



# EIGRP

EIGRP Classic Mode



## In This Section

- + Configuring Classic EIGRP
- + Verifying EIGRP Adjacencies
- + Verifying EIGRP Topology

# Classic EIGRP Overview

- + EIGRP prior to EIGRP Named Mode
  - + I.e. EIGRP prior to IOS 15.x
- + AKA EIGRP Autonomous System Configuration
  - + Process name is the AS number

# Enabling Classic EIGRP

- + Enable the global process
  - + router eigrp [AS]
  - + AS number must match to be adjacent
- + Enable the interface process
  - + network [address] [wildcard]
  - + Similar to OSPF network statement

# Verifying EIGRP

- + Verify EIGRP is enabled
  - + **show ip eigrp interfaces [detail]**
  - + **show ip protocols**
- + Verify EIGRP transport
  - + **debug eigrp packet [hello | ack | update | query | reply...]**

# Verifying EIGRP

- + Verify neighbor adjacencies
  - + **show ip eigrp neighbors [detail]**
  - + Queue count should be 0 if converged
- + Verify EIGRP topology
  - + **show ip eigrp topology [all-links]**
  - + **show ip eigrp topology [prefix/len]**





# EIGRP

EIGRP Named Mode





## In This Section

- + Configuring EIGRP Named Mode

## EIGRP Named Mode (Multi-AF Mode)

- + New EIGRP configuration as of IOS 15.x
  - + Old configuration is EIGRP Classic Mode
- + Better config hierarchy
  - + All config now goes under global process
- + Better IPv4 & IPv6 feature parity
  - + Syntax is unified for IPv4 and IPv6 under one process
  - + Uses address-families to distinguish v4 and v6
- + Newer features aren't supported in Classic Mode
  - + E.g. Wide Metrics, IPv6 VRF Lite, etc.

# Configuring EIGRP Named Mode

- + Enable the global process
  - + **router eigrp [name]**
  - + Name is not the AS number
- + Enable the address-family
  - + E.g. **address-family ipv4 unicast autonomous-system 1**
- + Enable the interface process
  - + **network [address] [wildcard]**
  - + Similar to OSPF network statement
- + Customize the interfaces
  - + **af-interface [default | int]**

# Converting EIGRP Classic to Named Mode

- + Conversion is automated as of IOS 15.4S
  - + **Router(config-router)# eigrp upgrade-cli**
- + Conversion is hitless
  - + Performs graceful restart automatically
- + Supports both IPv4 and IPv6 EIGRP
  - + Can be in different ASes





# EIGRP

EIGRP Classic Metric Calculation



## In This Section

- + EIGRP Classic Metric Calculation

# EIGRP Classic Path Selection

- + EIGRP uses lowest composite metric based on...
  - + Bandwidth
    - + Inverse lowest bandwidth along path in Kbps scaled by  $10^7 * 256$
  - + Delay
    - + Cumulative delay along path in tens of microseconds ( $\mu s$ ) scaled by 256
  - + Load
    - + Highest load along path
  - + Reliability
    - + Lowest reliability along path



# EIGRP Classic Metric Calculation

- + Composite metric is computed as...
  - +  $\text{metric} = [k1 * BW + (k2 * BW)/(256 - \text{load}) + k3 * \text{delay}]$
  - + If  $k5 \neq 0$ ,  $\text{metric} = \text{metric} * [k5/(\text{reliability} + k4)]$
- + “K” values allow for manual weighting
  - + Defaults are  $K1 = 1$ ,  $K2 = 0$ ,  $K3 = 1$ ,  $K4 = 0$ ,  $K5 = 0$
  - + Implies default composite is bandwidth + delay
- + Can be modified with **metric weights** command
  - + Must match for adjacency to occur





# EIGRP

EIGRP Wide Metric Calculation



## In This Section

- + EIGRP Wide Metric Calculation

# EIGRP Classic Bandwidth Metric Problem

- + Classic EIGRP Metric uses bandwidth as...
  - +  $\text{CLASSIC\_EIGRP\_BW} = 10^7 * 256 / \text{INT\_BW}$
- + As bandwidths increase, visibility is lost
  - + 1 GigE = 2560
  - + 10 GigE = 256
  - + 20 GigE = 256
  - + 40 GigE = 256
  - + 100 GigE = 256

# EIGRP Classic Delay Metric Problem

- + Minimum configurable delay in IOS is 10μs
- + As bandwidths increase, delay visibility is lost
  - + 1 GigE = 10μs
  - + 10 GigE = 10μs
  - + 20 GigE = 10μs
  - + 40 GigE = 10μs
  - + 100 GigE = 10μs

# EIGRP Wide Metrics

- + Wide Metric is now 64 bits
- + BW problem fixed by scaling higher
  - +  $\text{WIDE\_EIGRP\_BW} = 10^7 * 65536 / \text{INT\_BW}$
- + DLY problem fixed by moving to picoseconds
  - +  $\text{WIDE\_EIGRP\_DLY} = \text{DLY\_PICO} * 65536 / 10^6$

# EIGRP Wide Metric Formula

- + BW is now Throughput
  - +  $\text{Throughput} = K1 * (\text{EIGRP\_BANDWIDTH} * \text{EIGRP\_WIDE\_SCALE}) / \text{Interface Bandwidth (kbps)}$
- + DLY is now Latency
  - +  $\text{Latency} = K3 * (\text{Delay} * \text{EIGRP\_WIDE\_SCALE}) / \text{EIGRP\_DELAY\_PICO}$
- + Default metric is still the same formula
  - +  $\text{metric} = (K1 * \min(\text{Throughput})) + (K3 * \sum(\text{Latency}))$



# EIGRP Wide Metric Configuration

- + Enable EIGRP named mode
- + Wide Metrics automatically enabled
- + Still backwards compatible with Classic

# Wide Metric & RIB Scaling

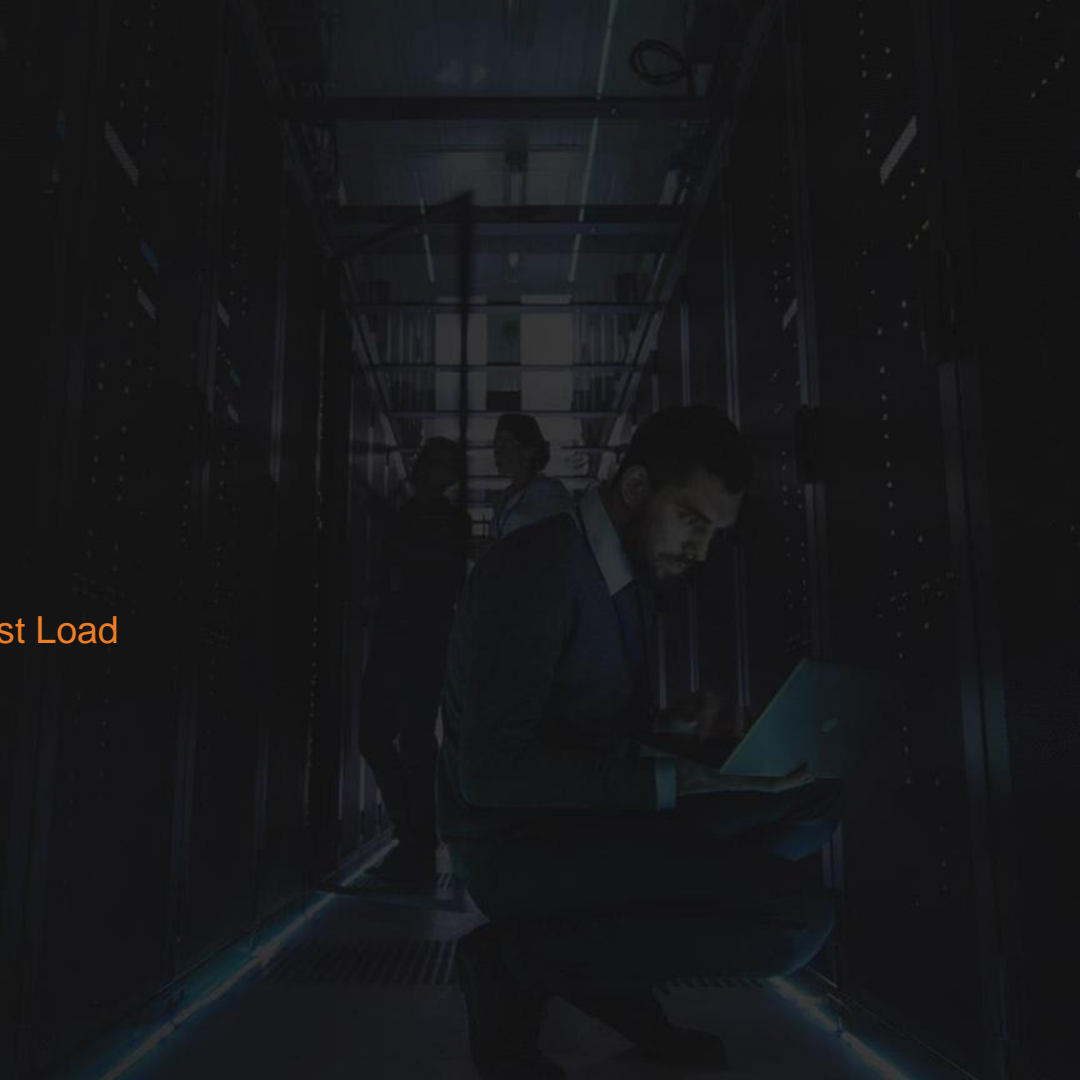
- + IOS RIB only supports 32-bit metrics
- + After EIGRP DUAL, metric is scaled down to fit in the RIB
  - + Default is to scale \* 1/128
  - + Can be modified with **metric rib-scale**





# EIGRP

EIGRP Traffic Engineering & Unequal Cost Load  
Balancing



## In This Section

- + EIGRP Feasibility Condition
- + Modifying EIGRP Path Selection
- + EIGRP Unequal Cost Load Balancing

# EIGRP Feasibility Condition

- + Feasibility Condition (FC) is used to select loop-free backup paths
  - + Paths meeting the FC are pre-computed during DUAL
- + Pre-computed backup paths have multiple advantages
  - + Sub-second reconvergence when primary path fails
  - + Fault isolation
  - + Unequal cost load distribution

# Feasibility Condition High Level Overview

- + First, find the best path
  - + Take the metric of the best path as  $X$
- + Next, find alternate backup paths
  - + Anyone with a metric less than  $X$  is closer to the destination than me
  - + Exclude anyone with a metric greater than or equal to  $X$

# Feasibility Condition Terminology

- + Successor
  - + Best path to a destination
- + Feasible Distance (FD)
  - + Composite metric of best path (successor)
- + Reported Distance (RD)
  - + Composite metric learned from neighbors
- + Feasible Successor (FS)
  - + Backup paths that meet the Feasibility Condition



# Feasibility Condition in Detail

- + Once best path is chosen, additional paths are examined for backup routes
- + FC finds loop-free backup routes via logic...
  - + If  $RD < FD$ , path is loop-free and viable backup
  - + e.g. if your metric is lower than mine, you are closer to the destination and loop-free
- + Paths that meet FC are Feasible Successors (FS)
  - + Only Feasible Successors can be used for unequal cost load balancing

# EIGRP Reconvergence

- + Reconvergence differs for paths with or without a FS
- + Paths without a Feasible Successor
  - + Loss of Successor sends route into Active state
  - + Send QUERY to all neighbors
  - + Reconverged when REPLY heard from all neighbors
- + Paths with a Feasible Successor
  - + Loss of Successor does not make route Active
  - + Feasible Successor promoted to Successor
  - + QUERY not generated
  - + Result is sub-second convergence

# EIGRP Traffic Engineering

- + Default EIGRP composite metric is Bandwidth + Delay
  - + Based on default K1 and K3 metric weights
- + Bandwidth is lowest bandwidth along the path on a per prefix basis
  - + Essentially the bandwidth bottleneck
  - + Hard to predict what a modification will affect
- + Delay is cumulative on a hop-by-hop basis
  - + Easier to influence path selection with
  - + Modified with **delay** interface command

# EIGRP Unequal Cost Load Balancing

- + EIGRP allows load distribution among unequal paths
  - + Only Feasible Successors are candidate for load balancing
- + Controlled by **variance** command
  - + If Feasible Distance \* variance > feasible successor, load balancing occurs
- + Traffic share is automatically calculated
  - + Links used in ratio proportional to their composite metrics
  - + Actual load balancing still controlled by switching path
    - + E.g. CEF per-flow





# EIGRP

EIGRP Authentication



## In This Section

- + EIGRP Authentication
- + Classic vs. Named Mode Authentication

# EIGRP Authentication

- + Supports two types of authentication
  - + MD5
  - + HMAC-SHA-256
- + MD5 Authentication
  - + Both Classical and Named EIGRP
  - + Uses key chains
  - + Supports automatic key rotation
- + SHA Authentication
  - + Only Named EIGRP
  - + No key chains, no rotation



# Classic EIGRP Authentication

- + Supports MD5 only
- + Define key chain globally
  - + Whitespace counts as a character
  - + Key number must match
  - + Supports automatic key rotation
- + Applied at interface level
  - + **ip authentication mode eigrp [AS] md5**
  - + **ip authentication key-chain eigrp [AS] [key-chain]**

# Automatic Key Rotation

- + Key chain supports multiple key numbers
  - + Router always sends lowest valid key
- + Key number's validity is based on time
  - + **accept-lifetime**
    - + When is key valid to be received
  - + **send-lifetime**
    - + When is key valid to be sent
- + Automatic rotation by defining different validity times
  - + Implies time must be agreed upon
  - + accept-lifetime should overlap in case of mismatch of time

# Named EIGRP Authentication

- + Supports MD5 and SHA
  - + MD5 uses same global key chains as Classic
  - + SHA uses static key
- + Defined under af-interface
  - + Can use **af-interface default**
  - + Specific interface overrides default





# EIGRP

EIGRP Summarization



## In This Section

- + EIGRP Query Scoping
- + EIGRP Summarization

# EIGRP Scalability

- + Sub-second convergence with Feasible Successors
  - + Pre-calculated backup routes which pass Feasibility Condition
- + What if I can't engineer Feasible Successors?
  - + If no backup routes, QUERY message is sent
  - + Convergence now a function of QUERY domain size
- + QUERY domain can be bounded by...
  - + Summarization
  - + Stub router advertisement

# EIGRP Query Scoping

- + QUERY is generated when a route becomes Active
  - + E.g. link loss, neighbor loss, metric change
- + QUERY is sent for exact match of lost route
  - + E.g. do you have a route to 1.2.3.0/24?
- + REPLY is immediately sent if no match
  - + E.g. 1.2.3.0/24 doesn't match my 1.0.0.0/8
  - + This is how summaries limit the QUERY domain



# EIGRP Summarization

- + EIGRP supports two types of summaries
  - + Auto-summary
  - + Manual summary
- + EIGRP hierarchy is arbitrary
  - + Summaries can be enforced anywhere
  - + Big design advantage over OSPF and IS-IS

# EIGRP Auto-Summary

- + EIGRP still supports legacy auto-summary
  - + Advertisements between major network boundaries are summarized to classful boundary
  - + VLSM is supported within the same major network
- + Auto-summary is disabled in newer IOS
  - + Can be enabled with **auto-summary**
  - + Process level in Classic EIGRP
  - + Base topology in Named EIGRP

# EIGRP Manual Summarization

- + Manual summarization
  - + Supports any bit boundary including 0.0.0.0/0
  - + Automatically suppresses subnet advertisements
  - + Can advertise subnets through **leak-map** argument
- + Classic EIGRP configuration
  - + **ip summary-address eigrp** at the link level
- + Named EIGRP configuration
  - + **summary-address** at **af-interface**

# EIGRP Summary Metrics

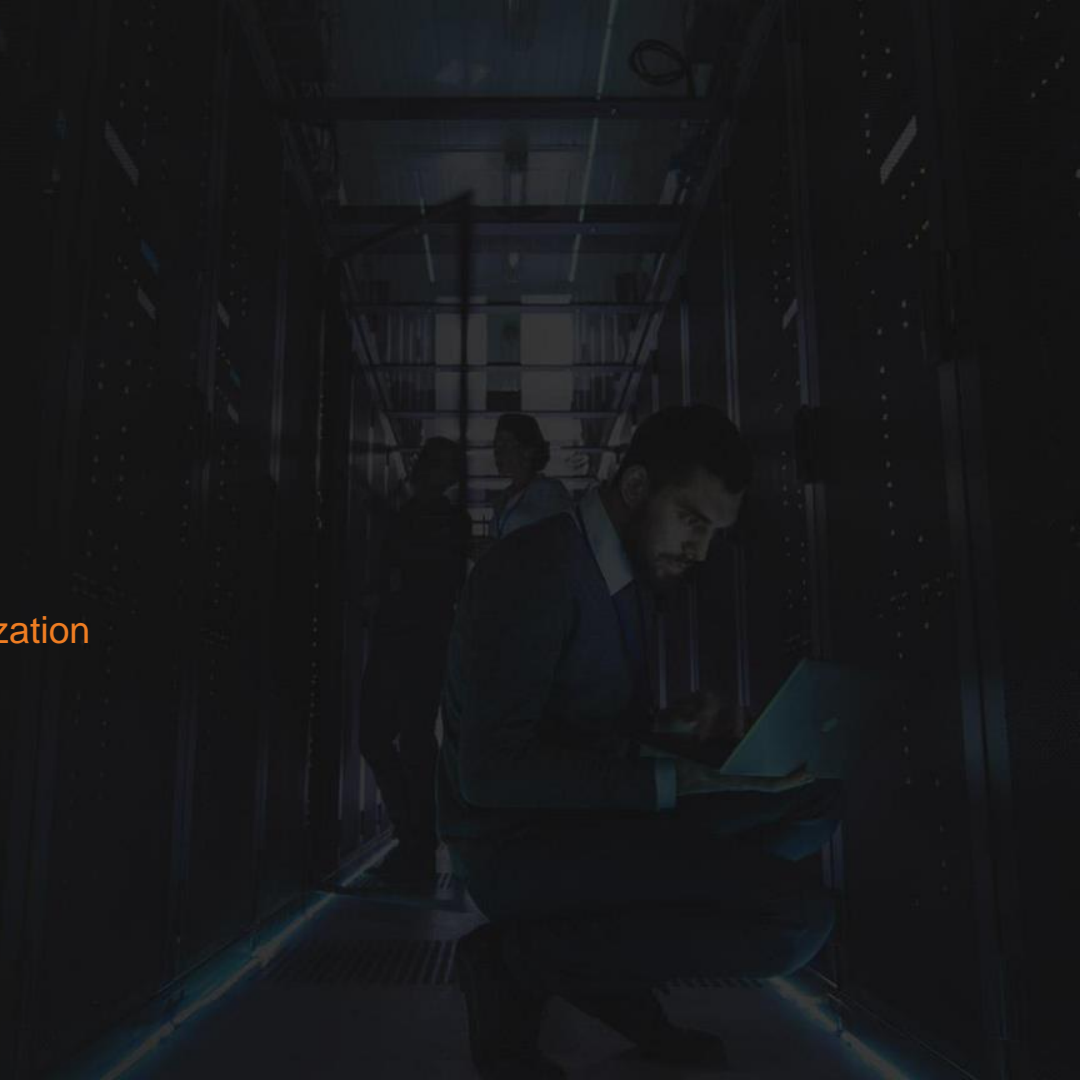
- + EIGRP summary route inherits metrics from its subnets
  - + Subnet with the best metric is used
- + Churning subnets means churning summary
  - + Change of best subnet metric means summary must be re-advertised
- + **summary-metric** can mitigate this
  - + Metric is statically defined per-summary in **topology base**





# EIGRP

EIGRP Traffic Engineering with Summarization



## In This Section

- + EIGRP Traffic Engineering with Summarization
- + EIGRP Leak Maps

# Traffic Engineering with Summarization

- + EIGRP automatically suppresses subnet advertisements of summaries
- + Since EIGRP hierarchy is arbitrary, selective summaries can be used for traffic engineering
  - + Longest match route is always preferred
  - + Implies router performing the summary is less-preferred



# EIGRP Leak Maps

- + Leak Map allows selective advertisement of subnet(s) in a summary
- + Can be used for longest match traffic engineering
- + Applied through a **route-map**
  - + i.e. **match ip address prefix-list...**





# EIGRP

EIGRP over DMVPN



## In This Section

- + EIGRP over DMVPN
  - + Split Horizon
  - + Next Hop Processing

# EIGRP Split-Horizon

- + EIGRP is Distance Vector
  - + Still relies on split horizon rule
- + Split Horizon rule
  - + Don't advertise routes out the interface you received them on
- + Why would you not want Split Horizon?
  - + Partial mesh NBMA
  - + E.g. DMVPN

# EIGRP Next Hop Processing

- + EIGRP updates includes “Next Hop Forwarding Address”
  - + I.e. the next-hop for the route
- + If zero, use IP address from the received IP header
  - + I.e. next-hop is who you learned the route from
- + Why would you not want this?
  - + Third party next hop
  - + E.g. DMVPN Phase 2





# EIGRP

EIGRP Stub Routing





## In This Section

- + EIGRP Stub Routing

# EIGRP Scalability

- + Sub-second convergence with Feasible Successors
  - + Pre-calculated backup routes which pass Feasibility Condition
- + What if I can't engineer Feasible Successors?
  - + If no backup routes, QUERY message is sent
  - + Convergence now a function of QUERY domain size
- + QUERY domain can be bounded by...
  - + Summarization
  - + Stub router advertisement

# EIGRP Stub Router Advertisement

- + Stub routers don't receive QUERY messages
- + Stub routers aren't used for transit
  - + Normally
- + Good design option for DMVPN spokes





# EIGRP

EIGRP Route Filtering



## In This Section

- + EIGRP Filtering Overview
- + EIGRP Filtering Methods
- + EIGRP Filtering Examples

# EIGRP Filtering Overview

- + Like RIP, EIGRP is a Distance Vector IGP
  - + Routers only know what directly connected neighbors tell them
  - + Routers do not need to agree on the topology details like Link State
  - + Implies that filtering can be applied anywhere

# EIGRP Route Filtering Methods

- + EIGRP filtering methods similar to RIPv2 include...
  - + Distribute-Lists
  - + Offset-Lists
  - + Administrative Distance
  - + Passive Interface
- + EIGRP specific filtering methods include...
  - + Distribute-List with Route-Maps
  - + Route Tag Enhancements
  - + Per Neighbor Prefix Limits







# EIGRP

EIGRP IPv6 Routing



# In This Section

- + IPv6 EIGRP Overview
- + IPv6 EIGRP Configuration
  - + Classic Mode
  - + Named Mode

# IPv6 EIGRP Overview

- + Separate control plane from IPv4 EIGRP
  - + Forms separate adjacencies
  - + Maintains separate topology
  - + Performs separate DUAL
- + Cannot advertise IPv4 prefixes
  - + Unlike OSPFv3, IS-IS, BGP

# IPv6 EIGRP Transport

- + Uses IPv6 protocol 88
- + Uses both multicast and unicast
  - + Multicast to FF02::A
  - + Unicast to link-local
- + Multicast can be disabled
  - + **neighbor** to link-local address

# IPv6 EIGRP Prerequisites

- + Enable IPv6 routing
  - + **ipv6 unicast-routing**
- + Ensure valid router-id
  - + Uses IPv4 address format
  - + If no up/up IPv4 interface you must manually specify eigrp router-id

# Configuring Classic Mode IPv6 EIGRP

- + Create global process
  - + **ipv6 router eigrp [AS]**
- + Enable global process
  - + **no shutdown**
  - + Default in some versions
- + Enable IPv6 EIGRP on the link
  - + **ipv6 eigrp [AS]**

# Verifying Classic IPv6 EIGRP

- + Nearly identical to IPv4 EIGRP
  - + **show ipv6 eigrp interface**
  - + **show ipv6 eigrp neighbor**
  - + **show ipv6 eigrp topology [all-links]**



# Configuring Named Mode IPv6 EIGRP

- + Create named process
  - + **router eigrp [name]**
- + Enable IPv6 AFI
  - + **address-family ipv6 unicast autonomous-system [as]**
  - + Does not need to match IPv4 AS
- + Process starts on all interfaces automatically
  - + Can be disabled with **shutdown** under **af-interface**

# Verifying Named Mode IPv6 EIGRP

- + Nearly identical to Named IPv4 EIGRP
  - + **show eigrp address-family ipv6 interface**
  - + **show eigrp address-family ipv6 neighbor**
  - + **show eigrp address-family ipv6 topology [all-links]**
- + Classic EIGRP show commands still supported
  - + **show ipv6 eigrp...**

