redistribute maximum-prefix** and **neighbor maximum-prefix** command configurations by default. If a single peer is configured with the **neighbor maximum-prefix** command, a process-level configuration or a configuration that is applied to all neighbors will be inherited.

- VRFs have been created and configured.
- EIGRP peering is established through the MPLS VPN.

> **Note** This task can be configured only in IPv4 VRF address family topology configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *virtual-instance-name*
4. **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*
5. **topology** {**base** | *topology-name* **tid** *number*}
6. **maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**]
7. **exit-af-topology**
8. **show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting**

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.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Creates an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number* **Example:** `Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000`	Enters address family configuration mode and creates a session for the VRF.
Step 5	**topology** {**base** | *topology-name* **tid** *number*}	Configures an EIGRP process to route traffic under the specified topology instance and enters address family topology configuration mode.

	Command or Action	Purpose
	Example: `Router(config-router-af)# topology base`	
Step 6	**maximum-prefix** *maximum* [*threshold*] [[**dampened**] [**reset-time** *minutes*] [**restart** *minutes*] [**restart-count** *number*] | **warning-only**] **Example:** `Router(config-router-af-topology)# maximum-prefix 10000 80 reset-time 10 restart 2 warning-only`	Limits the number of prefixes that are accepted under an address family by an EIGRP process. • The example configures a maximum-prefix limit of 10,000 prefixes, a reset time period of 10 minutes, a warning message to be displayed at 80 percent of the maximum-prefix limit, and a restart time period of 2 minutes.
Step 7	**exit-af-topology** **Example:** `Router(config-router-af-topology)# exit-af-topology`	Exits address family topology configuration mode.
Step 8	**show eigrp address-family** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] [*autonomous-system-number*] [**multicast**] **accounting** **Example:** `Router# show eigrp address-family ipv4 22 accounting`	(Optional) Displays prefix accounting information for EIGRP processes. **Note** Connected and summary routes are not listed individually in the output from this **show** command but are counted in the total aggregate count per process.

Example

The following is sample output from the **show eigrp address-family accounting** command:

```
Router# show eigrp address-family ipv4 22 accounting
EIGRP-IPv4 VR(saf) Accounting for AS(22)/ID(10.0.0.1)
Total Prefix Count: 3  States: A-Adjacency, P-Pending, D-Down
State Address/Source    Interface       Prefix   Restart  Restart/
                                        Count     Count   Reset(s)
 A    10.0.0.2          Et0/0              2         0        0
 P    10.0.2.4          Se2/0              0         2        114
 D    10.0.1.3          Et0/0              0         3        0
```

• Troubleshooting Tips, page 126

Troubleshooting Tips

If the maximum-prefix limit has been exceeded for redistribution the same number of times as the default or user-defined restart-count value, the **clear ip route *** or **clear eigrp address-family neighbors**command will need to be entered before normal redistribution will occur.

Configuration Examples for Configuring the Maximum-Prefix Limit

Example Configuring the Maximum-Prefix Limit for a Single Peer--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for a single peer. The maximum limit is set to 1000 prefixes, and the warning threshold is set to 80 percent. When the maximum-prefix limit is exceeded, the session with this peer will be torn down, all routes learned from this peer will be removed from the topology and routing tables, and this peer will be placed in a penalty state for 5 minutes (default penalty value).

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf VRF1
Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 1000 80
Router(config-router-af)# end
```

Example Configuring the Maximum-Prefix Limit for a Single Peer--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for a single peer. The maximum limit is set to 1000 prefixes, and the warning threshold is set to 80 percent. When the maximum-prefix limit is exceeded, the session with this peer will be torn down, all routes learned from this peer will be removed from the topology and routing tables, and this peer will be placed in a penalty state for 5 minutes (default penalty value).

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# neighbor 10.0.0.1 maximum-prefix 1000 80

Router(config-router-af)# exit-address-family
```

Example Configuring the Maximum-Prefix Limit for All Peers--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for all peers. The maximum limit is set to 10,000 prefixes, the warning threshold is set to 90 percent, the restart timer is set to 4 minutes, a decay penalty is configured for the restart timer with the **dampened**keyword, and all timers are configured to be reset to 0 every 60 minutes. When the maximum-prefix limit is exceeded, all peering sessions will be torn down, all routes learned from all peers will be removed from the topology and routing tables, and all peers will be placed in a penalty state for 4 minutes (user-defined penalty value). A dampening exponential decay penalty will also be applied.

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf VRF1
Router(config-router-af)# neighbor maximum-prefix 10000 90 dampened reset-time 60 restart
4
Router(config-router-af)# end
```

Example Configuring the Maximum-Prefix Limit for All Peers--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for all peers. The maximum limit is set to 10,000 prefixes, the warning threshold is set to 90 percent, the restart timer is set to 4 minutes, a decay penalty is configured for the restart timer with the **dampened**keyword, and all timers are configured to be reset to 0 every 60 minutes. When the maximum-prefix limit is exceeded, all peering sessions will be torn down, all routes learned from all peers will be removed from the topology and routing tables, and all peers will be placed in a penalty state for 4 minutes (user-defined penalty value). A dampening exponential decay penalty will also be applied.

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# neighbor maximum-prefix 10000 90 dampened reset-time 60 restart
4
Router(config-router-af)# exit-address-family
```

Example Configuring the Maximum-Prefix Limit for Redistributed Routes--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for routes learned through redistribution. The maximum limit is set to 5000 prefixes and the warning threshold is set to 95 percent. When the number of prefixes learned through redistribution reaches 4750 (95 percent of 5000), warning messages will be displayed in the console. Because the **warning-only** keyword is configured, the topology and routing tables will not be cleared and route redistribution will not be placed in a penalty state.

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf VRF1
Router(config-router-af)# redistribute maximum-prefix 5000 95 warning-only
Router(config-router-af)# end
```

Example Configuring the Maximum-Prefix Limit for Redistributed Routes--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for routes learned through redistribution. The maximum limit is set to 5000 prefixes and the warning threshold is set to 95 percent. When the number of prefixes learned through redistribution reaches 4750 (95 percent of 5000), warning messages will be displayed in the console. Because the **warning-only** keyword is configured, the topology and routing tables will not be cleared and route redistribution will not be placed in a penalty state.

```
Router(config)# router eigrp virtual-name1
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# network 172.16.0.0
Router(config-router-af)# topology base
Router(config-router-af-topology)# redistribute maximum-prefix 5000 95 warning-only
Router(config-router-af-topology)# exit-af-topology
```

Example Configuring the Maximum-Prefix Limit for an EIGRP Process--Autonomous System Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for an EIGRP process, which includes routes learned through redistribution and routes learned through EIGRP peering sessions. The maximum limit is set to 50,000 prefixes. When the number of prefixes learned through redistribution reaches 37,500 (75 percent of 50,000), warning messages will be displayed in the console.

When the maximum-prefix limit is exceeded, all peering sessions will be reset, the topology and routing tables will be cleared, and redistributed routes and all peering sessions will be placed in a penalty state.

```
Router(config)# router eigrp 100
Router(config-router)# address-family ipv4 vrf RED
Router(config-router-af)# maximum-prefix 50000
Router(config-router-af)# end
```

Example Configuring the Maximum-Prefix Limit for an EIGRP Process--Named Configuration

The following example, starting in global configuration mode, configures the maximum-prefix limit for an EIGRP process, which includes routes learned through redistribution and routes learned through EIGRP peering sessions. The maximum limit is set to 50,000 prefixes. When the number of prefixes learned through redistribution reaches 37,500 (75 percent of 50,000), warning messages will be displayed in the console.

When the maximum-prefix limit is exceeded, all peering sessions will be reset, the topology and routing tables will be cleared, and redistributed routes and all peering sessions will be placed in a penalty state.

```
Router(config)# router eigrp virtual-name
Router(config-router)# address-family ipv4 vrf VRF1 autonomous-system 45000
Router(config-router-af)# topology base
Router(config-router-af-topology)# maximum-prefix 50000
Router(config-router-af-topology)# exit-af-topology
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
EIGRP commands	*Cisco IOS IP Routing: EIGRP Command Reference*
EIGRP autonomous system configuration and EIGRP named configuration	Configuring EIGRP module
BGP cost community configuration tasks for EIGRP MPLS VPN PE-CE	BGP Cost Community Support module
Basic EIGRP configuration tasks	Configuring EIGRP module
EIGRP MPLS VPN configuration tasks	EIGRP MPLS VPN PE-CE Site of Origin (SoO) module
MPLS VPNs configuration tasks	Configuring MPLS Layer 3 VPNs module
System logging	Troubleshooting and Fault Management module of the *Cisco IOS Network Management Configuration Guide*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	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
None	--

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 Prefix Limit 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 13: Feature Information for EIGRP Prefix Limit Support

Feature Name	Releases	Feature Information
EIGRP Prefix Limit Support	12.0(29)S 12.2(33)SRE 12.3(14)T 15.0(1)M	The EIGRP Prefix Limit Support feature introduces the capability to limit the number of prefixes per VRF that are accepted from a specific peer or to limit all prefixes that are accepted by an Enhanced Interior Gateway Routing Protocol (EIGRP) process through peering and redistribution. The following commands were introduced or modified by this feature: **maximum-prefix**, **neighbor maximum-prefix** (EIGRP), **redistribute maximum-prefix** (EIGRP), **show ip eigrp accounting**, **show ip eigrp vrf accounting**. In Cisco IOS Release 15.0(1)M and 12.2(33)SRE, the following commands were introduced or modified: **maximum-prefix**,

Feature Name	Releases	Feature Information
		neighbor description, **redistribute maximum-prefix**(EIGRP), **show eigrp address-family accounting**, **show ip eigrp accounting**. In Cisco IOS Release 15.0(1)M and 12.2(33)SRE, the following command was replaced: **show ip eigrp vrf accounting**.

EIGRP Support for Route Map Filtering

The EIGRP Support for Route Map Filtering feature enables Enhanced Interior Gateway Routing Protocol (EIGRP) to interoperate with other protocols to leverage additional routing functionality by filtering inbound and outbound traffic based on complex route map options. Several extended filtering options are introduced to provide EIGRP-specific match choices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 document.

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 Support for Route Map Filtering

EIGRP Route Map Support

EIGRP support for route map filtering enables EIGRP to interoperate with other protocols by filtering inbound and outbound traffic based on route map options. Additional EIGRP-specific match choices are available to allow flexibility in fine-tuning EIGRP network operations.

EIGRP supports the route map filtering capability that exists for other routing protocols to filter routes being redistributed into their protocol. For more details about understanding and configuring route maps, see the Enabling Policy Routing section of the Configuring IP Routing Protocol-Independent Features module of the *Cisco IOS IP Routing: Protocol-Independent Configuration Guide* .

Match options allow EIGRP to filter internal and external routes based on source protocols, to match a metric against a range, and to match on an external protocol metric.

EIGRP can be configured to filter traffic using a route map and the **redistribute** or **distribute-list** command. Using a route map with the **redistribute** command allows routes that are redistributed from the routing table to be filtered with a route map before being admitted into an EIGRP topology table. Routes that are dynamically received from, or advertised to, EIGRP peers can be filtered by adding a route map option to the **distribute-list** command.

A route map may be configured with both the **redistribute** and the **distribute-list** commands in the same routing process. When a route map is used with a **distribute-list** command that is configured for inbound or outbound filtering, route packets that are learned from or advertised to EIGRP peers can be processed with the route map to provide better control of route selection during the route exchange process. Redistribution serves as a mechanism to import routes into the EIGRP topology table from a routing table. A route map configured with the **redistribute** command adds flexibility to the redistribution capability and results in a more specific redistributed route selection.

The use of route maps to filter traffic is the same for both autonomous-system configurations and named configurations. See the Configuring EIGRP module for more information about autonomous system and named configurations.

Demands for EIGRP to interoperate with other protocols and flexibility in fine-tuning network operation necessitate the capability to filter traffic using a route map.

How to Configure EIGRP Support for Route Map Filtering

Setting EIGRP Tags Using a Route Map for Autonomous System Configurations

Perform this task to set EIGRP tags for autonomous system configurations using a route map. The EIGRP metrics used for filtering are configured within a route map. The first match clause defines EIGRP routes that contain an external protocol metric between 400 and 600 inclusive; the second match clause defines EIGRP external routes that match a source protocol of BGP and the autonomous system 45000. When the two match clauses are true, a tag value of the destination routing protocol is set to 5. This route map can be used with the **distribute-list** command; see the Example Setting EIGRP Tags Using a Route Map Autonomous System Configuration, page 140 for an example configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*]
5. **match source-protocol** *source-protocol* [*autonomous-system-number*]
6. **set tag** *tag-value*
7. **exit**
8. **router eigrp** *as-number*
9. **network** *ip-address*
10. **distribute-list route-map** *map-tag* **in**

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.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map metric-range`	Enters route-map configuration mode.
Step 4	**match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*] **Example:** `Router(config-route-map)# match metric external 500 +- 100`	Specifies a match clause that filters inbound updates that match an internal or external protocol metric. • *metric-value* --Internal protocol metric, which can be an EIGRP five-part metric. The range is from 1 to 4294967295. • **external** --External protocol metric. The range is from 1 to 4294967295. • **+-** *deviation-number* --(Optional) Represents a standard deviation. The deviation can be any number. There is no default. **Note** When you specify a metric deviation with the **+** and **-** keywords, the router will match any metric that falls inclusively in that range. **Note** The external protocol metric is not the same as the EIGRP assigned route metric, which is a figure computed from EIGRP vectorized metric components (delay, bandwidth, reliability, load, and MTU).

	Command or Action	Purpose
Step 5	**match source-protocol** *source-protocol* [*autonomous-system-number*] **Example:** `Router(config-route-map)# match source-protocol bgp 45000`	Specifies a match clause that matches external routes from sources that match the source protocol. • *source-protocol* --Protocol to match. The valid keywords are **bgp**, **connected**, **eigrp, isis**, **ospf**, **rip**, and **static**. There is no default. • *autonomous-system-number* --(Optional) Autonomous system number. The *autonomous-system-number* argument is not applicable to the **connected**, **static**, and **rip** keywords. The range is from 1 to 65535. There is no default.
Step 6	**set tag** *tag-value* **Example:** `Router(config-route-map)# set tag 5`	Sets a tag value on the route in the destination routing protocol when all the match criteria of a route map are met.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 8	**router eigrp** *as-number* **Example:** `Router(config)# router eigrp 1`	Configures the EIGRP routing process and enters router configuration mode.
Step 9	**network** *ip-address* **Example:** `Router(config-router)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 10	**distribute-list route-map** *map-tag* **in** **Example:** `Router(config-router)# distribute-list route-map metric-range in`	Filters networks received in updates.

Setting EIGRP Tags Using a Route Map for Named Configurations

Perform this task to set EIGRP tags for named configurations using a route map. The EIGRP metrics used for filtering are configured within a route map. The first match clause defines EIGRP routes that contain an external protocol metric between 400 and 600 inclusive; the second match clause defines EIGRP external routes that match a source protocol of BGP and the autonomous system 45000. When the two match clauses are true, a tag value of the destination routing protocol is set to 5. This route map can be used with the **distribute-list** command, see the Example Setting EIGRP Tags Using a Route Map Named Configuration, page 141 for an example configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **set metric** *bandwidth delay reliability loading mtu*
5. **match ip route-source** {*access-list-number*| *access-list-name*} [*...access-list-number* | *...access-list-name*]
6. **match metric** {*metric-value*| **external** *metric-value*} [**+-** *deviation-number*]
7. **match source-protocol** *source-protocol* [*autonomous-system-number*]
8. **set tag** *tag-value*
9. **exit**
10. **router eigrp** *virtual-instance-name*
11. Do 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*
12. **network** *ip-address* [*wildcard-mask*]
13. **af-interface** {**default** | *interface-type interface-number*}
14. **next-hop-self eigrp**
15. **exit-af-interface**
16. **topology** {**base** | *topology-name* **tid** *number*}
17. **distribute-list route-map** *map-tag* **in**

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.
Step 3	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]	Enters route-map configuration mode.

	Command or Action	Purpose
	Example: `Router(config)# route-map metric-range`	
Step 4	**set metric** *bandwidth delay reliability loading mtu* **Example:** `Router(config-route-map)# set metric 10000 10 255 1 1500`	(Optional) Sets the metric value for EIGRP in a route map.
Step 5	**match ip route-source** {*access-list-number*| *access-list-name*} [*...access-list-number* | *...access-list-name*] **Example:** `Router(config-route-map)# match ip route-source 5 80`	Redistributes routes that have been advertised by routers and access servers at the address specified by the access lists.
Step 6	**match metric** {*metric-value*| **external** *metric-value*} [+- *deviation-number*] **Example:** `Router(config-route-map)# match metric external 500 +- 100`	Specifies a match clause that includes EIGRP routes that match an internal or external protocol metric. • *metric-value* --Internal protocol metric, which can be an EIGRP five-part metric. The range is from 1 to 4294967295. • **external** --External protocol metric. The range is from 1 to 4294967295. • **+-** *deviation-number* --(Optional) Represents a standard deviation. The deviation can be any number. There is no default. **Note** When you specify a metric deviation with the **+** and **-** keywords, the router will match any metric that falls inclusively in that range. **Note** The external protocol metric is not the same as the EIGRP assigned route metric, which is a figure computed from EIGRP vectorized metric components (delay, bandwidth, reliability, load, and MTU).
Step 7	**match source-protocol** *source-protocol* [*autonomous-system-number*] **Example:** `Router(config-route-map)# match source-protocol bgp 45000`	Specifies a match clause that includes EIGRP external routes that match a source protocol. • *source-protocol* --Protocol to match. The valid keywords are **bgp**, **connected**, **eigrp**, **isis**, **ospf**, **rip**, and **static**. There is no default. • *autonomous-system-number* --(Optional) Autonomous system number. The *autonomous-system-number* argument is not applicable to the **connected**, **static**, and **rip** keywords. The range is from 1 to 65535. There is no default.
Step 8	**set tag** *tag-value*	Sets a tag value on the route in the destination routing protocol when all the match criteria of a route map are met.

	Command or Action	Purpose
	Example: `Router(config-route-map)# set tag 5`	
Step 9	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode and returns to global configuration mode.
Step 10	**router eigrp** *virtual-instance-name* **Example:** `Router(config)# router eigrp virtual-name1`	Configures the EIGRP routing process and enters router configuration mode.
Step 11	Do 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:** `Router(config-router)# address-family ipv4 autonomous-system 45000`	Enters address family configuration mode to configure an EIGRP IPv4 or IPv6 routing instance.
Step 12	**network** *ip-address* [*wildcard-mask*] **Example:** `Router(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 13	**af-interface** {**default** | *interface-type interface-number*} **Example:** `Router(config-router-af)# af-interface default`	Enters address family interface configuration mode to configure interface-specific EIGRP commands.
Step 14	**next-hop-self eigrp** **Example:** `Router(config-router-af-interface)# next-hop-self eigrp`	Enables EIGRP to advertise routes with the local outbound interface address as the next hop.
Step 15	**exit-af-interface**	Exits address-family interface configuration mode.

	Command or Action	Purpose
	Example: `Router(config-router-af-interface)# exit-af-interface`	
Step 16	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Router(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 17	**distribute-list route-map** *map-tag* **in** **Example:** `Router(config-router-af-topology)# distribute-list route-map metric-range in`	Filters networks received in updates.

Configuration Examples for EIGRP Support for Route Map Filtering

- Example Setting EIGRP Tags Using a Route Map Autonomous System Configuration, page 140
- Example Setting EIGRP Tags Using a Route Map Named Configuration, page 141

Example Setting EIGRP Tags Using a Route Map Autonomous System Configuration

The following example shows how to configure a route map to match an EIGRP external protocol metric route with an allowable deviation of 100, a source protocol of BGP, and an autonomous system 45000. When the two match clauses are true, the tag value of the destination routing protocol is set to 5. The route map is used to distribute incoming packets for an EIGRP process.

```
Router(config)# route-map metric-range
Router(config-route-map)# match metric external 500 +- 100
Router(config-route-map)# match source-protocol bgp 45000
Router(config-route-map)# set tag 5
Router(config-route-map)# exit
Router(config)# router eigrp 1
Router(config-router)# network 172.16.0.0
Router(config-router)# distribute-list route-map metric_range in
```

The following example shows how to configure a route map to match EIGRP routes with a metric of 110, 200, or an inclusive range of 700 to 800. When the match clause is true, the tag value of the destination routing protocol is set to 10. The route map is used to redistribute EIGRP packets.

```
Router(config)# route-map metric-eigrp
Router(config-route-map)# match metric 110 200 750 +- 50
Router(config-route-map)# set tag 10
Router(config-route-map)# exit
Router(config)# router eigrp 1
```

```
Router(config-router)# network 172.21.1.0/24
Router(config-router)# redistribute eigrp route-map metric-eigrp
```

Example Setting EIGRP Tags Using a Route Map Named Configuration

The following example shows how to configure a route map to match an EIGRP external protocol metric route with an allowable deviation of 100, a source protocol of BGP, and an autonomous system 45000. When the two match clauses are true, the tag value of the destination routing protocol is set to 5. The route map is used to distribute incoming packets for an EIGRP process.

```
Router(config)# route-map metric_range
Router(config-route-map)# match metric external 500 +- 100
Router(config-route-map)# match source-protocol bgp 45000
Router(config-route-map)# set tag 5
Router(config-route-map)# exit
Router(config)# router eigrp virtual-name

Router(config-router)# address-family ipv4 autonomous-system 45000

Router(config-router-af)# network 172.21.1.0/24
Router(config-router-af)# topology base
Router(config-router-af-topology)# distribute-list route-map metric_range in
```

The following example shows how to configure a route map to match EIGRP routes with a metric of 110, 200, or an inclusive range of 700 to 800. When the match clause is true, the tag value of the destination routing protocol is set to 10. The route map is used to redistribute EIGRP packets.

```
Router(config)# route-map metric_eigrp
Router(config-route-map)# match metric 110 200 750 +- 50
Router(config-route-map)# set tag 10
Router(config-route-map)# exit
Router(config)# router eigrp virtual-name
Router(config-router)# address-family ipv4 autonomous-system 45000
Router(config-router-af)# network 172.21.1.0/24
Router(config-router-af)# topology base
Router(config-router-af-topology)# distribute-list route-map metric-range in
```

AdditionalReferences

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
EIGRP overview and configuration	The Configuring EIGRP section of the *Cisco IOS IP Routing: EIGRP Configuration Guide*
EIGRP commands including syntax, usage guidelines, and examples	*Cisco IOS IP Routing: EIGRP Command Reference*

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	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

RFC	Title
None	--

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 Support for Route Map Filtering

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 Support for Route Map Filtering

Feature Name	Releases	Feature Information
EIGRP Support for Route Map Filtering	12.2(33)SRA 12.2(33)SRE 12.2(33)SXH 12.2(33)XNE 12.3(8)T 15.0(1)M 15.0(1)S	The EIGRP Support for Route Map Filtering feature enables EIGRP to interoperate with other protocols by filtering inbound and outbound traffic based on complex route map options. Several extended filtering options are introduced to provide EIGRP-specific match choices. The following commands were introduced or modified by this feature: **match metric** (IP), **match source-protocol**, **show ip eigrp topology**. In Cisco IOS Release 15.0(1)M, 12.2(33)SRE, and 12.2(33)XNE the following command was introduced or modified for this feature: **show eigrp address-family topology**