EIGRP

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

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

- Finding Feature Information, on page 1
- Information About Configuring EIGRP, on page 1
- How to Configure EIGRP, on page 12
- Configuration Examples for EIGRP, on page 40
- Additional References for EIGRP, on page 51
- Feature Information for EIGRP, on page 53

Finding Feature Information

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

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

Information About Configuring EIGRP

EIGRP Features

- Increased network width--With IP Routing Information Protocol (RIP), the largest possible width of your network is 15 hops. When EIGRP is enabled, the largest possible width is increased to 100 hops, and the EIGRP metric is large enough to support thousands of hops.

- Fast convergence--The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.
- Partial updates--EIGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for EIGRP packets.
- Neighbor discovery mechanism--This simple protocol-independent hello mechanism is used to learn about neighboring devices.
- Scaling--EIGRP scales to large networks.

EIGRP Autonomous System Configuration

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

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

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

EIGRP Named Configuration

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

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

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

EIGRP Neighbor Relationship Maintenance

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

The reliable transport protocol is responsible for the guaranteed, ordered delivery of Enhanced Interior Gateway Routing Protocol (EIGRP) packets to all neighbors. The reliable transport protocol supports intermixed

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

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

Neighbor Authentication

The authentication of packets being sent between neighbors ensures that a device accepts packets only from devices that have the same preshared key. If this authentication is not configured, you can intentionally or accidentally add another device to the network or send packets with different or conflicting route information onto the network, resulting in topology corruption and denial of service (DoS).

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

DUAL Finite State Machine

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

Protocol-Dependent Modules

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

Goodbye Message

The goodbye message is a feature designed to improve EIGRP network convergence. The goodbye message is broadcast when an EIGRP routing process is shut down to inform adjacent peers about an impending topology change. This feature allows supporting EIGRP peers to synchronize and recalculate neighbor

relationships more efficiently than would occur if the peers discovered the topology change after the hold timer expired.

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

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

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

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

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

EIGRP Metric Weights

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

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

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

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

Mismatched K Values

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

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

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

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

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

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

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

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

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

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

Routing Metric Offset Lists

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

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

EIGRP Cost Metrics

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

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

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

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

Table 1: EIGRP Vector Metrics

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

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

Table 2: EIGRP K-Value Defaults

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

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

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

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

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

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

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

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

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

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

Route Summarization

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

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

Summary Aggregate Addresses

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

Floating Summary Routes

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

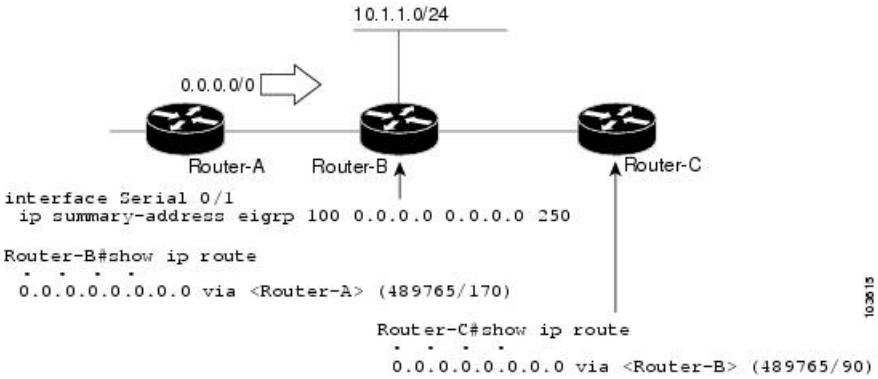
The figure below shows a network with three devices, Device-A, Device-B, and Device-C. Device-A learns a default route from elsewhere in the network and then advertises this route to Device-B. Device-B is configured so that only a default summary route is advertised to Device-C. The default summary route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

```
Device-B(config)# interface Serial 0/1
Device-B(config-if)# ip summary-address eigrp 192.168.0.1 255.255.0.0 95
```

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

```
Device-B(config)# Router eigrp virtual-name1
Device-B(config-router)# address-family ipv4 unicast vrf vrf1 autonomous-system 1
Device-B(config-router-af)# interface serial 0/1
Device-B(config-router-af-interface)# summary-address 192.168.0.1 255.255.0.0 95
```

Figure 1: Floating Summary Route Applied to Device-B



The configuration of the default summary route on Device-B sends a 0.0.0.0/0 summary route to Device-C and blocks all other routes, including the 10.1.1.0/24 route, from being advertised to Device-C. However, this configuration also generates a local discard route—a route for 10.10.10.10/0 on the null 0 interface with an administrative distance of 5—on Device-B. When this route is created, it overrides the EIGRP-learned default route. Device-B will no longer be able to reach destinations that it would normally reach through the 10.10.10.10/0 route.

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

```
Device-B(config-if)# ip summary-address eigrp 192.168.0.1 255.255.0.0 95
```

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

```
Device-B(config)# router eigrp virtual-name1
Device-B(config-router)# address-family ipv4 unicast vrf vrf1 autonomous-system 1
Device-B(config-router-af)# af-interface serial0/1
Device-B(config-router-af-interface)# summary-address eigrp 192.168.0.1 255.255.0.0 95
```

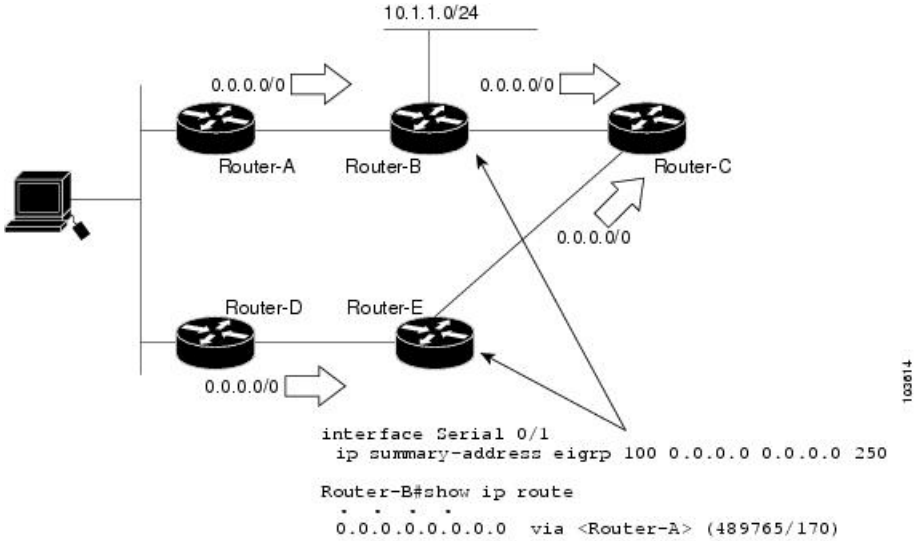
The administrative distance of 250, applied in the **summary-address** command, is now assigned to the discard route generated on Device-B. The 10.10.10.10/0, from Device-A, is learned through EIGRP and installed in the local routing table. Routing to Device-C is restored.

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

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

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

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



However, if the link between Device-A and Device-B fails, the network may incorrectly direct traffic because Device-B will continue to advertise the default route (10.10.10.10/0) to Device-C. In this scenario, Device-C still forwards traffic to Device-B, but Device-B drops the traffic. To avoid this problem, you should configure the summary address with an administrative distance only on single-homed remote devices or areas that have only one exit point between two segments of the network. If two or more exit points exist (from one segment of the network to another), configuring the floating default route can result in the formation of a null route (a route that has quick packet dropping capabilities).

Hello Packets and the Hold-Time Intervals

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

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

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

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

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

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

Split Horizon

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

By default, split horizon is enabled on all interfaces.

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

EIGRP Dual DMVPN Domain Enhancement

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

EIGRP, by default, sets the local outbound interface as the next-hop value while advertising a network to a peer, even when advertising routes out of the interface on which the routes were learned. This default setting can be disabled by using the **no ip next-hop-self** command in autonomous system configurations or the **no next-hop-self** command in named configurations. When the **next-hop self** command is disabled, EIGRP does not advertise the local outbound interface as the next hop if the route has been learned from the same interface. Instead, the received next-hop value is used to advertise learned routes. However, this functionality only evaluates the first entry in the EIGRP table. If the first entry shows that the route being advertised is learned on the same interface, then the received next hop is used to advertise the route. The **no next-hop-self** configuration ignores subsequent entries in the table, which may result in the **no-next-hop-self** configuration being dishonored on other interfaces.

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

Link Bandwidth Percentage

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

EIGRP vNETs

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

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

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

The following are some of the limitations of EIGRP vNETs:

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

EIGRP vNET Interface and Command Inheritance

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

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

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

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

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

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

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

How to Configure EIGRP

Enabling EIGRP Autonomous System Configuration

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

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 4	**network** *network-number* **Example:** `Device(config-router)# network 172.16.0.0`	Associates a network with an EIGRP routing process.
Step 5	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Enabling the EIGRP Named Configuration

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

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

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

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

Configuring Optional EIGRP Parameters in an Autonomous System Configuration

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **network** *ip-address* [*wildcard-mask*]

5. **passive-interface** [**default**] [*interface-type interface-number*]
6. **offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*]
7. **metric weights** *tos k1 k2 k3 k4 k5*
8. **no auto-summary**
9. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 1`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**passive-interface** [**default**] [*interface-type interface-number*] **Example:** `Device(config-router)# passive-interface`	(Optional) Suppresses EIGRP hello packets and routing updates on interfaces while still including the interface addresses in the topology database.
Step 6	**offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*] **Example:** `Device(config-router)# offset-list 21 in 10 gigabitethernet 0/0/1`	(Optional) Applies an offset to routing metrics.
Step 7	**metric weights** *tos k1 k2 k3 k4 k5* **Example:**	(Optional) Adjusts the EIGRP metric or K value. • EIGRP uses the following formula to determine the total metric to the network:

	Command or Action	Purpose
	`Device(config-router)# metric weights 0 2 0 2 0 0`	EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))) **Note** If K5 is 0, then (K5/ (Reliability + K4)) is d
Step 8	**no auto-summary** **Example:** `Device(config-router)# no auto-summary`	(Optional) Disables automatic summarization. **Note** Automatic summarization is enabled by defa
Step 9	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring Optional EIGRP Parameters in a Named Configuration

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

Step 5	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router-af)# network 172.16.0.0`	Specifies a network for the EIGRP routing process.
Step 6	**metric weights** *tos k1 k2 k3 k4 k5 k6* **Example:** `Device(config-router-af)# metric weights 0 2 0 2 0 0 0`	(Optional) Adjusts the EIGRP metric or K value. • EIGRP uses the following formula to determine the total 32-bit metric to the network: EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + (K3*Delay)*(K5/(Reliability + K4))) • EIGRP uses the following formula to determine the total 64-bit metric to the network: EIGRP Metric = 256*((K1*Throughput) + (K2*Throughput)/(256-Load) + (K3*Latency)+ (K6*Extended Attributes))*(K5/(Reliability + K4)))

	Command or Action	Purpose
		Note If K5 is 0, then (K5/ (Reliability + K4)) is
Step 7	**af-interface** *interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 8	**passive-interface** **Example:** `Device(config-router-af-interface)# passive-interface`	Suppresses EIGRP hello packets and routing updates on interfaces while still including the interface addresses in the topology database.
Step 9	**bandwidth-percent** *maximum-bandwidth-percentage* **Example:** `Device(config-router-af-interface)# bandwidth-percent 75`	Configures the percentage of bandwidth that may be used by an EIGRP address family on an interface.
Step 10	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 11	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 12	**offset-list** [*access-list-number* | *access-list-name*] {**in** | **out**} *offset* [*interface-type interface-number*] **Example:** `Device(config-router-af-topology)# offset-list 21 in 10 gigabitethernet 6/2`	(Optional) Applies an offset to routing metrics.
Step 13	**no auto-summary** **Example:** `Device(config-router-af-topology)# no auto-summary`	(Optional) Disables automatic summarization. **Note** Automatic summarization is enabled by de
Step 14	**end** **Example:** `Device(config-router-af-topology)# end`	Returns to privileged EXEC mode.

Configuring the EIGRP Redistribution Autonomous System Configuration

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

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

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

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

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **network** *ip-address* [*wildcard-mask*]
5. **redistribute** *protocol*
6. **distance eigrp** *internal-distance external-distance*
7. **default-metric** *bandwidth delay reliability loading mtu*
8. **end**

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 4	**network** *ip-address* [*wildcard-mask*] **Example:** `Device(config-router)# network 172.16.0.0`	Associates networks with an EIGRP routing process.
Step 5	**redistribute** *protocol* **Example:** `Device(config-router)# redistribute rip`	Redistributes routes from one routing domain into another routing domain.
Step 6	**distance eigrp** *internal-distance external-distance* **Example:** `Device(config-router)# distance eigrp 80 130`	Allows the use of two administrative distances—internal and external.
Step 7	**default-metric** *bandwidth delay reliability loading mtu* **Example:** `Device(config-router)# default-metric 1000 100 250 100 1500`	Sets metrics for EIGRP.
Step 8	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Route Summarization Autonomous System Configuration

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system*
4. **no auto-summary**
5. **exit**
6. **interface** *type number*
7. **no switchport**
8. **bandwidth** *kpbs*
9. **ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*]
10. **ip bandwidth-percent eigrp** *as-number percent*
11. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**no auto-summary** **Example:** `Device(config-router)# no auto-summary`	Disables automatic summarization of subnet routes into network-level routes
Step 5	**exit** **Example:** `Device(config-router)# exit`	Exits router configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/3`	Enters interface configuration mode.
Step 7	**no switchport** **Example:** `Device(config-if)# no switchport`	Puts an interface into Layer 3 mode
Step 8	**bandwidth** *kpbs* **Example:** `bandwidth 56`	Sets the inherited and received bandwidth values for an interface
Step 9	**ip summary-address eigrp** *as-number ip-address mask* [*admin-distance*] [**leak-map** *name*] **Example:**	(Optional) Configures a summary aggregate address.

	Command or Action	Purpose
	`Device(config-if)# ip summary-address eigrp 100 10.10.10.10 255.255.255.0`	
Step 10	**ip bandwidth-percent eigrp** *as-number percent* **Example:** `Device(config-if)# ip bandwidth-percent eigrp 209 75`	(Optional) Configures the percentage of bandwidth that may be used by EIGRP on an interface.
Step 11	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Route Summarization Named Configuration

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

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

Step 5	**af-interface {default** | *interface-type interface-number***}** **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.
Step 6	**summary-address** *ip-address mask* [*administrative-distance* [**leak-map** *leak-map-name*]] **Example:** `Device(config-router-af-interface)# summary-address 192.168.0.10 255.255.0.0`	Configures a summary address for EIGRP.
Step 7	**exit-af-interface** **Example:** `Device(config-router-af-interface)# exit-af-interface`	Exits address family interface configuration mode.
Step 8	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.

	Command or Action	Purpose
Step 9	**summary-metric** *network-address subnet-mask bandwidth delay reliability load mtu* **Example:** `Device(config-router-af-topology)# summary-metric 192.168.0.10/16 10000 10 255 1 1500`	(Optional) Configures a fixed metric for an EIGRP summary aggregate address.
Step 10	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Event Logging Autonomous System Configuration

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 4	**eigrp event-log-size** *size* **Example:** `Device(config-router)# eigrp event-log-size 5000010`	(Optional) Sets the size of the EIGRP event log.
Step 5	**eigrp log-neighbor-changes** **Example:** `Device(config-router)# eigrp log-neighbor-changes`	(Optional) Enables logging of EIGRP neighbor adjacency changes. • By default, the system logs EIGRP neighbor adjacency changes to help you monitor the stability of the routing system and detect problems.
Step 6	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router)# eigrp log-neighbor-warnings 300`	(Optional) Enables the logging of EIGRP neighbor warning messages.
Step 7	**end** **Example:** `Device(config-router)# end`	Exits router configuration mode and returns to privileged EXEC mode.

Configuring the EIGRP Event Logging Named Configuration

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

Step 5	**eigrp log-neighbor-warnings** [*seconds*] **Example:** `Device(config-router-af)# eigrp log-neighbor-warnings 300`	(Optional) Enables the logging of EIGRP neighbor warning messages.
Step 6	**eigrp log-neighbor-changes** **Example:** `Device(config-router-af)# eigrp log-neighbor-changes`	(Optional) Enables logging of EIGRP neighbor adjacency changes. • By default, the system logs EIGRP neighbor adjacency changes to help you monitor the stability of the routing system and detect problems.

	Command or Action	Purpose
Step 7	**topology** {**base** | *topology-name* **tid** *number*} **Example:** `Device(config-router-af)# topology base`	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 8	**eigrp event-log-size** *size* **Example:** `Device(config-router-af-topology)# eigrp event-log-size 10000`	(Optional) Sets the size of the EIGRP event log.
Step 9	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

Configuring Equal and Unequal Cost Load Balancing Autonomous System Configuration

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

Configuring Equal and Unequal Cost Load Balancing Named Configuration

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

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

	Command or Action	Purpose
	`Device(config-router-af-topology)# maximum-paths 5`	
Step 8	**variance** *multiplier* **Example:** `Device(config-router-af-topology)# variance 1`	Controls load balancing in an internetwork based on EIGRP.
Step 9	**end** **Example:** `Device(config-router-af-topology)# end`	Exits address family topology configuration mode and returns to privileged EXEC mode.

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

Note Cisco recommends not to adjust the hold time.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router eigrp** *autonomous-system-number*
4. **exit**
5. **interface** *type number*
6. **no switchport**
7. **ip hello-interval eigrp** *autonomous-system-number seconds*
8. **ip hold-time eigrp** *autonomous-system-number seconds*
9. **end**

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *autonomous-system-number* **Example:** `Device(config)# router eigrp 101`	Enables an EIGRP routing process and enters router configuration mode. • A maximum of 30 EIGRP routing processes can be configured.
Step 4	**exit** **Example:** `Device(config-router)# exit`	Exits to global configuration mode.
Step 5	**interface** *type number* **Example:** `Device(config)# interface Gigabitethernet 1/0/9`	Enters interface configuration mode.
Step 6	**no switchport** **Example:** `Device(config-if)# no switchport`	Puts an interface into Layer 3 mode
Step 7	**ip hello-interval eigrp** *autonomous-system-number seconds* **Example:** `Device(config-if)# ip hello-interval eigrp 109 10`	Configures the hello interval for an EIGRP routing process.
Step 8	**ip hold-time eigrp** *autonomous-system-number seconds* **Example:** `Device(config-if)# ip hold-time eigrp 109 40`	Configures the hold time for an EIGRP routing process. **Note** Do not adjust the hold time without cons
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

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

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**router eigrp** *virtual-instance-name* **Example:** `Device(config)# router eigrp virtual-name1`	Enables an EIGRP routing process and enters router configuration mode.
Step 4	Enter one of the following: • **address-family ipv4** [**multicast**] [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*	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:
	`Device(config-router)# address-family ipv4 autonomous-system 45000`
	`Device(config-router)# address-family ipv6 autonomous-system 45000`	

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

Disabling the Split Horizon Autonomous System Configuration

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

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

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

DETAILED STEPS

Procedure

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

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

EIGRP, by default, sets the next-hop value to the local outbound interface address for routes that it is advertising, even when advertising those routes back from the same interface from where they were learned. Perform this task to change this default setting and configure EIGRP to use the received next-hop value when advertising these routes. Disabling next-hop-self is primarily useful in DMVPN spoke-to-spoke topologies.

By default, split horizon is enabled on all interfaces.

SUMMARY STEPS

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

- **address-family ipv6** [**unicast**] [**vrf** *vrf-name*] **autonomous-system** *autonomous-system-number*

5. **af-interface** {**default** | *interface-type interface-number*}
6. **no split-horizon**
7. **no next-hop-self [no-ecmp-mode]**
8. **end**

DETAILED STEPS

Procedure

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

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

Step 5	**af-interface** {**default**	*interface-type interface-number*} **Example:** `Device(config-router-af)# af-interface gigabitethernet 0/0/1`	Enters address family interface configuration mode and configures interface-specific EIGRP commands.

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

Monitoring and Maintaining the EIGRP Autonomous System Configuration

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

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

Displays prefix accounting information for EIGRP processes.

Example:

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

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

Displays information about interfaces that are configured for EIGRP.

Example:

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

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

Displays neighbors discovered by EIGRP.

Example:

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

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

Displays neighbors discovered by EIGRP

Example:

```
Device# show ip eigrp neighbors
```

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

Displays entries in the EIGRP topology table.

Example:

```
Device# show ip eigrp topology
```

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

Displays the number of EIGRP packets sent and received.

Example:

```
Device# show ip eigrp traffic
```

Monitoring and Maintaining the EIGRP Named Configuration

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

SUMMARY STEPS

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

DETAILED STEPS

Procedure

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Device# enable
```

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

Displays prefix accounting information for EIGRP processes.

Example:

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

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

Displays information about EIGRP address-family events.

Example:

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

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

Displays information about interfaces that are configured for EIGRP.

Example:

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

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

Displays the neighbors that are discovered by EIGRP.

Example:

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

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

Displays information about EIGRP timers and expiration times.

Example:

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

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

Displays entries in the EIGRP topology table.

Example:

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

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

Displays the number of EIGRP packets that are sent and received.

Example:

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

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

Displays general information, including the versions of the EIGRP protocol features that are currently running on the device.

Example:

```
Device# show eigrp plugins
```

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

Displays further information about EIGRP protocols that are currently running on a device.

Example:

```
Device# show eigrp protocols
```

Configuration Examples for EIGRP

Example: Enabling EIGRP—Autonomous System Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.10
```

Example: Enabling EIGRP—Named Configuration

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.10
```

Example: EIGRP Parameters—Autonomous System Configuration

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

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.10
Device(config-router)# passive-interface
Device(config-router)# offset-list 21 in 10 ethernet 0
Device(config-router)# metric weights 0 2 0 2 0 0
Device(config-router)# no auto-summary
Device(config-router)# exit
```

Example: EIGRP Parameters—Named Configuration

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

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# network 172.16.0.10
```

```
Device(config-router-af)# metric weights 0 2 0 2 0 0 0
Device(config-router-af)# metric rib-scale 100
Device(config-router-af)# af-interface gigabitethernet 0/0/1
Device(config-router-af-interface)# passive-interface
Device(config-router-af-interface)# bandwidth-percent 75
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af-interface)# topology base
Device(config-router-af-topology)# offset-list 21 in 10 gigabitethernet 0/0/1
Device(config-router-af-topology)# no auto-summary
Device(config-router-af-topology)# exit-af-topology
```

Example: EIGRP Redistribution—Autonomous System Configuration

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

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# network 172.16.0.10
Device(config-router)# redistribute rip
Device(config-router)# distance eigrp 80 130
Device(config-router)# default-metric 1000 100 250 100 1500
```

Example: EIGRP Route Summarization—Autonomous System Configuration

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

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 101
Device(config-router)# no auto-summary
Device(config-router)# exit
Device(config)# interface Gigabitethernet 1/0/1
Device(config-if)# no switchport
bandwidth 56
Device(config-if)# ip summary-address eigrp 100 192.168.0.10 255.255.255.0
Device(config-if)# ip bandwidth-percent eigrp 209 75
```

Note You should not use the **ip summary-address eigrp** summarization command to generate the default route (10.10.10.10) from an interface because this creates an EIGRP summary default route to the null 0 interface with an administrative distance of 5. The low administrative distance of this default route can cause this route to displace default routes learned from other neighbors through the routing table. If the default route learned from the neighbors is displaced by the summary default route, or if the summary route is the only default route present, all traffic destined for the default route will not leave the router; instead, traffic will be sent to the null 0 interface, where it is dropped. The recommended way to send only the default route out of a given interface is to use the **distribute-list** command. You can configure this command to filter all outbound route advertisements sent out from the interface with the exception of the default (10.10.10.10).

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 192.168.0.10 only from Ethernet interface 0/0.

```
Device> enable
Device# configure terminal
Device(config)# router eigrp virtual-name1
Device(config-router)# address-family ipv4 autonomous-system 45000
Device(config-router-af)# af-interface ethernet 0/0
Device(config-router-af-interface)# summary-address 192.168.10.10 255.255.0.0
Device(config-router-af-interface)# exit-af-interface
Device(config-router-af)# topology base
Device(config-router-af-topology)# summary-metric 192.168.0.10/16 10000 255 255 255 0
```

Example: EIGRP Event Logging—Autonomous System Configuration

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

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

Example: EIGRP Event Logging—Named Configuration

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

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

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

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

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

```
Device(config-router)# maximum-paths 5
Device(config-router)# variance 1
```

Example: Equal and Unequal Cost Load Balancing—Named Configuration

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

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

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

```
Device> enable
Device# configure terminal
Device(config)# router eigrp 1
Device(config-router)# exit
Device(config)# interface Gibabitethernet 1/0/9
Device(config-if)# no switchport
Device(config-if)# ip hello-interval eigrp 109 10
Device(config-if)# ip hold-time eigrp 109 40
```

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

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

Example: Disabling the Split Horizon—Autonomous System Configuration

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

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

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

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

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

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

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

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

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

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

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

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

Building configuration...
Derived configuration : 478 bytes
!
interface GigabitEthernet0/0/0.3
 description Subinterface for VNET vrf1
 vrf forwarding vrf1
 encapsulation dot1Q 3
 ip address 192.0.2.1 255.255.255.0
 ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 x
 ip bandwidth-percent eigrp 1 3
```

```
 ip dampening-change eigrp 1 30
 ip hello-interval eigrp 1 6
 ip hold-time eigrp 1 18
 no ip next-hop-self eigrp 1
 no ip split-horizon eigrp 1
 end
```

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

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


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

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

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

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

Building configuration...
Derived configuration : 479 bytes
!
interface GigabitEthernet0/0/0.3
```

```
 description Subinterface for VNET vrf1
 vrf forwarding vrf1
 encapsulation dot1Q 3
 ip address 192.0.2.1 255.255.255.0
 no ip authentication mode eigrp 1 md5
 ip authentication key-chain eigrp 1 y
 ip bandwidth-percent eigrp 1 99
 no ip dampening-change eigrp 1
 ip hello-interval eigrp 1 7
 ip hold-time eigrp 1 21
 ip next-hop-self eigrp 1
 ip split-horizon eigrp 1
 end
```

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

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

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

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

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

Building configuration...

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

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

Building configuration...
Current configuration : 183 bytes
!
interface GigabitEthernet0/0/0
 vnet trunk
 ip address 192.0.2.1 255.255.255.0
 ip hello-interval eigrp 1 7
```

```
 ipv6 enable
 vnet name vrf1
  ip hello-interval eigrp 1 10
 !
 vnet global
 !
end

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

Building configuration...

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

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

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

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

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

Example: Monitoring and Maintaining the EIGRP Autonomous System Configuration

The **show ip eigrp** command displays prefix accounting information for EIGRP processes. The following is sample output from this command:

```
Device# 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
```

The **show ip eigrp events** command displays the EIGRP event log. The following is sample output from this command:

```
Device# 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
```

The **show ip eigrp interfaces** command displays information about interfaces that are configured for EIGRP. The following is sample output from this command:

```
Device# 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
```

The **show ip eigrp neighbors** command displays neighbors discovered by EIGRP. The following is sample output from this command:

```
Device# 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
```

The **show ip eigrp topology** command displays entries in the EIGRP topology table. The following is sample output from this command:

```
Device# 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
```

The **show ip eigrp traffic** command displays the number of EIGRP packets sent and received. The following is sample output from this command:

```
Device# 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)
```

Example: Monitoring and Maintaining the EIGRP Named Configuration

In this example, the **show eigrp address-family** command displays prefix accounting information for EIGRP processes:

```
Device# 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
```

In this example, the **show eigrp address-family** command displays information about EIGRP address-family events:

```
Device# 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
```

In this example, the **show eigrp address-family** command displays information about interfaces that are configured for EIGRP:

```
Device# 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
```

In this example, the **show eigrp address-family** command displays information about the neighbors that are discovered by EIGRP:

```
Device# 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
```

In this example, the **show eigrp address-family** command displays information about EIGRP timers and expiration times:

```
Device# 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)
```

In this example, the **show eigrp address-family** command displays entries in the EIGRP topology table:

```
Device# 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)
```

In this example, the **show eigrp address-family** command displays information about the number of EIGRP packets that are sent and received:

```
Device# 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
```

In this example, the **show eigrp plugins** command displays general information, including the versions of the EIGRP protocol features that are currently running on the device:

```
Device# 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
```

In this example, the **show eigrp protocols** command displays general information about EIGRP protocols that are currently running on a device:

```
Device# 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
```

Additional References for EIGRP

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
EIGRP commands	IP Routing: EIGRP Command Reference
EIGRP FAQ	EIGRP Frequently Asked Questions

Related Topic	Document Title
EIGRP L2/L3 API and Tunable Metric for Mobile Adhoc Networks feature	"Mobile Ad Hoc Networks for Router-to-Radio Communications" module of *the IP Mobility Configuration Guide*
EIGRP Technology Support	Enhanced Interior Gateway Routing Protocol
EIGRP Technology White Papers	Enhanced Interior Gateway Routing Protocol
IPv6 Routing EIGRP Support	*IPv6 Routing: EIGRP Support*
Protocol-independent features that work with EIGRP	*IP Routing: Protocol-Independent Configuration Guide*
Service Advertisement Framework	*Service Advertisement Framework Configuration Guide*
Service Advertisement Framework commands	Service Advertisement Framework Command Reference

Standards and RFCs

Standard/RFC	Title
FIPS PUB 180-2	*SECURE HASH STANDARD (SHS)*
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 https://cfnng.cisco.com/. An account on Cisco.com is not required.