Possibilities

#CiscoLive

EIGRP

Introduction and Overview

Steven Moore CCIE #4927, @smoore_bits

DGTL-BRKENT-1102

cisco Live!

#CiscoLive

cisco

Agenda

- Section 1
  - Intro to Distance Vector
  - Basic EIGRP Configuration
- Section 2
  - Neighbors
  - Packets
  - Metrics
- Section 3
  - Topology Table
  - Convergence
  - Stub
  - Summarization
- Section 4
  - Basic Design
  - Event Log
  - EIGRP Redistribution

Agenda

- Introduction
  - Distance Vector Routing
  - Basic Configuration
- Core EIGRP Concepts
  - Peering – Who do we talk to?
  - Packets – What do we communicate?
  - Metrics – How far is too far?

Routing Fundamentals

Routing Protocol Background

- Routing protocols share the same fundamental, essential components.
  - Establish Communication

    Who are they exchanging information with, and how?
  - Exchange Routes

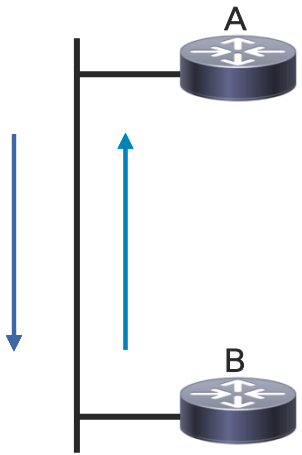
    What information is sent, and how?
  - Perform Computation

    What algorithm is used to compute loop free paths?
  - Route Installation

    What routes are the best? Can we install them?
- EIGRP is no exception!
- Understanding how EIGRP implements each of these will help us learn, use, and operate networks with EIGRP.

Routing Protocol Background

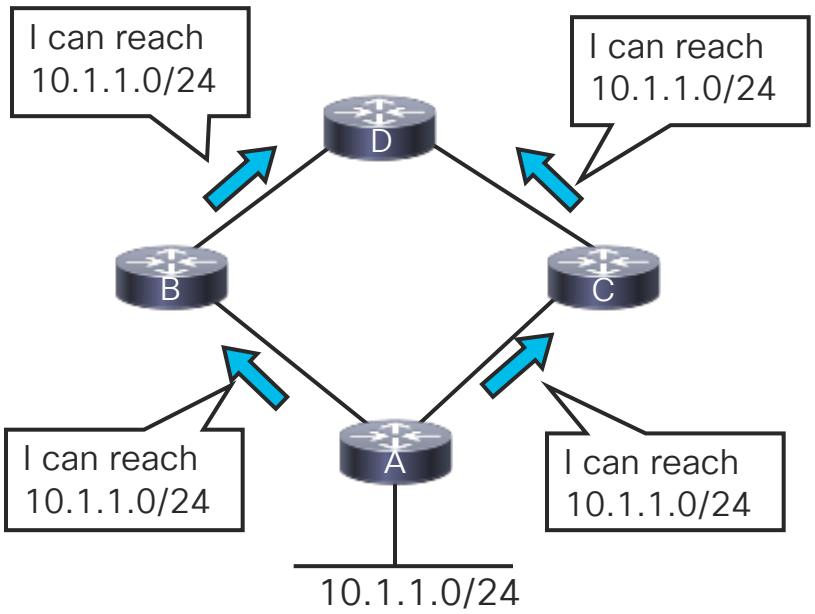
- Logical Sequence of Events
- EIGRP:
  - Peers Form – 3 way handshake
  - Routes Exchanged – Reliable Transport
  - Path Computation – Topology Table, DUAL
  - Routing Table Updated (if necessary)
  - Peers Updated (if necessary)



Distance Vector Routing

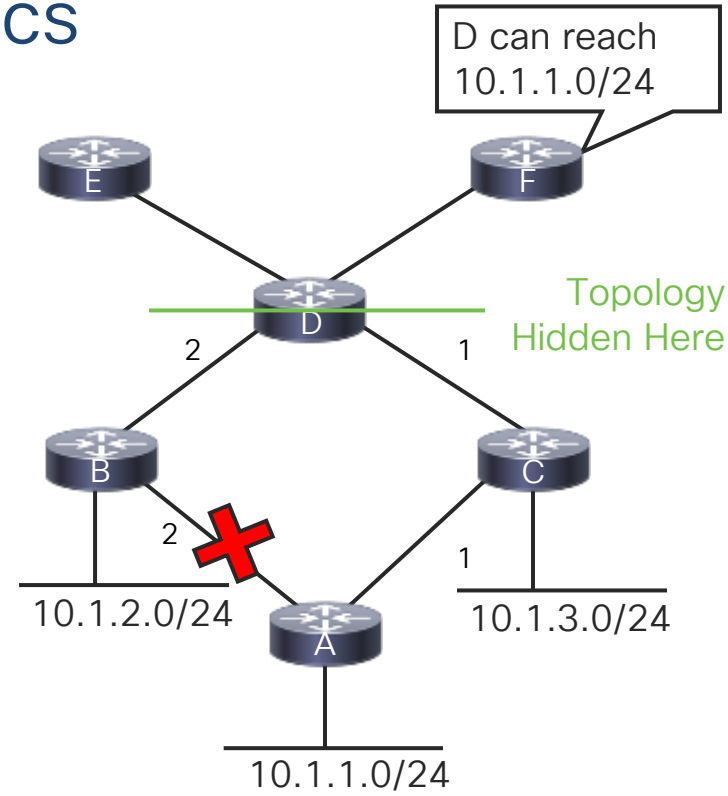
Distance Vector Routing Basics

- Topology information beyond the next hop is naturally hidden in distance vector protocols
- EIGRP only knows prefix and next-hop information
- A advertises that it can reach 10.1.1.0/24
- B and C only advertise to D that they can reach 10.1.1.0/24, not that they are connected to A, which is then connected to 10.1.1.0/24
- D now knows to reach 10.1.1.0/24 it can use B or C, but D does not know what routers or connections exist beyond B and C



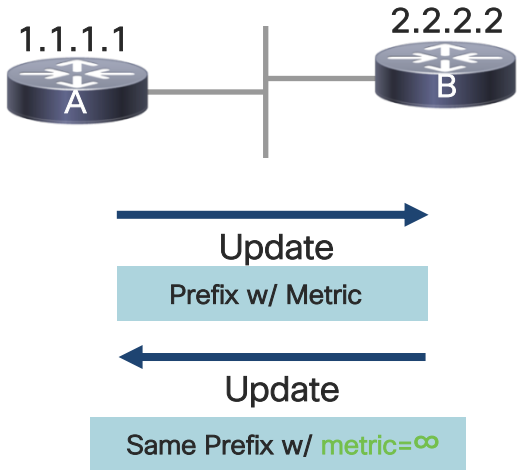
Distance Vector Routing Basics

- Hiding topology information hides information about changes in the topology
- D advertises reachability to 10.1.1.0/24 to E and F
  - If the A to B link fails, D can still reach 10.1.1.0/24 (although the metric might change)
  - If F continues to use D to reach 10.1.1.0/24
  - Does F need to know about the A to B link failure?
  - No!
- What's the issue if D advertises reachability?
  - When the A to B link fails, D will send an update to F
  - F may then go active, and potentially send a Query to its peers
  - This results in increased CPU, memory, and convergence time for a path F can only reach though D



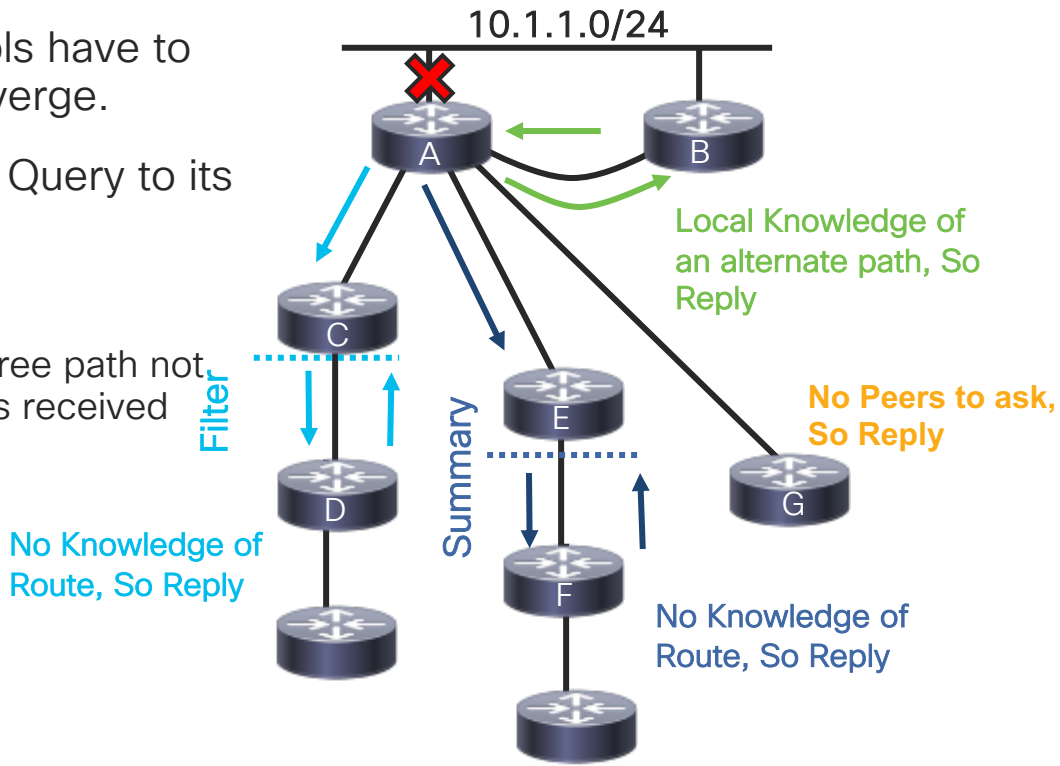
Distance Vector Routing Basics

- Common Concerns – But are they true?
  - Slower Convergence – Why?
  - Count to Infinity – What?
  - Topological “Blindness” – Tradeoffs
- Poison Reverse
- Split Horizon



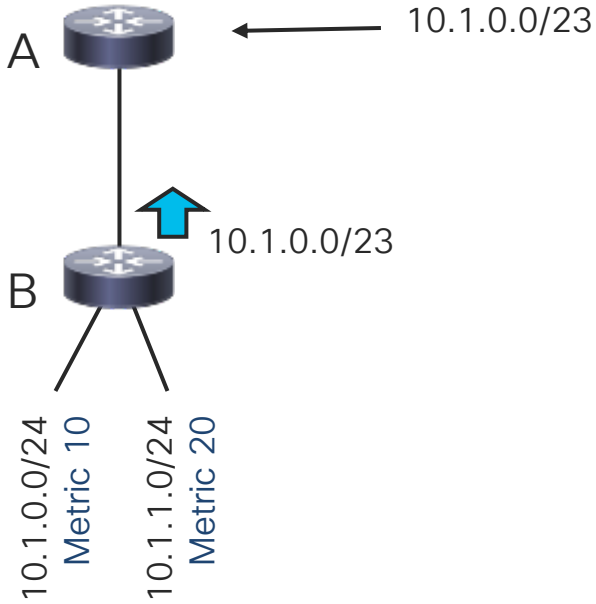
Distance Vector Routing Basics - Resolution

- Without full topological info, protocols have to cooperatively resolve routes to converge.
- When EIGRP goes active, it sends a Query to its peers looking for the lost route.
- The Query is bounded by:
  - Local knowledge of an alternate loop-free path not learned through the peer the query was received from
  - No local knowledge of the route because of filtering
  - No local knowledge of the route because of summarization
  - No peers to query



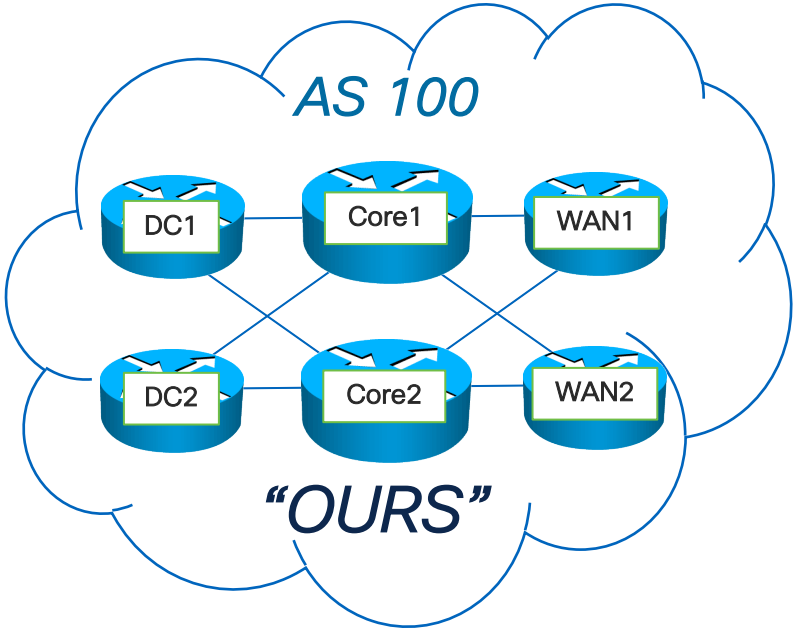
Route Summarization

- A summary:
  - Aggregates information – fewer, less specific routes downstream
  - Hides topology changes by filtering out components
- Only advertise the summary if a component route is present
  - Component is any route falling within the summary address range
  - Component Routes are automatically filtered and not sent downstream



EIGRP Autonomous System

- A collection of devices under the same administrative control
- Shares a consistent routing policy
- Creates the outermost edges of the network



Basic Configuration

Basic EIGRP Configuration

- Classic mode:
  Configuring “router eigrp” command with a number.

```
router eigrp [virtual-instance-name | asystem]
 [no] shutdown
   .
   .
   .
```

- Named / Address Family mode:
  Configuring “router eigrp” command with the virtual-instance-name

```
router eigrp [virtual-instance-name | asystem]
 [no] shutdown
   .
   .
   .
```

Basic EIGRP Configuration – These are the same

- Classic EIGRP

```
interface GigabitEthernet0/0/1
 ip address 10.1.1.1 255.255.255.0
 ip summary-address eigrp 1 10.0.0.0 255.0.0.0

router eigrp 1
 summary-metric 10.0.0.0/8 10000 10 255 0 1500
 network 10.1.1.0 0.0.0.255
 network 10.1.2.0 0.0.0.255
 network 11.2.2.0 0.0.0.255
```

- Address-Family Based

```
router eigrp ROCKS
 !
 address-family ipv4 unicast autonomous-system 1
  !
  af-interface GigabitEthernet0/0/1
   summary-address 10.0.0.0 255.0.0.0
  exit-af-interface
  !
  topology base
   summary-metric 10.0.0.0/8 10000 10 255 0 1500
  exit-af-topology
  network 10.1.1.0 0.0.0.255
  network 10.1.2.0 0.0.0.255
  network 11.2.2.0 0.0.0.255
 exit-address-family
```

Basic EIGRP Configuration

- Named / Address Family mode:
  Configuring “router eigrp” command with the virtual-instance-name
  - Named mode supports both IPv4 and IPv6, and VRF (virtual routing and forwarding) instances
  - Named mode allows you to create a single Instance of EIGRP which can be used for all family types
  - Named mode supports multiple VRFs limited only by available system resources
  - Named mode does not enable EIGRP for IPV4 routing unless configured

```
router eigrp [virtual-instance-name | asystem]
 [no] shutdown
   .
   .
   .
```

EIGRP Configuration – Multiple Address Families

- Single place for all commands needed to completely define an instance.

  "show run | section router eigrp"

- Defines what you're routing/distributing
  - Provide support for both routing (address-family) and services (service-family)
  - Can be configured for VRFs
- Assure subcommands are clear as to their scope
  - Static neighbors, peer-groups, stub, etc, ..
  - neighbor, neighbor remote, etc.

```
router eigrp [virtual-instance-name]
  address-family <protocol> [vrf <name>] autonomous-system <#>
   …
  exit-address-family
  service-family <protocol> [vrf <name>] autonomous-system <#>
   …
  exit-service-family
```

EIGRP Multi-Address Family Support – Interfaces

- EIGRP specific interface properties are configuration in the af-interface mode. for example; authentication, timers, and bandwidth control
- “af-interface default” applies to ALL interfaces
  - Not all commands are supported
- “af-interface <interface>” applies to ONLY one interface
  - Only “eigrp” specific commands are available
  - Properties which are Interface specific, such as delay and bandwidth, are still configured under the interface

```
router eigrp [virtual-instance-name]
  address-family <protocol> autonomous-system <#>
    af-interface default
      …
    exit-af-interface
    af-interface <interface>
      …
    exit-af-interface
  exit-address-family
```

EIGRP Configuration – Multiple Address Families

- Design deployment techniques are the same
  for IPv4 and IPv6
  - Same Route Types (Internal, External, Summary)
  - Configuration and Troubleshooting similar
  - Minimal differences mean reduced training
- Reduced Configuration complexity
  - EIGRP IPv4 and IPv6 can be run concurrently
  - Common IPv4 and IPv6 address configs
  - Each address family has a separate topology table
- Can be phased in, or applied in new deployments

```
router eigrp ROCKS
  address-family ipv4 autonomous-system 1
    network 10.0.0.0 255.0.0.0
!
  address-family ipv4 vrf cisco autonomous-system 2
    network 192.168.0.0
!
  address-family ipv6 autonomous-system 1
    af-interface Ethernet0/0
      shutdown
    exit-af-interface
!
  address-family ipv6 vrf cisco autonomous-system 3
    af-interface default
      no shutdown
  exit-af-interface
```

EIGRP Config – Convert to Named Mode

XE3.10 S

- New CLI added to convert classic mode CLI to Named Mode CLI
  - One-way conversion only
- CLI conversion **only**
  - Does not reset peers
  - Does not reset routes

```
RTR-A#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
RTR-A(config)#do sh run | sec router eigrp
router eigrp 4453
 summary-metric 10.0.0.0/16 10000 10 255 0 1500
 network 10.0.0.0
RTR-A(config-router)#eigrp upgrade-cli ROCKS
Configuration will be converted from router eigrp 4453 to router eigrp
ROCKS.
Are you sure you want to proceed? ? [yes/no]: yes
RTR-A(config)#
*EIGRP: Conversion of router eigrp 4453 to router eigrp ROCKS-
Completed.
RTR-A(config)#do sh run | sec router eigrp
router eigrp ROCKS
 address-family ipv4 unicast autonomous-system 4453
  topology base
   summary-metric 10.0.0.0/16 10000 10 255 0 1500
  exit-af-topology
  network 10.0.0.0
 exit-address-family
```

EIGRP Configuration – More Resources

- Config Guides available on cisco.com, for example:
- https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/17-2/configuration_guide/rtng/b_172_rtng_9500_cg/configuring_eigrp.html

Section 2
Core EIGRP Concepts

Who:
Neighbors

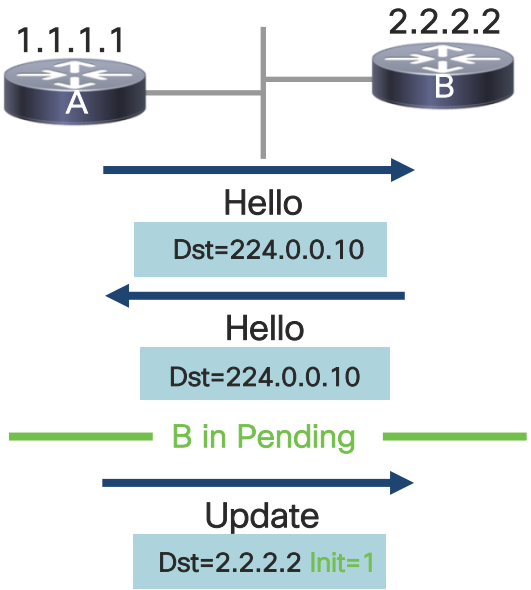
Neighbor Formation

- EIGRP, by default, uses link-local multicast for neighbor communication. (224.0.0.10)
- Neighbors will dynamically form within the same AS, as consistent with other IGP's
- Unicast neighbors are supported but generally not used outside of a few exceptions

How do neighbors get established?

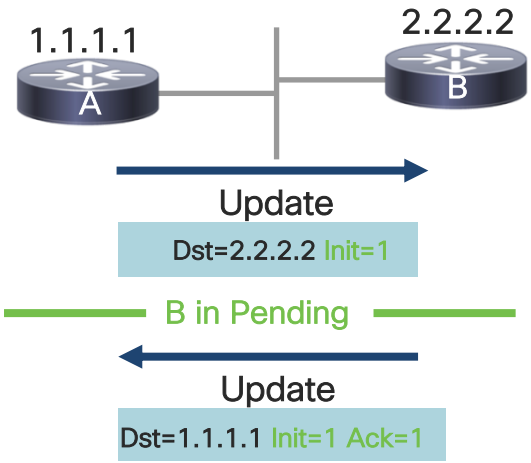
Neighbor Formation

- Router A must receive an initial Hello from B before it will accept reliable packets from it
- When A receives the first Hello from B, it places B in the *pending state*
- A transmits a unicast “NULL” Update which has the initialization (init) bit set, and no routing data
- While A has B is in this state, A will not send it any Queries or other Updates to B



Neighbor Formation

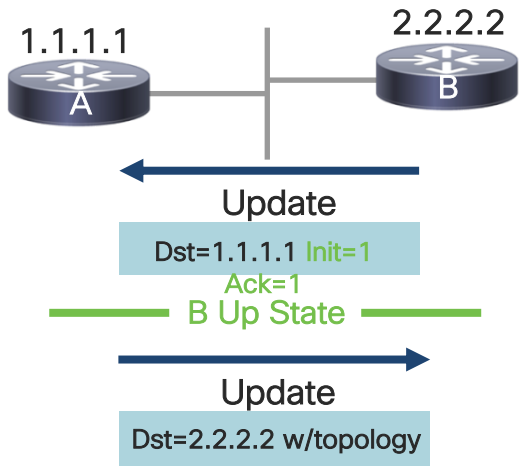
- When B receives this Update with the init bit set, it sends an Update with the init bit set as well
- The acknowledgement (Ack) for A's initial Update is piggybacked onto this Update packet—it is never transmitted by itself
- Thus, there is no way for A to receive the Ack for its initial Update without also receiving B's initial unicast Update



Neighbor Formation

- While waiting on the acknowledgement, Query and Update packets from B are ignored
- If the acknowledgement is never received, A will time B out, and the process will start over
- Once the Ack for its initial update is received, A moves B from *pending state* to *up state*
- A then begins sending it's full topology information to B

✓ This validates bi-directional unicast and multicast communication, completing the 3-Way handshake



Neighbors

- The most useful command for checking neighbor status is **`show ip eigrp neighbors`**
- Some of the important information provided by this command are
  - Hold time—time left that you'll wait for an EIGRP packet from this peer before declaring him down
  - Uptime—how long it's been since the last time this peer was initialized
  - SRTT (Smooth Round Trip Time)—average amount of time it takes to get an Ack for a reliable packet from this peer
  - RTO (Retransmit Time Out)—how long to wait between retransmissions if Acks are not received from this peer

Neighbors

Show IP EIGRP Neighbors

```
RtrA#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address         Interface     Hold  Uptime   SRTT   RTO  Q   Seq
                                        (sec)          (ms) Cnt Num
2   10.1.1.1        Et0           12    6d16h    20     200  0   233
1   10.1.4.3        Et1           13    2w2d     87     522  0   452
0   10.1.4.2        Et1           10    2w2d     85     510  0   3
```

Outstanding Packets

Last Reliable Packet Sent

Seconds Remaining Before Declaring Neighbor Down

How Long Since the Last Time Neighbor Was Discovered

How Long It Takes for This Neighbor to Respond to Reliable Packets

How Long We'll Wait Before Retransmitting if No Acknowledgement

Neighbors – Detail

- The detailed relative of the show ip eigrp neighbor command; some of the additional information available via the detailed version of this command include
  - Number of retransmissions and retries for each neighbor
  - Version of Cisco IOS and EIGRP
  - Stub information (if configured)

```
rtr302-ce1#show ip eigrp neighbor detail
IP-EIGRP neighbors for process 1
H   Address      Interface      Hold  Uptime    SRTT    RTO     Q   Seq  Type
                                (sec)           (ms)         Cnt  Num
1   17.17.17.2   Et1/0           14   00:00:03  394    2364    0   124
   Version 12.0/1.2, Retrans: 0, Retries: 0
   Stub Peer Advertising ( CONNECTED SUMMARY ) Routes
0   50.10.10.1   Et0/0           13   04:04:39   55     330    0    13
   Version 12.0/1.2, Retrans: 2, Retries: 0
```

Neighbors – Interface Detail

```
RtrB#show ip eigrp interface detail
IP-EIGRP interfaces for process 1
                        Xmit Queue    Mean    Pacing Time   Multicast    Pending
Interface    Peers      Un/Reliable   SRTT    Un/Reliable   Flow Timer   Routes
Et0/0          1            0/0       737        0/10          5376        0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/3  Un/reliable ucasts: 6/3
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
Et1/0          1            0/0       885        0/10          6480        0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/2  Un/reliable ucasts: 5/3
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
```

Neighbors – Interface Detail

- There is also a show ip eigrp interface which contains a subset of this info; You may want to just use that if you don't need all the detail
- This command supplies a lot of information about how the interfaces are being used and how well they are obeying; some of the interesting information available via this command is:
  - Retransmissions sent—this shows how many times EIGRP has had to retransmit packets on this interface, indicating that it didn't get an ack for a reliable packet; having retransmits is not terrible, but if this number is a large percentage of packets sent on this interface, something is keeping neighbors from receiving (and acking) reliable packets
  - Out-of-sequence rcvd—this shows how often packets are received out of order, which should be a relatively unusual occurrence; again, it's nothing to worry about if you get occasional out-of-order packets since the underlying delivery mechanism is best-effort—if the number is a large percentage of packets sent on the interface, however, then you may want to look into what's happening on the interface—are there errors?
- You can also use this command to see if an interface only contains stub neighbors and if authentication is enabled

Neighbors

- EIGRP Log-Neighbor-Changes is on by default since 12.2(12)
- Turn it on and leave it on
- Best to send to buffer log

```
RtrA# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
RtrA(config) # router eigrp 1
RtrA(config-router) # eigrp log-neighbor-changes
RtrA(config-router) # logging buffered 10000
RtrA(config) # service timestamps log datetime msec
```

Neighbors

- EIGRP log-neighbor-changes is the best tool you have to understand why neighbor relationships are not stable. It should be enabled on every router in your network—CSCdx67706 (12.2(12)) made it the default behavior; as explained on the previous slide, the uptime value from show ip eigrp neighbors will tell you the last time a neighbor bounced, but not how often or why—with log-neighbor-changes on and logging buffered, you keep not only a history of when neighbors have been reset, but the reason why… absolutely invaluable
- Logging buffered is also recommended, because logging to a syslog server is not bulletproof; for example, if the neighbor bouncing is between the router losing neighbors and the syslog server, the messages could be lost—it's best to keep these types of messages locally on the router, in addition to the syslog server
- It may also be useful to increase the size of the buffer log in order to capture a greater duration of error messages—you would hate to lose the EIGRP neighbor messages because of flapping links filling the buffer log; if you aren't starved for memory, change the buffer log size using the command logging buffered 10000 in configuration mode
- The service timestamps command above puts more granular timestamps in the log, so it's easier to tell when the neighbor stability problems occurred

Neighbors – What about when they don't work?

- Log-Neighbor-Changes Messages
- So this tells us why the neighbor is bouncing—but what do they mean?
- Hint: peer restarted means you have to ask the peer; he's the one that restarted the session

```
Neighbor 10.1.1.1 (Ethernet0) is down: peer restarted
Neighbor 10.1.1.1 (Ethernet0) is up: new adjacency
Neighbor 10.1.1.1 (Ethernet0) is down: holding time expired
Neighbor 10.1.1.1 (Ethernet0) is down: retry limit exceeded

Sometimes others, but not often
```

Manual Changes

- Some manual configuration changes can also reset EIGRP neighbors, depending on the Cisco IOS version
  - Summary changes (manual and auto)
  - Route filter changes
  - Stub setting changes
- This is normal behavior for much older code
  - CSCdy20284 removed many of these neighbor resets
    - Implemented in 12.2S, 12.3T, and 12.4 (approximately 2005)
- Mismatch of K-values (metric weights) will prevent peers from forming also (best just not to change them at all).

Manual Changes

- Summary changes
  - When a summary changes on an interface, components of the summary may need to be removed from any neighbors reached through that interface; neighbors through that interface are reset to synch up topology entries
- Route filter changes
  - Similar to summary explanation above; neighbors are bounced if a distribute-list is added/removed/changed on an interface in order to synch up topology entries
- In the past, we also bounced neighbors when interface metric info changed (delay, bandwidth), but we no longer do that (CSCdp08764)
- CSCdy20284 was implemented to stop bouncing neighbors when many manual changes occur; in late 12.2S, 12.3T, and 12.4, summary and filter changes no longer bounce neighbors
- Changing Stub setting or router-ids still resets peers! Remember to make these changes during maintenance windows

What:
Updates & Advertisements

Updates & Advertisements

EIGRP Sends both reliable and Unreliable Packet types

- Unreliable packets are:
  - Hellos
  - Acknowledgement
- Reliable packets are:
  - Updates
  - Queries
  - Replies
  - SIA-Queries
  - SIA-Replies
- Reliable packets are sequenced, require an acknowledgement, and are retransmitted up to 16 times if not acknowledged

EIGRP Packets

5 Basic Packet Types

EIGRP uses five different packet types to handle session management and pass DUAL Message types:

- HELLO Packets (includes ACK)
- QUERY Packets (includes SIA-Query)
- REPLY Packets (includes SIA-Reply)
- REQUEST Packets
- UPDATE Packets

EIGRP packets are directly encapsulated into a network-layer protocol, such as IPv4 or IPv6. While EIGRP is capable of using additional encapsulation (such as AppleTalk, IPX, etc.) no further encapsulation is specified in this document. (RFC-7868)

Updates & Advertisements

5 Basic Packet Types

- Hello/Acks
  Hellos are used for peer discovery/maintenance. They do not require acknowledgment. A hello packet with a non-zero sequence number is also used as an acknowledgment (ack). Hellos are normally multicast and Acks are always sent using a unicast address

- Updates
  Updates are used to convey reachability of destinations. When a new peer is discovered, update packets are sent so the peer can populate its topology table. In some cases, update packets will be unicast. In other cases, such as a link cost change, updates are multicast. Updates are always transmitted reliably

- Query/Reply
  Queries are sent when destinations go into Active state. Queries are normally multicast to all peers on all interfaces except for the interface to the previous successor. If a receiving peer does not have an alternative path to the destination, it will in turn Query its peers until the *query boundary* is reached. Once the Query is sent, the router must wait for all the Replies from all peers before it can compute a new successor. Replies are sent containing the answer (which may be a valid metric or infinity/not reachable) and are unicast to the originator of the query. Both queries and replies are transmitted reliably

- SIA-Query/SIA-Reply
  If any peer fails to Reply to a Query, the destination is said to be Stuck In Active (SIA), and the peer may be reset. At ½ the SIA time (default 90 seconds) the router will send an SIA-Query to the non-replying peer. The peer must send either an SIA-Reply indicating the destination is still active, or a Reply. Both SIA-Queries and SIA-Replies are transmitted reliably

EIGRP Packets – Basic Encoding

		31	16	15	0
UPDATE	**Packet Header Section**	Header Version	Opcode	Checksum	
REQUEST		Flags			
QUERY		Sequence Number			
REPLY		ACK Number			
HELLO		VRID		Autonomous System	
ACK	**TLV Data**	TLV Encoding (variable length)			
SIAQUERY					
SIAREPLY					

Flags

Init Flag (0x01) - This bit is set in the initial update packet sent to a newly discovered peer. It requests the peer to download a full set of routes

CR Flag (0x02) - This bit indicates that receivers should only accept the packet if they are in Conditionally Received mode

RS (0x04) - The Restart flag is set in the Hello and the Init update packets during the NSF signaling period

EOT (0x08) - The End-of-Table flag marks the end of the startup updates sent to a new peer

EIGRP Packets – Basic Encoding

- **Header Version** – EIGRP Packet Header Format version. Current Version is 2. This field is not the same as the TLV Version field.
- **Opcode** – EIGRP opcode indicating function packet serves.
- **Checksum** – Each packet will include a checksum for the entire contents of the packet. The check-sum will be the standard ones complement of the ones complement sum. The packet is discarded if the packet checksum fails.
- **Flags** – Defines special handling of the packet.
- **Sequence** – 32-bit sequence number. Each packet that is transmitted will have a unique sequence number with respect to a sending router. A value of 0 means that an acknowledgment is not required.
- **Ack** – 32-bit sequence number. Acknowledgment number with respect to receiver of the packet. If the value is 0, there is no acknowledgment present. A non-zero value can only be present in unicast addressed packets. *A Hello packet with a nonzero ACK field should be decoded as an ACK packet rather than a Hello packet.*
- **VRID** – Virtual Router ID. 16-bit unsigned number, which identifies the virtual network this this packet, is associated. Packets received with an unknown, or unsupported VRID will be discarded.
- **AS number** – Autonomous System – 16 bit unsigned number of the sending system. A router that receives a packet from a peer must have the same AS number or the packet is ignored.

EIGRP Packets – Generic TLV Encoding

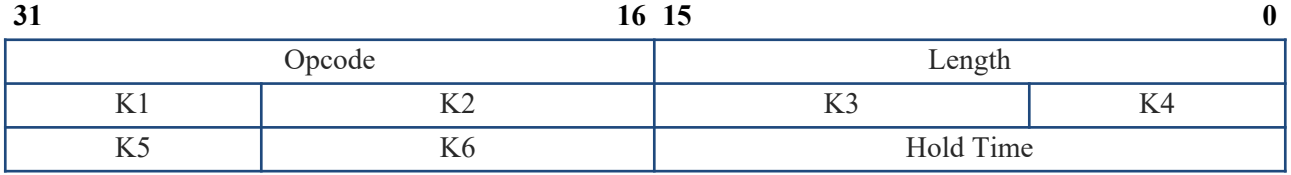
	31 16	15 0
	Type	Length
PARAMETER	Vector Data (Variable)	
AUTHENTICATION	0x0002	0x0002
SEQUENCE	0x00	
SOFTWARE VERSION	0x00	
MULTICAST SEQUENCE	0x00	
PEER INFORMATION	0x00	
PEER TERMINATION	0x00	
PEER TID LIST	---	

Protocol	ID	VERSION
General	0x0000	Classic
IPv4	0x0100	Classic
AppleTalk	0x0200	Classic
IPX	0x0300	Classic
IPv6	0x0400	Classic
Based on AFI	0x0600	Multi-Protocol

- Generic TLVs apply to **all** address and service families
- Length includes the Type and Length fields
- Vector data is variable length

EIGRP Packets – Parameter TLV Encoding

- The Hello packet may carry the Parameter TLV to indicate the default coefficients(K-values) should not used for computing the composite metric.



Opcode– transmitted as 0x0001,
Length – transmitted as 0x000C
Number bytes in the Vector of the TLV. . Currently transmitted as 4
K-values - The K-values associated with the EIGRP composite metric equation. The default values for weights are:
Hold Time - The amount of time in seconds that a receiving router should consider the sending peer valid.

Default K values are: $K_1 == K_3 == 1$ and $K_2 == K_4 == K_5 == K_6 == 0$

EIGRP Packets – Classic TLV Metric Encoding

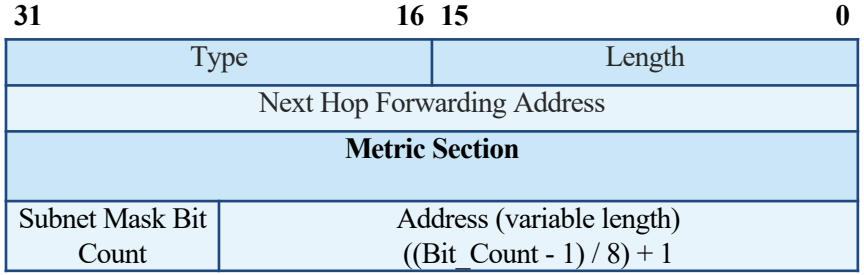
	31	16	15	0
Metric Section	Scaled Delay			
	Scaled Bandwidth			
	MTU			Hop Count
	Reliability	Load	Internal Tag	Opaque

- **Scaled Delay** – Accumulative delay along an unloaded path to the destination. Expressed in units of 10 µsec•256
- **Scaled Bandwidth** - Path bandwidth measured in bits per second In units of 10,000,000/kilobits per second •256

- **MTU** - The minimum maximum transmission unit size for the path to the destination
- **Hop Count** - The number of router traversals to the destination
- **Reliability** - The current error rate for the path. Measured as an error percentage. A value of 255 indicates 100% reliability
- **Load** - The load utilization of the path to the destination. Measured as a percentage of load. A value of 255 indicates 100% load
- **Internal-Tag** - A tag assigned by the network administrator that is untouched by EIGRP

EIGRP Packets – Classic TLV Internal Routes

- Used For:
  - Internal IPv4 prefixes - TLV (Type 0x0102)
  - Internal IPv6 prefixes - TLV (Type 0x0402)

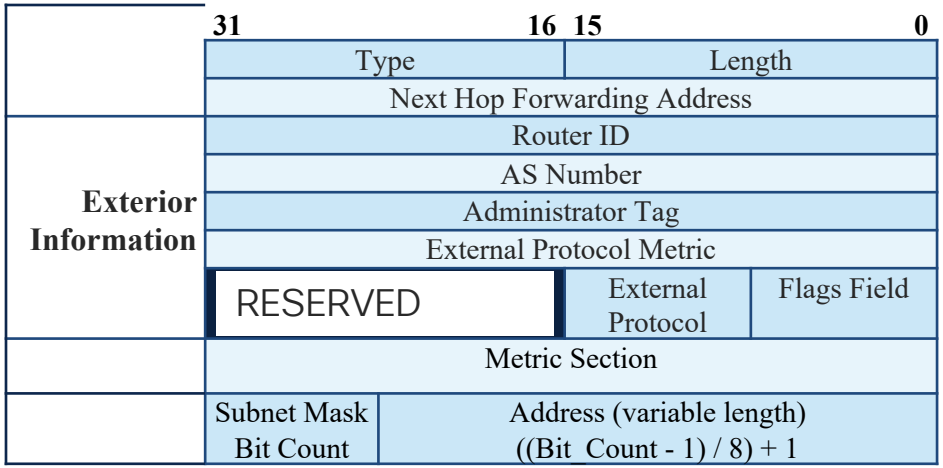


- Next Hop Forwarding Address – Specific address to use for the destination's nexthop. If the value is 0, the source address of the sending router is used as the next-hop for the route
- Metric Section – Accumulative metric for destinations contained in this TLV
- Destination Section - The protocol specific address being sent

EIGRP Packets – Classic TLV External Routes

- Used For:
  - External IPv4 prefixes - TLV (Type 0x0103)
  - External IPv6 prefixes - TLV (Type 0x0403)
- External Protocol:

Protocols	Value
IGRP	1
EIGRP	2
Static	3
RIP	4
HELLO	5
OSPF	6
ISIS	7
EGP	8
BGP	9
IDRP	10
Connected	11



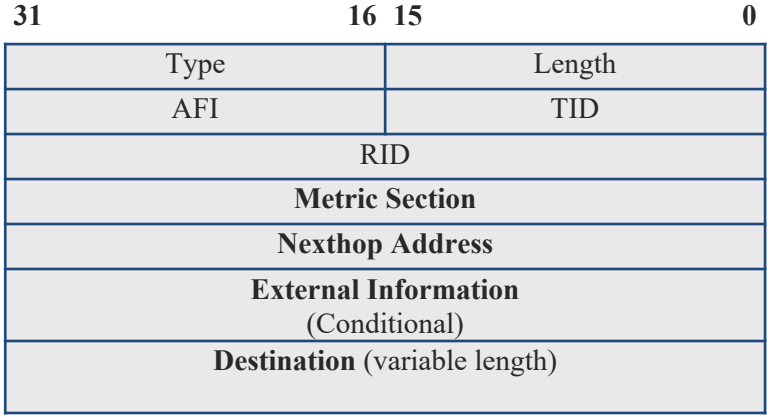
- Flags:
  - Candidate Default (Bit 1) - If set, this destination should be regarded as a candidate for the default route.

EIGRP Packets – Classic TLV External Routes

- **Next Hop Forwarding Address** – The IP address to store in the routing table as the next hop for the destinations described in this packet.
- **Router ID** – The IP address of the router that has redistributed this external route into the EIGRP autonomous system. The address should be the largest unsigned address of any inter-face IP address.
- **AS Number** – The autonomous system number that the route resides in.
- **Administrator Tag** – A tag assigned by the network administrator that is untouched by EIGRP. This allows a network administrator to filter routes in other EIGRP border routers based on this value.
- **External Protocol Metric** – The value of the composite metric which resides in the routing table as learned by the foreign protocol. If the External Protocol is IGRP or EIGRP from another routing process, the value can optionally be the composite metric or 0, and the metric information is stored in the metric section.
- **External Protocol** – Defines the external protocol that this route was learned by. The following values are assigned:
- Flags Field
- **Internal** (Bit 0) – Indicates if the destination is considered in the same autonomous system. If so, the internal flag will be set and the AS Number will be assigned to be the AS that the route will be redistributed into.
- **CD** (Candidate Default – Bit 1) – If set, this destination should be regarded as a candidate for the default route. An EIGRP default route is selected from all the advertised candidate default routes with the smallest metric.

EIGRP Packets – Multi-Protocol TLV Encoding

- Common Type Code for all Protocols
  - All Internal prefixes – TLV (Type 0x0602)
  - All External prefixes – TLV (Type 0x0603)
- All attributes are organized independent of the transport, or destination address/service family
- Optional Metric Attributes:
  - Extended Metrics
  - Extended Community Tags
  - Exterior Information



- **Address Family Identifier (AFI)** - defines the IANA type and format for the destination data.
- **Topology ID (TID) –** 16bit number used to identify a specific sub-topology the prefixes is associated with
- **Router ID (RID)** – A unique 32bit number that identifies the router sourcing the route
- **Destination Section** - The address, as defined by the AFI, being sent

EIGRP Packets – Multi-Protocol TLV Metrics

- **Offset** – Number of 16bit words in the Extended Attribute section; used to determine the start of the destination information. If no Extended Attributes are attached, offset will be zero.
- **Priority**: Priority of the prefix when transmitting a group of destination addresses to neighboring routers. A priority of zero indicates no priority is set. Currently transmitted as 0
- **Reliability** – The current error rate for the path. Measured as an error percentage. A value of 255 indicates 100% reliability
- **Load** – The load utilization of the path to the destination. Measured as a percentage of load. A value of 255 indicates 100% load.
- **MTU** – The minimum maximum transmission unit size for the path to the destination. Not used in metric calculation, but available to underlying protocols
- **Hop Count** – The number of router traversals to the destination.
- **Delay** – The one-way latency along an unloaded path to the destination expressed in units of picoseconds per kilobit.
- **Bandwidth** – The path bandwidth measured in kilobit per second as presented by the interface.
- **Reserved** – Transmitted as 0x0000
- **Opaque Flags** – 16 bit protocol specific flags.
- **Extended Attributes** – (Optional) When present, defines extendable per destination attributes.

EIGRP Packets – Multi-Protocol Extended Metrics

- Extended Metrics is a way to influence routing decisions though the use of non-traditional vectors
- Currently there are 6 Extended Metric opcodes defined

Value	Description	Type
1	Scaled Metric	informational
2	Administrator Tag	Policy/Filtering
3	Extended Community Tag(s)	Policy/Filtering
4	Jitter	Metric Modifier
5	Quiescent Energy	Metric Modifier
6	Energy	Metric Modifier
7	Add Path	Nexthop Selection

- Extended Metric vectors that modify the composite metric, are added controlled by the composite K_6

EIGRP Packets – Multi-Protocol TLV Metrics

	31	16	15	0
Metric Section	Offset	Priority	Reliability	Load
	MTU			Hop Count
	Latency			
	Throughput			
	Reserved		Opaque Flags	
	Extended Metrics			

- Multi-Protocol metrics, which support the new "Wide Metrics" transmits Latency and Throughput values un-scaled
- In addition to the Classic vector metrics, there are new fields not previously available:

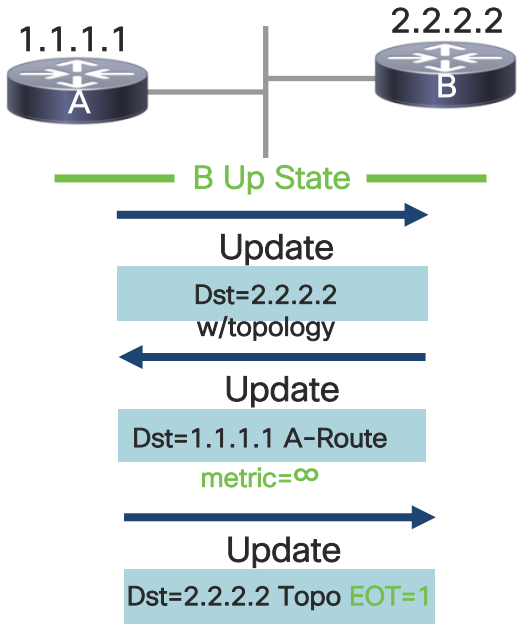
Priority: Priority of the prefix when transmitting a group of destination addresses to neighboring routers. A priority of zero indicates no priority is set. Currently transmitted as 0

Extended Metrics – When present, defines extendable per destination attributes. This field is not normally transmitted

Updates & Advertisements

Advertisement

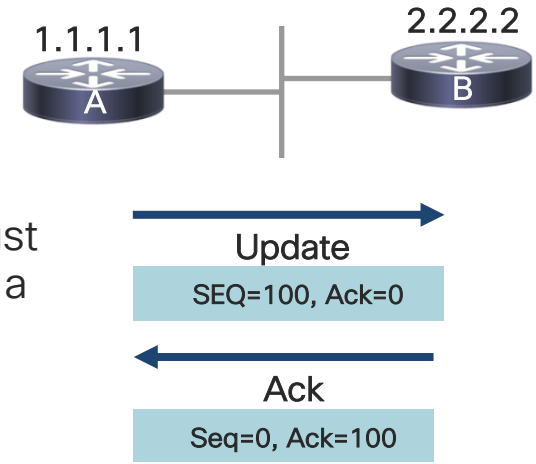
- For each route A sends to B, B sends a poison reverse
- This makes certain the two routers tables are accurate as well as making sure other routers on the interface they share use the right path for each prefix
- When a router finishes sending its table, it sends an end-of-table indicator



Updates & Advertisements

Sequence Numbers and Acks

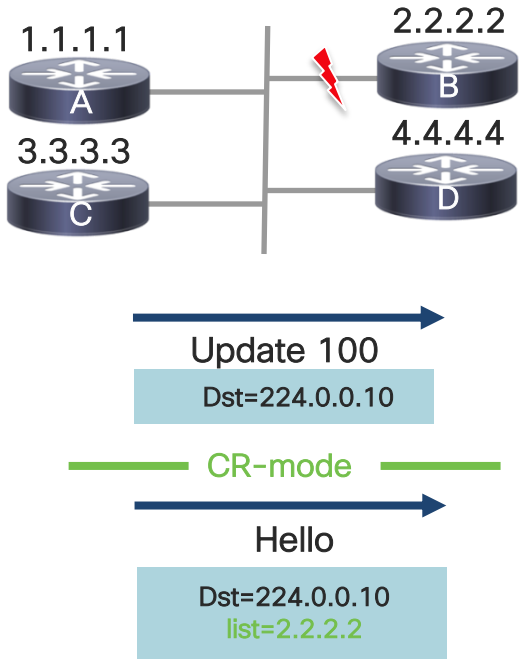
- An Update packet transmitted contains a sequence number that is acknowledged by a receipt of a Ack packet
- If the Update or the Ack packet is lost on the network, the Update packet will be retransmitted
- In the case of the Query packets, the Query packet also must be acknowledged (“I heard the question”, later followed by a Reply packet (“Here is the answer”).
  - Note that both responses are required and perform different functions
- Replies also contain sequence numbers and must be acknowledged



Updates & Advertisements

Conditional Receive (CR-mode)

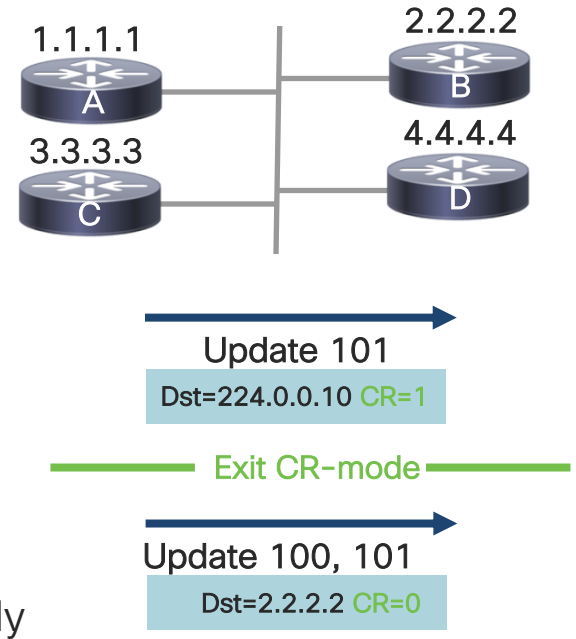
- A sends a multicast Update packet
- B, C and D receive the Update and send an acknowledgment
- B's acknowledgments is lost on the network
- Before the retransmission timer expires, A has an event that requires it to send a new multicast update on this interface
- A detects that B has not Acked the last packet and enters the Conditional Receive process
- A builds a Hello packet with a SEQUENCE TLV containing B's address
- This special Hello packet is multicasted unreliably out the interface



Updates & Advertisements

Conditional Receive (CR-mode)

- C and D process the special Hello packet looking for their address in the list. If not found, they put themselves in Conditional Receive (CR-mode) mode
- Any subsequent reliable packets received on C and D with the CR-flag set are accepted and processed
- B does not put itself in CR-mode because it finds its address in the list
- Reliable packets received by B with the CR-flag must be discarded and not acknowledged
- Once A has sent the CR Update(s), it exits CR-mode
- A will unicast the previous, unacknowledged packets directly to B



Updates & Advertisements

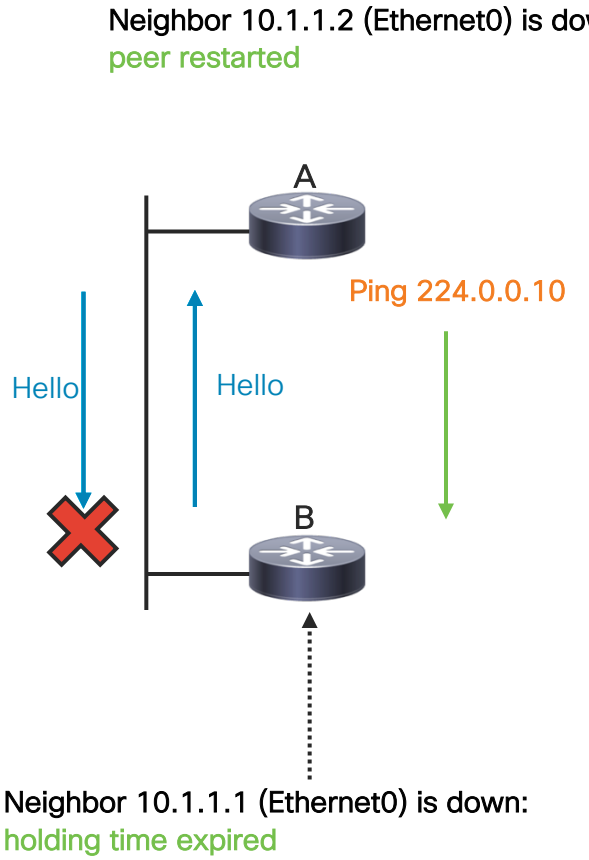
- What else can go wrong?
- How do we observe?
- What does EIGRP do?
- What should you do?

Log-Neighbor-Changes Messages

- Peer restarted—the other router reset our neighbor relationship; you need to go to that device to see why it thought our relationship had to be reset
- New adjacency—established a new neighbor relationship with this neighbor; happens at initial startup and after recovering from a neighbor going down
- Holding time expired—we didn't hear any EIGRP packets from this neighbor for the duration of the hold time; this is typically 15 seconds for most media (180 seconds for low-speed NBMA)
- Retry limit exceeded—this neighbor didn't acknowledge a reliable packet after at least 16 retransmissions (actual duration of retransmissions is also based on the hold time, but there were at least 16 attempts)

Holding Time Expired

- The holding time expires when an EIGRP packet is not received during hold time
  - Typically caused by congestion or physical errors
- Next Step:
- Ping the multicast address (224.0.0.10) from the **other** router
  - If there are a lot of interfaces or neighbors, you should use extended ping and specify the source address or interface



Holding Time Expired

- When an EIGRP packet is received from a neighbor, the hold timer for that neighbor resets to the hold time supplied in that neighbor's hello packet, then the value begins decrementing
  - The hold timer for each neighbor is reset back to the hold time when each EIGRP packet is received from that neighbor (long ago and far way, it needed to be a hello received, but now any EIGRP packet will reset the timer)
  - Since hellos are sent every five seconds on most networks, the hold time value in a show ip eigrp neighbors is normally between 10 and 15 (resetting to hold time (15), decrementing to hold time minus hello interval or less, then going back to hold time)
- Why would a router not see EIGRP packets from a neighbor?
  - It may be gone (crashed, powered off, disconnected, etc.)
  - It (or we) may be overly congested (input/output queue drops, etc.)
  - Network between us may be dropping packets (CRC errors, frame errors, excessive collisions)

Holding Time Expired

- Another troubleshooting tool available is to do the command “debug eigrp packet hello”; this will produce debug output to the console or buffer log (depending on how you have it configured) that will show the frequency of hellos sent and received
- You should make sure you have the timestamps for the debugs set to a value that you can actually see the frequency; something like:
  - `service timestamps debug datetime msec`
- Remember that any time you enable a debug on a production router, you are taking a calculated risk; it’s always better to use all of the safer troubleshooting techniques before resorting to debugs—sometimes they’re necessary, however

Holding Time Expired

```
RtrA# debug eigrp packet hello
EIGRP Packets debugging is on (HELLO)
19:08:38.521: EIGRP: Sending HELLO on Serial1/1
19:08:38.521:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
19:08:38.869: EIGRP: Received HELLO on Serial1/1 nbr 10.1.6.2
19:08:38.869:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
19:08:39.081: EIGRP: Sending HELLO on FastEthernet0/0
19:08:39.081:   AS 1, Fags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
```

Remember—Any Debug Can Be Hazardous

Retry Limit Exceeded

- EIGRP sends both unreliable and reliable packets
  - Hellos and Acks are unreliable
  - Updates, Queries, Replies, SIA-queries and SIA-replies are reliable
- Reliable packets are sequenced and require an acknowledgement
  - Reliable packets are retransmitted up to 16 times if not acknowledged

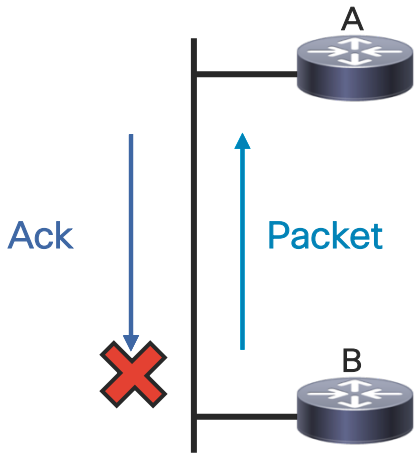
Retry Limit Exceeded

- Exceeding the retry limit means that we're sending reliable packets which are not getting acknowledged by a neighbor—when a reliable packet is sent to a neighbor, it must respond with a unicast acknowledgement; if a router is sending reliable packets and not getting acknowledgements, one of two things are probably happening
  - The reliable packet is not being delivered to the neighbor
  - The acknowledgement from the neighbor is not being delivered to the sender of the reliable packet
- These errors are normally due to problems with delivery of packets, either on the link between the routers or in the routers themselves—congestion, errors, and other problems can all keep unicast packets from being delivered properly; look for queue drops, errors, etc., when the problem occurs, and try to ping the unicast address of the neighbor to see if unicasts in general are broken or whether the problem is specific to EIGRP

Retry Limit Exceeded

- Reliable packets are re-sent after Retransmit Time Out (RTO)
  - Typically 6 x Smooth Round Trip Time (SRTT)
  - Minimum 100 ms
  - Maximum 5000 ms (five seconds)
  - 16 retransmits takes between roughly 40 and 80 seconds
- If a reliable packet is not acknowledged before 16 retransmissions and the hold time has not expired, re-initialize the neighbor

Neighbor 10.1.1.2 (Ethernet0) is down: peer restarted



Neighbor 10.1.1.1 (Ethernet0) is down: retry limit exceeded

Retry Limit Exceeded

- The Retransmit Timeout (RTO) is used to determine when to retry sending a packet when an Ack has not been received, and is (generally) based on 6 X Smooth Round Trip Time (SRTT); the SRTT is derived from previous measurements of how long it took to get an Ack from this neighbor—the minimum RTO is 100 Msec and the maximum is 5000 Msec; each retry backs off 1.5 times the last interval
- The minimum time required for 16 retransmits is approximately 40 seconds (minimum interval of 100 ms with a max interval of 5000 ms); for example, If there isn't an acknowledgement after 100 ms, the packet is retransmitted and we set a timer for 150 ms—if it expires, we send it again and set the timer for 225 ms, then 337 ms, etc., until 5000 ms is reached; 5000 ms is then repeated until a total of 16 retransmissions have been sent
- The maximum time for 16 retransmits is approximately 80 seconds, if the initial retry is 5000 ms and all subsequent retries are also 5000 ms

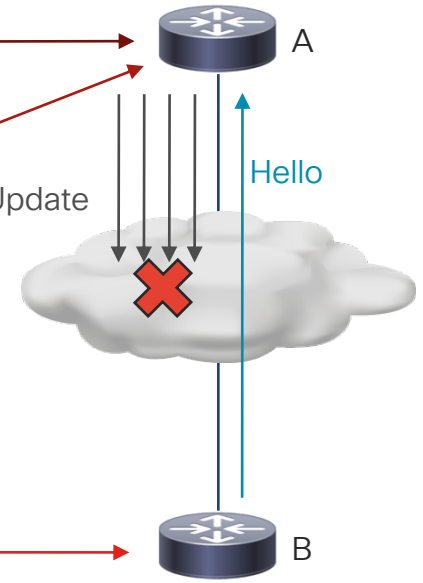
Retry Limit Exceeded

- If a reliable packet is retransmitted 16 times without an acknowledgement, EIGRP checks to see if the duration of the retries has reached the hold time, as well
- Since the hold time is typically 15 sec on anything but low-speed NBMA, it normally isn't a factor in the retry limit; NBMA links that are T1 or less, however, wait an additional period of time after re-trying 16 times, until the hold-time period (180 seconds) has been reached before declaring a neighbor down due to retry limit exceeded
- This was done to give the low-speed NBMA networks every possible chance to get the Acks across before downing the neighbor
- Remember this if you modify the hold times!

Unidirectional Links

```
RtrB#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H Address      Interface  Hold   Q     Seq
                                 Cnt   Num

1 10.1.102.2      Et0        14     4       0
```




```
%DUAL-5-NBRCHANGE: IP-EIGRP 1: Neighbor 10.1.5.4 (Serial1) is down:
retry limit exceeded
```

```
RtrA#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
RtrA#
```

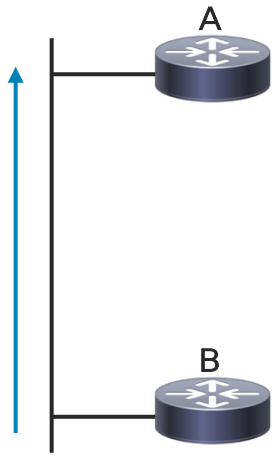
Unidirectional Links

- In this example, we see what happens when a link is only working in one direction; unidirectional links can occur because of a duplicate IP address, a wedged input queue, link errors, or any other reason you can think of that would allow packets to be delivered only in one direction on a link
- RtrB doesn't even realize that RtrA exists—RtrA is sending out his hellos, waiting for a neighbor to show up on the network; what it doesn't realize is that the RtrB is already out there and trying to bring up the neighbor relationship
- RtrB, on the other hand, sees the hellos from RtrA, sends his own hellos and then sends an update to RtrA to try to get their topology tables/routing tables populated—unfortunately, since the updates are also not being received by RtrA, it of course isn't sending acknowledgements; RtrB tries it 16 times and then resets his relationship with RtrA and starts over
- You'll spot this symptom by the retry limit exceeded messages on RtrB, RtrB having RtrA in his neighbor table with a continual Q count, and RtrA not seeing RtrB, at all
- CSCdy45118 has been implemented to create a reliable neighbor establishment process (three-way handshake) and reliable neighbor maintenance (neighbor taken down more quickly when unidirectional link encountered). 12.2T, 12.3 and up

Retry Limit Exceeded

- Ping the neighbor's unicast address
  - Vary the packet size
  - Try large numbers of packets
- This ping can be issued from either neighbor; the results should be the same
- Common causes
  - Mismatched MTU (check for giants)
  - Unidirectional link
  - Dirty link (check show interface for errors)

```
RtrB# ping
Protocol[ip]:
Target IP address: 10.1.1.1
Repeat count [5]: 100
Datagram Size: 1500
Timeout in seconds[2]:
Extended commands[n]: y
....
```

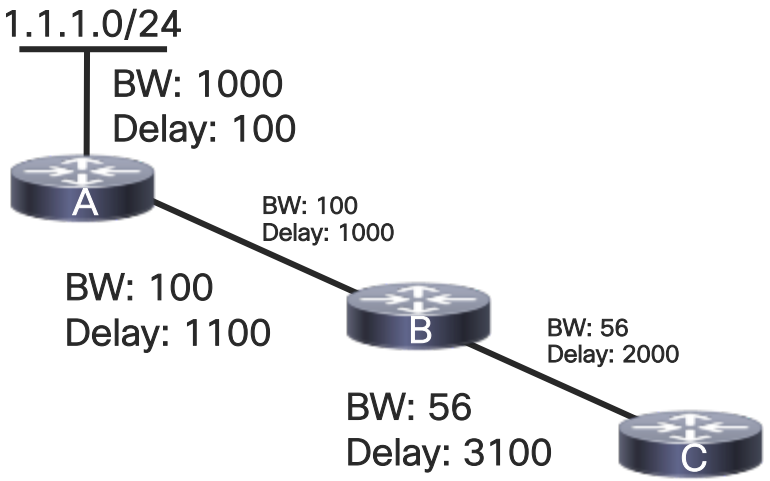



Metrics:
How far is too far?
cisco Live!

EIGRP Metrics

Computing Metrics

- What's important?
  - Consider the Path ->
    - END to END, HOP by HOP
  - Minimum Bandwidth along the Path
  - Add up the Total Delay
- Combine into the magic formula!

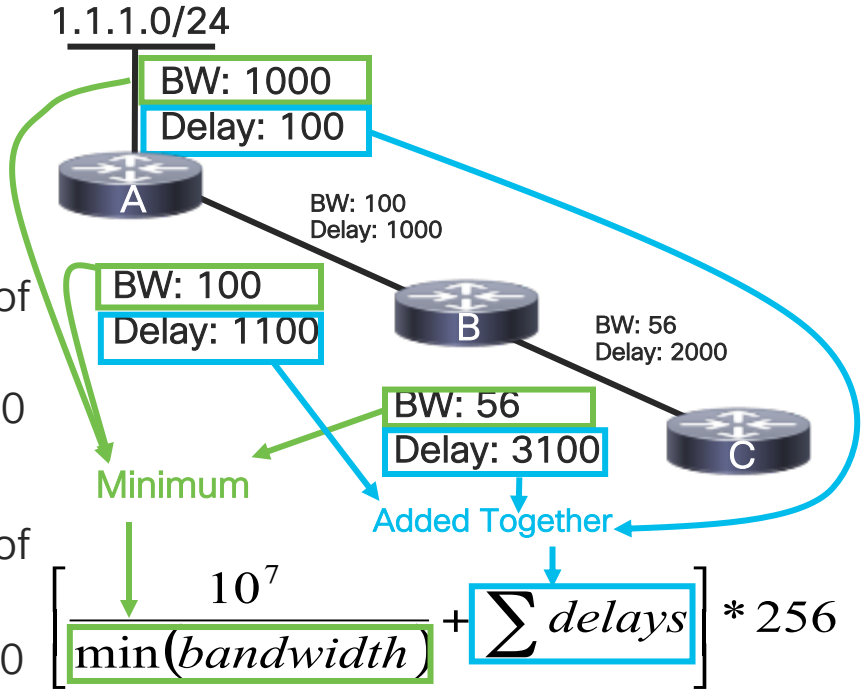


$$\left[\frac{10^7}{\min(bandwidth)} + \sum delays\right] * 256$$

EIGRP Metrics

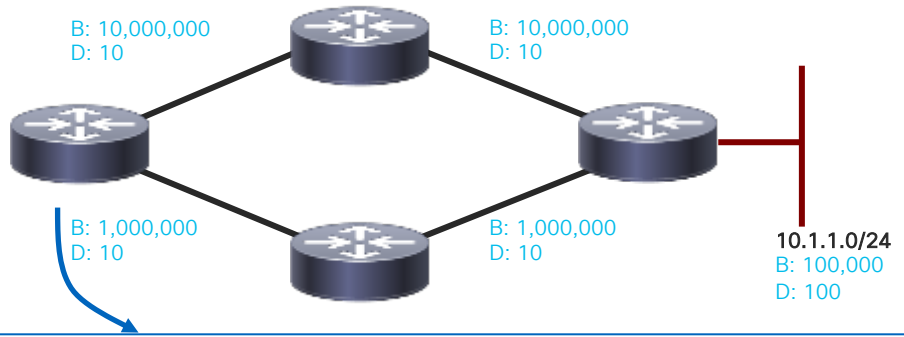
Computing Metrics

- Router A advertises 1.1.1.0/24 to B
  - Bandwidth is set to 1000
  - Delay is set to 100
- Router B
  - Compares current bandwidth to bandwidth of link to A; sets bandwidth to 100
  - Adds delay along link to A, for a total of 1100
- Router C
  - Compares current bandwidth to bandwidth of link to B; sets bandwidth to 56
  - Adds delay along link to B, for a total of 3100



Classic Metric Formula

- With the simplified EIGRP Formula:
- $$metric = \left[\frac{10^7}{\min(bandwidth)} + \sum delays \right] * 256$$
- The path has a minimum bandwidth of 100,000 kbps
- The path though the Ten Gigabit Bundle has a total delay of 120 microseconds
- But so does the path through the Gigabit Ethernet!




```
10.4.4.2 (TenGigabitEthernet2/0), from 10.4.4.2, Send flag is 0x0
      Composite metric is (28672/28416), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 120 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
  10.5.5.3 (GigabitEthernet3/0), from 10.5.5.3, Send flag is 0x0
      Composite metric is (28672/28416), Route is Internal
      Vector metric:
        Minimum bandwidth is 100000 Kbit
        Total delay is 120 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

Classic Metric Formula

- EIGRP's calculated metric is called the composite metric
- Its computed from individual metrics called vector metrics
  - minimum bandwidth, total delay, load, reliability
- Interface metrics are converted before use
  - bandwidth (in kilobits per second): 107 / Interface bandwidth
  - delay (in 10s of microseconds): interface delay / 10ms
  - load, reliability: converted to range of 0–255

$$\text{metric} = \left[\left(K_1 \bullet \text{bandwidth} + \frac{K_2 \bullet \text{bandwidth}}{256 - \text{Load}} + \left(K_3 \bullet \text{Delay} \right) \right) \bullet \frac{K_5}{K_4 + \text{Reliability}} \right] \bullet 256$$

- Constants (K1 through K5) are used to control the computation
  - Default K values are: K1 == K3 == 1 and K2 == K4 == K5 == 0
  - When K5 is equal to 0 then [K5/(K4 + reliability)] is defined to be 1

Computing Classic Metrics

- Router C uses the formula to compute a composite metric
  - This isn't what the router computes, though—why?
    - The router drops the remainder after the first step!

$$\left[\frac{10^7}{\min(bandwidth)} + \sum delays\right] * 256$$

$$\left[\frac{10^7}{56} + 3100\right] * 256 = 46507885$$

- Why the 256 multiplier?
  - EIGRP uses a 32-bit metric space
  - IGRP used a 24-bit metric space
  - To convert between the two, multiply or divide by 256!

$$\left(\frac{10^7}{56} = 178571\right)$$

$$(178571 + 3100) * 256 = 46507776$$

?

Wide Metric Support – New Formula

- EIGRP still uses vector metrics, but they are not scaled, and are processed differently

$$\left[\left(K_1 \bullet \text{Throughput} + \left\{\frac{K_2 \bullet \text{Throughput}}{256 - \text{Load}}\right\}\right) + (K_3 \bullet \text{Latency}) + (K_6 \bullet \text{Ext Metrics})\right] \bullet \frac{K_5}{K_4 + \text{Reliability}}$$

- New vector metrics are derived from values reported by router
  - Latency – derived from interface delay
  - Throughput – derived from interface bandwidth
  - Load – derived from interface load
  - Reliability – derived from interface reliability
  - Ext Metrics – derived from router and/or configuration

$$latency = \left[delay * 10^6\right] OR \left[\frac{10^{13}}{bandwidth}\right]$$

$$throughput = \left[\frac{6.5536 * 10^{11}}{bandwidth}\right]$$

$$metric = \left[\min(throughput) + \sum latency\right]$$

- Constants (K1 through K6) are used to control the computation
  - Default K values are: K1 == K3 == 1 and K2 == K4 == K5 == K6 == 0

Wide Metric Support – Computing Metrics

- By default, EIGRP computes throughput using the maximum theoretical throughput
- The formula for the conversion for max-throughput value directly from the interface without consideration of congestion-based effects is as follows:

$$\text{Max-Throughput} = \left(K_1 \bullet \frac{\text{EIGRP\_BANDWIDTH} \bullet \text{EIGRP\_WIDE\_SCALE}}{\text{Bandwidth}}\right)$$

- If K2 is used, the effect of congestion, as a measure of load reported by the interface, will be used to simulate the available throughput, by adjusting the maximum throughput according to the formula:

$$\text{Net-Throughput} = \left[\text{Max-Throughput} + \left(\frac{K_2 \bullet \text{Max-Throughput}}{256 - \text{Load}}\right)\right]$$

- This inversion of bandwidth value results in a larger number (more time), ultimately generating a worse metric.
- The inverted value is used only by the local router, the original bandwidth value is send to its neighbors

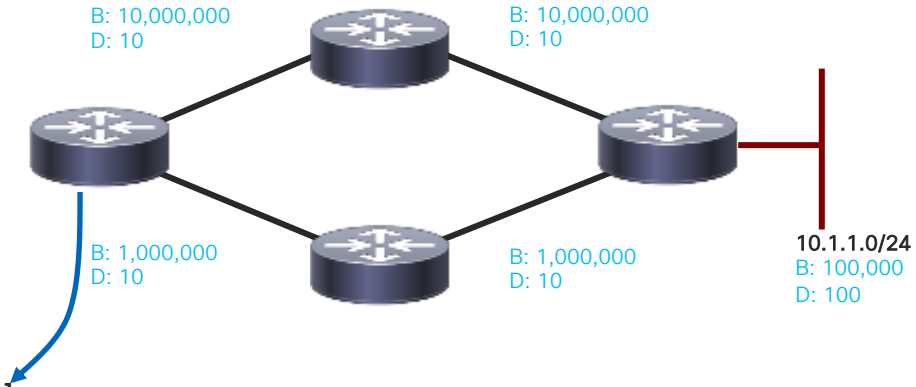
Wide Metric Support – Computing Metrics

- K3 is used to allow latency-based path selection. Latency and delay are similar terms that refer to the amount of time it takes a bit to be transmitted to an adjacent peer. EIGRP uses one-way based latency values provided either by IOS interfaces or computed as a factor of the links bandwidth
- $$\text{Latency} = \left(K_3 \bullet \frac{\text{Delay} \bullet \textbf{EIGRP\_WIDE\_SCALE}}{\textbf{EIGRP\_DELAY\_PICO}} \right)$$
- For IOS interfaces that do not exceed 1 gigabit, this value will be derived from the reported interface delay, converted to picoseconds
- $$\text{Delay} = \left(\text{Interface Delay} \bullet \textbf{EIGRP\_DELAY\_PICO} \right)$$
- For IOS interfaces beyond 1 gigabit, IOS does not report delays properly, therefore a computed delay value will be used
- $$\text{Delay} = \left(\frac{\textbf{EIGRP\_BANDWIDTH} \bullet \textbf{EIGRP\_DELAY\_PICO}}{\text{Interface Bandwidth}} \right)$$

Wide Metric Support – New Formula

* Due to rib metric scaling, use of route-maps "set metric" can result in compatibility issues with older peers

- Wide Metrics enables us to:
  - Configure delay values in pico-seconds
  - Pass unscaled delay/bandwidth values between peers
  - Maintain backwards compatibility*
- Only available in named mode

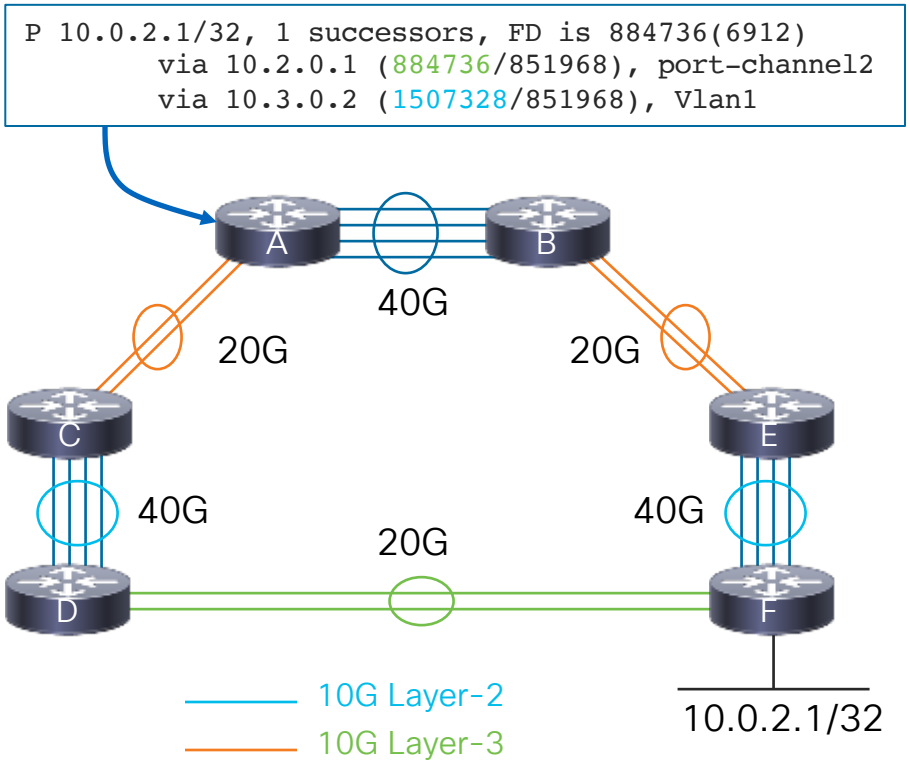



```
Router# show eigrp address-family ipv4 topology
EIGRP-IPv4 VR(WideMetric) Topology Entry for AS(4453)/ID(3.3.3.3) for 10.1.1.0/16
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 262144, RIB is 2048
  Descriptor Blocks:
  10.4.4.2 (TenGigabitEthernet2/0), from 10.4.4.2, Send flag is 0x0
      Composite metric is (262144/196608), route is Internal
      Vector metric:
        Minimum bandwidth is 10000000 Kbit
        Total delay is 3000000 picoseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
        Originating router is 100.1.1.1
```

RIB Metric still in 32bit form

Wide Metric Support – Considerations

- Consider the following deployment
  - Each router is an N7K with 10Gig physical links and default configs
  - port-channel2 is a bundle of 2 links
  - switched-virtual-interface (Vlan1) is a bundle of 4 links
- What path would you expect data to from router A to follow to get to the loop back on router F?
- Lets look at the topology table entry for 10.0.2.1/32 on router A
- Was it expected!? Why not?



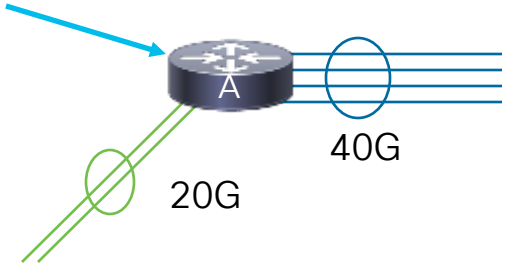
Wide Metric Support – Considerations

- Lets look at the interface values
- The port-channel looks good…
- But the SVI interface is reporting 1Gig on the N7K!
- Use the eigrp delay command to set the proper interface delay

```
RrtA# show ip interface port-channel2
port-channel2 is up
admin state is up
  Hardware: Port-Channel, address: 0026.51bc.d447
Internet Address is 10.13.0.1/30
  MTU 1500 bytes, BW 20000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
```

```
RrtA# show ip interface vlan1
Vlan1 is up, line protocol is up, autostate enabled
  Hardware is EtherSVI, address is  0026.51bc.d447
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
   reliability 255/255, txload 1/255, rxload 1/255
```

```
interface Vlan1
ip delay eigrp AS1 250000 picoseconds
  bandwidth 40000000
```



* Only need to set the bandwidth in the event min-bandwidth is of concern

Summary

Summary: What Have We Learned So Far?

- Distance Vector routing basics: Hop-by-hop, information hiding
- EIGRP supports traditional router based configuration and Address-Family based
- EIGRP's Neighbors – utilize a 3-way handshake, very scalable
- EIGRP packets – 5 types, reliable/unreliable, reliable transport for delivery
- Metrics
  - Minimum Bandwidth, sum of the Delay, hop by hop along the path
  - Wide Metrics allows EIGRP to detect links speeds up to 4.2 Terabytes
- What's next?
  - Convergence for EIGRP, Stub, and Summarization
  - Check out CiscoLive online for past EIGRP sessions for more details
    - BRKRST-2331 for EIGRP Troubleshooting
    - BRKRST-2336 for EIGRP Design

Agenda

- Section 1
  - Intro to Distance Vector
  - Basic EIGRP Configuration
- Section 2
  - Neighbors
  - Packets
  - Metrics
- Section 3
  - Topology Table
  - Convergence
  - Stub
  - Summarization
- Section 4
  - Basic Design
  - Event Log
  - EIGRP Redistribution

Agenda

- The Topology Table
- How does EIGRP converge?
  - Understanding DUAL
  - Feasible Successor
  - Non-Feasible Successor
- Stub
- Summarization
- Summary & Conclusion

Convergence:
Which way do we go?

Understanding Convergence

- The one constant in life is that nothing is constant. Not even our networks. ☺
- There are three network change scenarios to consider:
  - Convergence with Feasible Successors
  - Convergence with Non-Feasible Successors
  - When things don't converge as you had planned!
- Understanding your topology and the appropriate scenario is essential to know where to look and what to look for.
- Get up close and personal with your topology table and event-logs!

Topology Table & DUAL

Topology Table

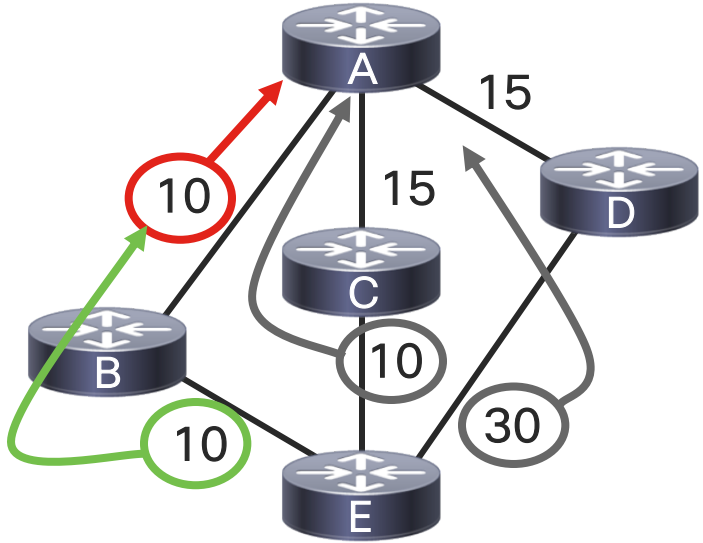
- The topology table is probably the most critical structure in EIGRP
  - Contains all known paths, local, learned, and external (redistributed)
  - Contains building blocks used by DUAL
  - Used to create updates for neighbors
  - Used to populate the routing table
- Understanding the topology table contents is extremely important for understanding your network and troubleshooting EIGRP

Topology Table

- Remember our order of operation!
  - Form Neighbors
  - Exchange Updates
  - Use Metrics to determine which path is best, through which the destination is “closest”.
  - Best path(s) are installed into the routing table (rib)
  - Neighbors are updated
- Topology Table is like the local database of all known paths, from which we determine what is best.

Topology Table & DUAL – Key Terms

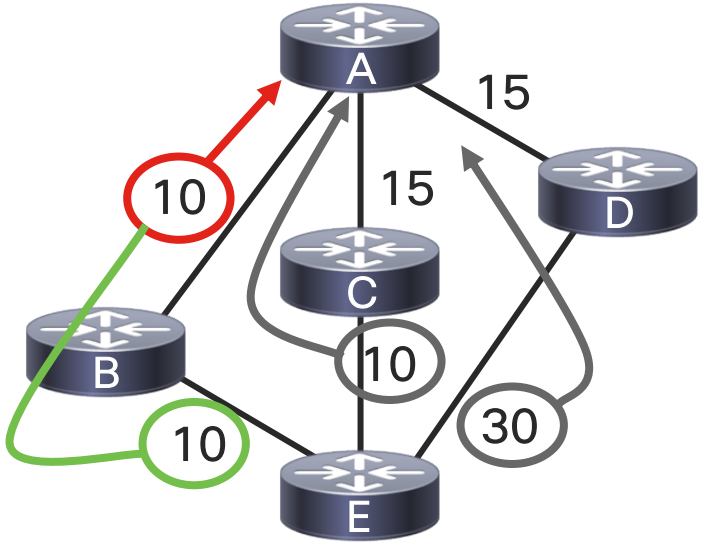
- From Router A's perspective, looking at updates coming inbound:
- Reported Distance (RD)
  - Reported Distance is the total distance along a path to a destination network as advertised by an upstream peer
  - Example: 10, 10, and 30
- Computed Distance (CD)
  - The Reported Distance plus link cost to reach the upstream peer
  - Example: 10 + 10 = 20, 10+15, 30+15
- Feasible Distance (FD)
  - Feasible Distance is the ***lowest*** Computed Distance to a particular destination
  - Example: 20 is less than 25, 20 is less than 45, thus 20 is the FD for this particular destination



"Particular Destination"

Topology Table & DUAL – Key Terms

- **Feasibility Condition**
  - If a router's Reported Distance is less than our Feasible Distance, then this router is a loop-free path to this destination and meets the Feasibility Condition (RD < FD) (RD of 10 is less than FD of 20)
- **Feasible Successor**
  - A router is a Feasible Successor if it satisfies the Feasibility Condition for a particular destination
  - C-RD:10 < 20, thus **C is a FS**. D-RD:30 is not < 20, **D is NOT a FS**.
- **Successor**
  - A router is a Successor if it satisfies the Feasibility Condition AND it provides the lowest distance (metric) to that destination
- **Active and Passive State**
  - A route is in the Passive state when it has a successor for the destination. The router goes to Active state when current successor no longer satisfies the Feasibility Condition and there are no Feasible Successors identified for that destination



"Particular Destination"

Topology Table – Where?

```
R# show eigrp address-family ipv4 topology

EIGRP-IPv4 Topology Table for AS(1)/ID(1.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply, r - reply Status, s - sia Status

P 2.2.2.0/24, 1 successors, FD is 21026560
       via 60.1.1.2 (21026560/20514560), FastEthernet1/0
       via 60.1.2.1 (46740736/20514560), FastEthernet1/1
```

Feasible Distance

Successor

State

Computed Distance

Reported Distance

Feasible Successor

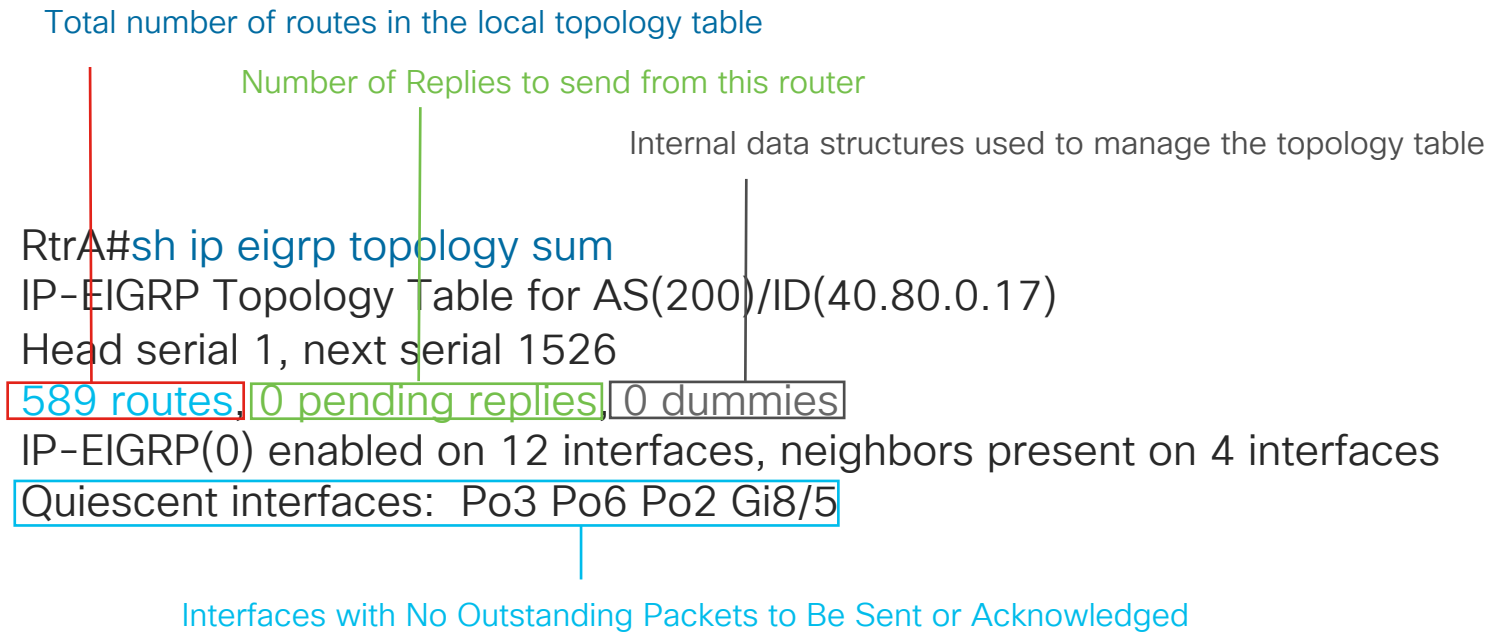
RD(20514560) < FD(21026560)

Topology Table

- One of the reasons that EIGRP is called an advanced distance vector protocol is that it retains more information than just the best path for each route it receives—this means that it can potentially make decisions more quickly when changes occur, because it has a more complete view of the network than RIP, for example; the place this additional information is stored is in the topology table
- The topology table contains an entry for every route EIGRP is aware of, and includes information about the paths through all neighbors that have reported this route to him—when a route is withdrawn by a neighbor, EIGRP will look in the topology table to see if there is a feasible successor, which is another downstream neighbor that is guaranteed to be loop-free; if so, EIGRP will use that neighbor and never have to go looking farther
- Contrary to popular belief, the topology table also contains routes which are not feasible; these are called possible successors and may be promoted to feasible successors, or even successors if the topology of the network were to change
- The following slides show a few different ways to look at the topology table and give hints on how to evaluate it

Topology Table

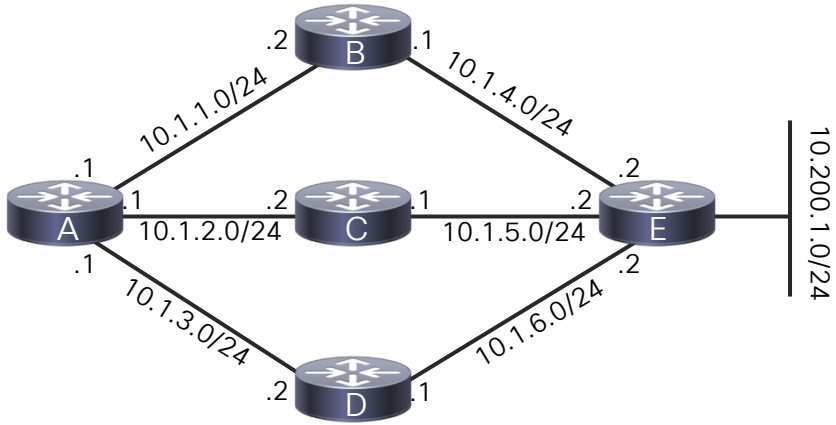
Show IP EIGRP Topology Summary



Topology Table

Show IP EIGRP Topology

- Displays a list of successors and feasible successors for all destinations known by EIGRP




```
RtrA#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)
..snip…..
P 10.200.1.0/24, 1 successors, FD is 21026560
         via 10.1.1.2 (21026560/20514560), Serial1/0
         via 10.1.2.2 (46740736/20514560), Serial1/1
```

Feasible distance

Successor

Feasible successor

Computed distance

Reported distance

Topology Table

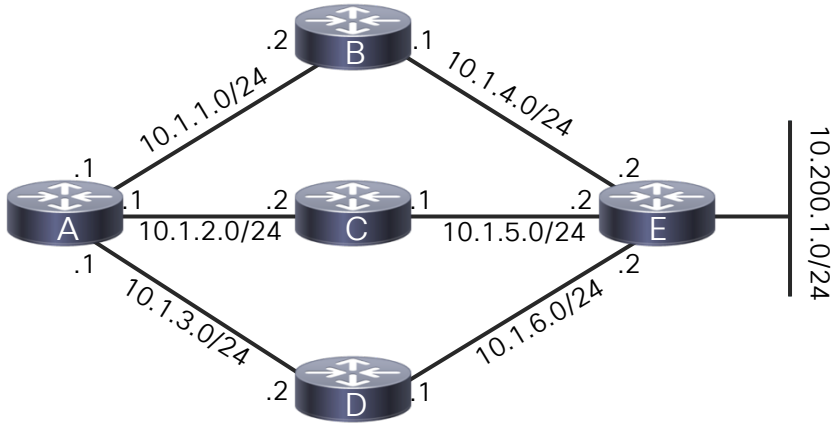
Show IP EIGRP Topology

- The most common way to look at the topology table is with the generic show ip eigrp topology command; this command displays all of the routes in the EIGRP topology table, along with their successors and feasible successors
- In the above example, the P on the left side of the topology entry displayed means the route is Passive—if it has an A, it means the route is Active; the destination being described by this topology entry is for 10.200.1.0 255.255.255.0—this route has one successor, and the feasible distance is 21026560; the feasible distance is normally the metric that would appear in the routing table if you did the command show ip route 10.200.1.0 255.255.255.0 (but not always)
- Following the information on the destination network, the successors and feasible successors are listed—the successors (one or more) are listed first, then the feasible successors are listed; the entry for each next-hop includes the IP address, the computed distance through this neighbor, the reported distance this neighbor told us, and which interface is used to reach him
- 10.1.2.2 is a feasible successor because his reported distance (21514560) is less than our current feasible distance (21026560) (It's a smaller distance, and that means it is closer.)

Topology Table

Show IP EIGRP Topology All-links

- Displays a list of all neighbors who are providing EIGRP with an alternative path to each destination




```
RtrA#show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)
…..snip…..
P 10.200.1.0/24, 1 successors, FD is 21026560
         via 10.1.1.2 (21026560/20514560), Serial1/0
         via 10.1.2.2 (46740736/20514560), Serial1/1
         via 10.1.3.2 (46740736/46228736), Serial1/2
```

Feasible distance

Successor

Feasible successor

Possible successor

Computed distance

Reported distance

Topology Table

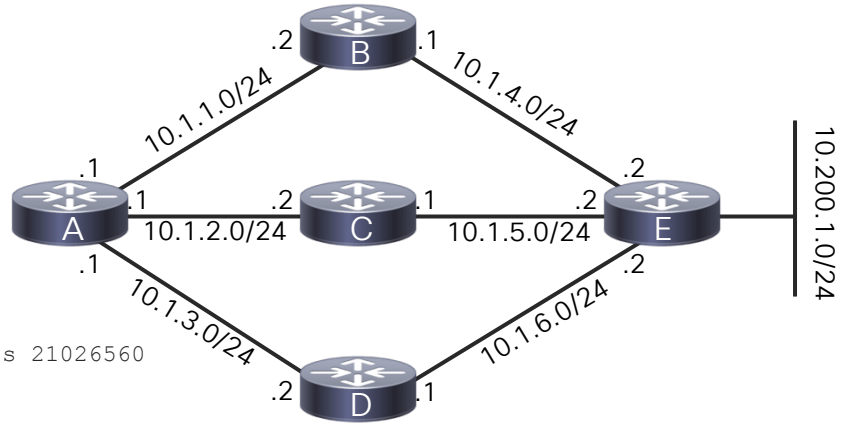
Show IP EIGRP Topology All-links

- If you want to display all of the paths which EIGRP contains in its topology table, use the show ip eigrp topology all-links command
- You'll notice in the above output that not only are the successor (10.1.1.2) and feasible successor (10.1.2.2) shown, but another router that doesn't qualify as either is also displayed; the reported distance from 10.1.3.2 (46228736) is far worse than the current feasible distance (21026560), so it isn't feasible
- This command is often useful to understand the true complexity of network convergence—I've been on networks with pages of non-feasible alternative paths in the topology table because of a lack of summarization/distribution lists; these large numbers of alternative paths can cause EIGRP to work extremely hard when transitions occur and can actually keep EIGRP from successfully converging

Topology Table

Show IP EIGRP Topology <net><mask>

- Displays detailed information for all paths received for a particular destination




```
RtrA#show ip eigrp topology 10.200.1.0/24
IP-EIGRP topology entry for 10.200.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21026560
  Routing Descriptor Blocks:
  10.1.1.2 (Serial1/0), from 10.1.1.2, Send flag is 0x0
      Composite metric is (21026560/20514560), Route is Internal
      Vector metric:
        ....
  10.1.2.2 (Serial1/1), from 10.1.2.2, Send flag is 0x0
      Composite metric is (46740736/20514560), Route is Internal
      Vector metric:
        ....
  10.1.3.2 (Serial1/2), from 10.1.3.2, Send flag is 0x0
      Composite metric is (46740736/46228736), Route is Internal
      Vector metric:
        ....
```

Topology Table

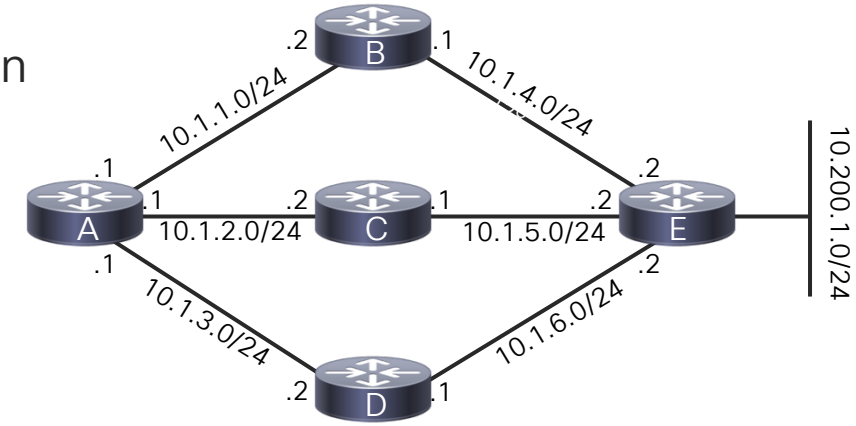
Show IP EIGRP Topology <network><mask>

- If you really want to know all of the information EIGRP stores about a particular route, use the command show ip eigrp topology <network><mask>
  - Note that the mask can be supplied in dotted decimal or /xx form
- In the above display, you'll see that EIGRP not only stores which next-hops have reported a path to the target network, it stores the metric components used to reach the total (composite) metric
- You also may notice that EIGRP contains a hop count in the vector metrics—the hop count isn't actually used in calculating the metric, but instead was included to limit the apparent maximum diameter of the network; in EIGRP's early days, developers wanted to ensure that routes wouldn't loop forever and put this safety net in place—in today's EIGRP, it actually isn't necessary any longer, but is retained for compatibility

Topology Table

External Topology Table Entry

- Showing the topology table entry for an external route shows additional information about the route




```
RtrA#show ip eigrp topology 30.1.1.0/24
IP-EIGRP topology entry for 30.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 46738176
  Routing Descriptor Blocks:
  10.1.3.2 (Serial1/2), from 10.1.3.2, Send flag is 0x0
  ....
      External data:
        Originating router is 64.1.4.14
        AS number of route is 0
        External protocol is Static, external metric is 0
        Administrator tag is 0 (0x00000000)
```

Topology Table

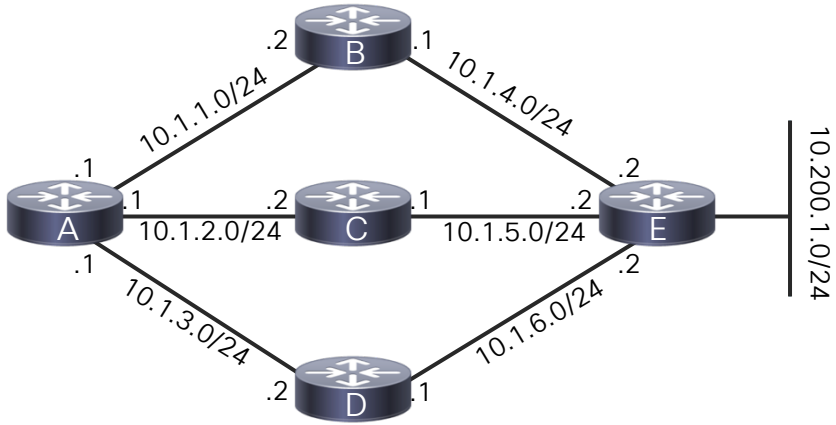
External Topology Table Entry

- If you perform the command show ip eigrp topology <network><mask> for an external route (one redistributed into EIGRP from another protocol), even more information is displayed
- The initial part of the display is identical to the command output for an internal (native) route—the one exception is the identifier of the route as being external; another section is appended to the first part, however, containing external information—the most interesting parts of the external data are the originating router and the source of the route
- The originating router is the router who initially redistributed the route into EIGRP—note that the value for the originating router is router-id of the source router, which doesn't necessarily need to belong to an EIGRP-enabled interface; the router-id is selected in the same way OSPF selects router-ids, starting with highest IP address on a loopback interface, if any are defined, or using the highest IP address on the router if there aren't loopback interfaces—note that if a router receives an external route and the originating router field is the same as the receiver's router-id, it rejects the route—this is noted in the event log as ignored, dup router
- The originating routing protocol (where it was redistributed from) is also identified in the external data section; this is often useful when unexpected routes are received and you are hunting the source

Topology Table

Show IP EIGRP Topology Zero

- Zero successor routes are those that fail to get installed in the routing table by EIGRP because there is a route with a better admin distance already installed




```
RtrA#show ip eigrp topology zero
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)
....
P 10.200.1.0/24, 0 successors, FD is Inaccessible
         via 10.1.1.2 (21026560/20514560), Serial1/0
         via 10.1.2.2 (46740736/20514560), Serial1/1
         via 10.1.3.2 (46740736/46228736), Serial1/2
```

```
RtrA#show ip route 10.200.1.0 255.255.255.0
Routing entry for 10.200.1.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.1.1.2
      Route metric is 0, traffic share count is 1
```

Topology Table

Show IP EIGRP Topology Zero

- And last, the show ip eigrp topology zero command is available to display the topology table entries that are not actually being used by the routing table
- Typically, zero successor entries are ones that EIGRP attempted to install into the routing table, but found a better alternative there already; in our example above, when EIGRP tried to install its route (with an administrative distance of 90), it found a static route already there (with an administrative distance of one) and thus couldn't install it—in case the better route goes away, EIGRP retains the information in the topology table, and will try to install the route again if it is notified that the static (or whatever) route is removed
- Routes that are active sometimes also show up as zero successor routes, but they are transient and don't remain in that state
- This command isn't often used or useful

DUAL

DUAL

- Diffusing Update Algorithm
  - A “Diffusing Computation” is one where a device engages their neighbors with subtasks to collectively resolve the answer. *(Dijkstra, Scholten)*
- Evaluates all the Updates and the entries stored in the Topo Table
- Handles changes in the Topology
  - Provides loop free convergence for the EIGRP network
  - Provides a process for Active (fluctuating/unconverged) routes to either be come Passive (stable/converged) or be removed
- *But how does it work?*

DUAL – How it works

- Basic idea – if your metric to the destination is less than mine, the path through you can not be a loop
- Consider the following network

  For simplicity, all links cost (1)

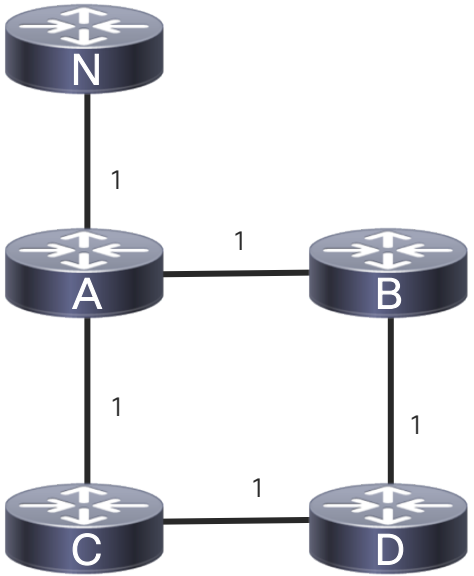
  Each router has the following Feasible Distance to reach N:

  A: Successor = N, FD=1

  B: Successor = A, FD=2

  C: Successor = A, FD=2

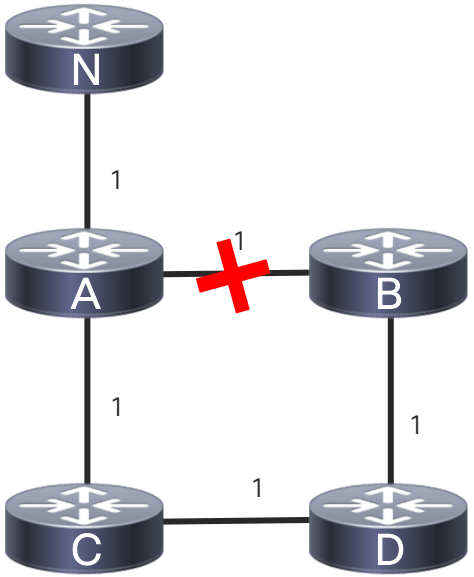
  D: Successor = B, FS = C, FD=3



Note: From Router-D point of view, both Router-B and Router-C are equal cost. But Router-D must pick one as a successor, the other will be the FS. Which one Router-D picks will be dependent on the order Router-D received the information

DUAL – How it works

- If the link between A and B fails;
- B sends a query informing its peers that it has lost its successor
- D receives the query and determines if it has any other feasible successors. If it does not, it has to start a route computation and enter the active state
- However in this case, C is a feasible successor because its cost (2) is less than than D's current cost (3) to destination N
- D will switch to C as its new successor

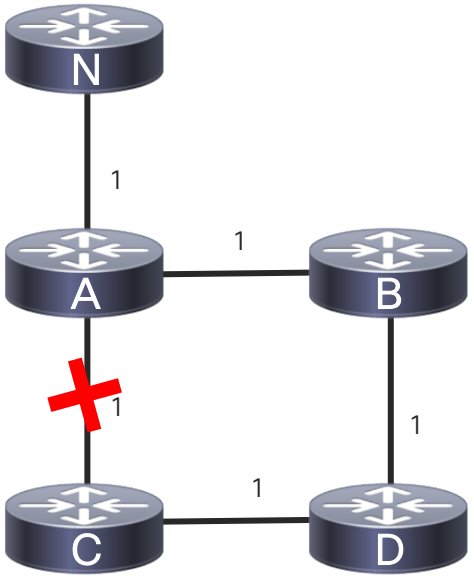


Note A and C did not participate because they were unaffected by the change.

DUAL – How it works

- Now let's cause a route computation to occur
- If the link between A and C fails;
- C determines that it has lost its successor and has no other feasible successors

  D is not considered a feasible successor because its advertised metric (3) is greater than C's current cost (2) to reach destination N
- C sends a query to its only peer D
- D replies because its successor has not changed
- When C receives the reply it can choose its new feasible successor D with a cost of (4) to reach destination N



Note A and B were unaffected by the topology change and D needed to simply reply to C.

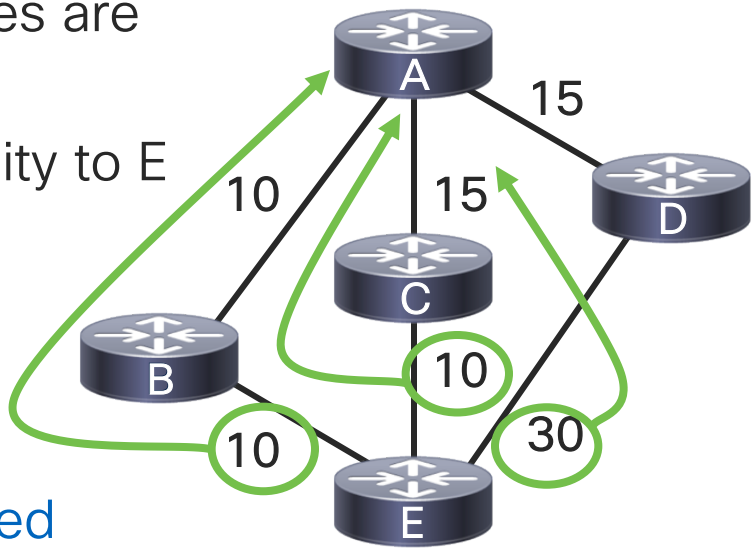
DUAL – Choosing a Successor/FS

- How does EIGRP determine which routes are loop free?
- Each of A's peers is reporting reachability to E

  B with a cost of 10

  C with a cost of 10

  D with a cost of 30



- These three costs are called the reported distance (RD); the distance each peer is reporting to a given destination

DUAL – Choosing a Successor/FS

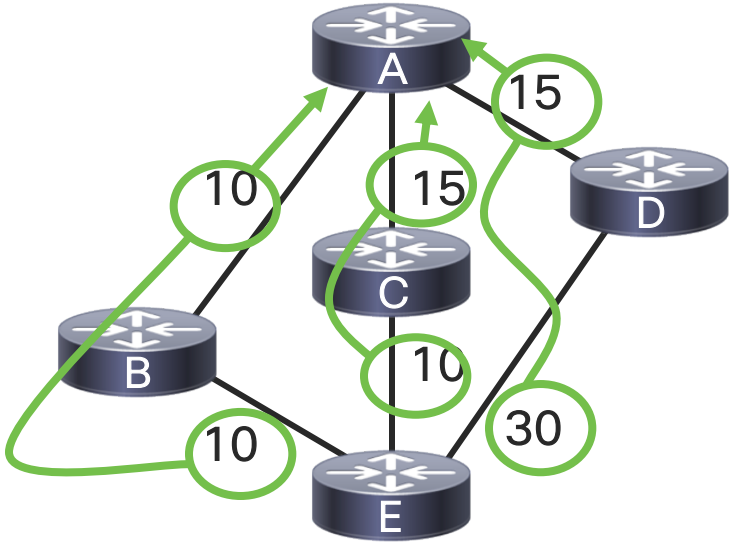
- At A, the total cost to reach E is:

  20 through B

  25 through C

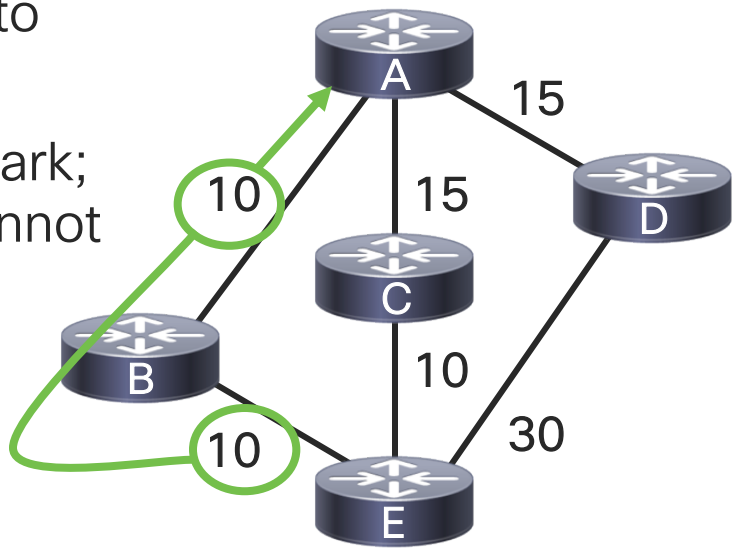
  45 through D

- The best of these three paths is the path through B, with a cost of 20
- This is the feasible distance (FD)



DUAL – Choosing a Successor/FS

- A uses these two pieces of information to determine which paths are loop free
- The best path (FD) is used as a benchmark; all paths with RD's lower than the FD cannot contain loop

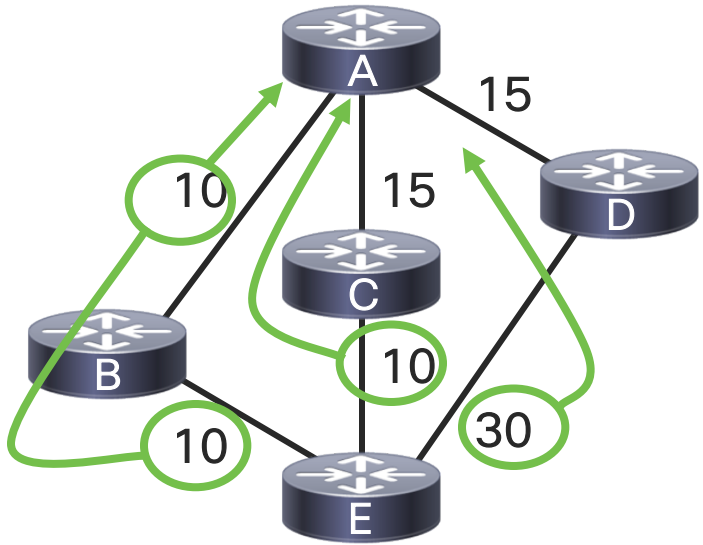


- The algorithm may mark some loop free paths as possible loops
- It is guaranteed never to mark a looped path as loop free

DUAL – Choosing a Successor/FS

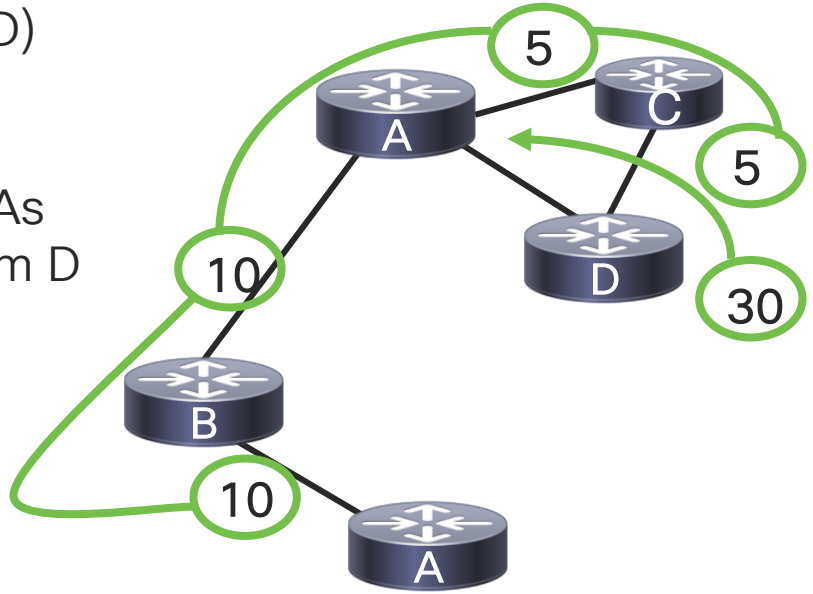
For example, consider how A views the path available path:

- The path through B is the best path (FD), at 20
- C can reach E with a cost of 10; 10 (RD) is less than 20 (FD), so this path is loop free
- D can reach E with a cost of 30; 30 (RD) is not less than 20 (FD), so EIGRP assumes this path could be a loop



DUAL

- Question:
  Why should DUAL consider the 30 (RD) from D as a possible loop?
- Answer:
  Because, mathematically it could be. As far as A is concerned, the 30 (RD) from D could be the loop we see here



Understanding Convergence

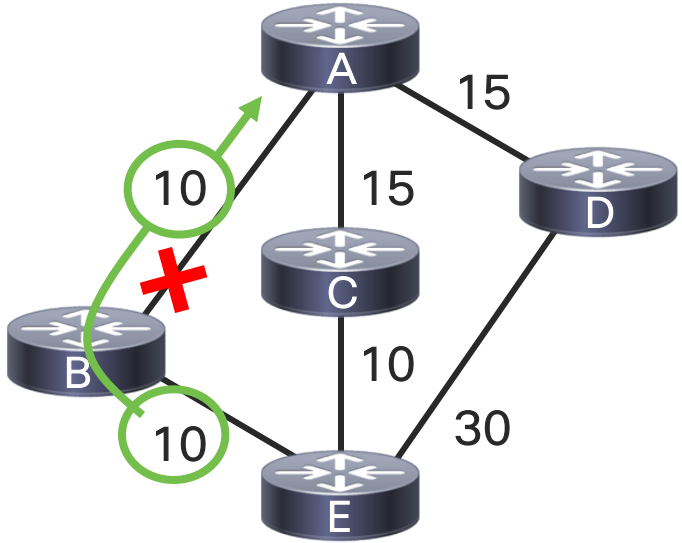
EIGRP Convergence – With Feasible Successor

- EIGRP selects Successor and Feasible Successor (FS)
- Successor is the best route
- FS is 2nd best route
- Must be mathematically loop-free (meets feasibility condition)
- FS acts as a “backup route”
- Kept in topology table (not routing table)
- Up to 6 Feasible Successors
- Built into the protocol, nothing to enable

Failover to Feasible Successor

- What happens if the best path fails, through B (Successor)?
- EIGRP will examine the available paths to E
- Finding a path previously declared loop free (Feasible Successor) , it begins using it immediately

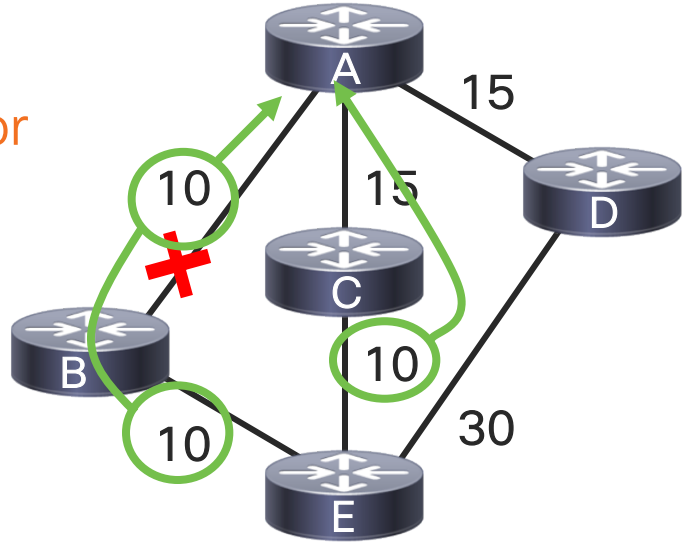
✓C now becomes the Successor (best path)



Failover

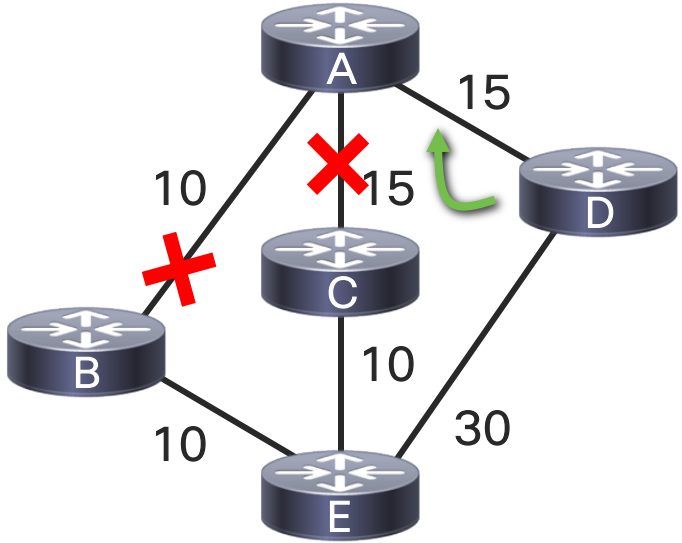
- A now examines its topology information based on the new successor metric to determine if there is an Feasible Successor (FS) available
- The Reported Distance (RD) through the remaining peer is D, is 30;
  - 30 (RD) is more than 25 (FD)

✗ So this path is still considered a possible loop



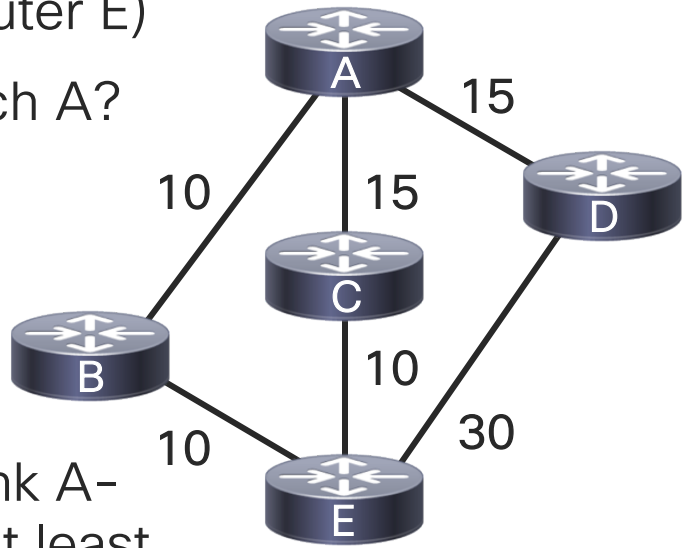
DUAL – Failover to a Non-Feasible Successor

- D receives a query from Router-A and examines its topology information
- Since its best path is not through A, the path it has to E is still valid
- D sends a reply to this query, indicating it still has a valid loop free path to E
- ✓Once A receives this reply, it begins using the path through D



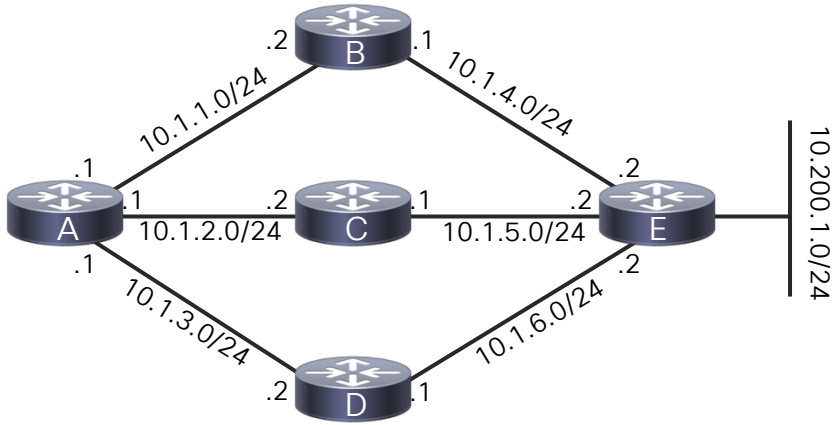
DUAL – Failover to Feasible Successor

- What about the *other* direction? (From Router E)
- Are there any Feasible Successors to reach A?
  - FD is 20 through B
  - RD from C is 15
  - RD from D is 15
  - RD < FD, so both satisfy the Feasibility Condition (FC)
  - We have two FS!
- In order for there to be only one FS, the link A-D or A-C would need to be increased to at least 20



Convergence of a Feasible Successor

- Near immediate rewrite of the next hop in the rib/fib.
- Extremely fast, and linear convergence based on prefix count.




```
RtrA#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)
..snip…..
P 10.200.1.0/24, 1 successors, FD is 21026560
         via 10.1.1.2 (21026560/20514560), Serial1/0
         via 10.1.2.2 (46740736/20514560), Serial1/1
```

Feasible distance

Successor

Feasible successor

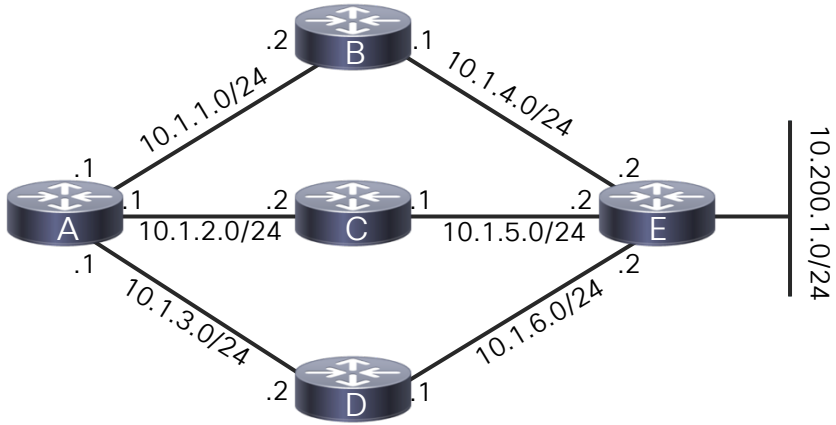
Computed distance

Reported distance

RD (20514560) < FD (21026560) = FS!

Convergence of a Feasible Successor

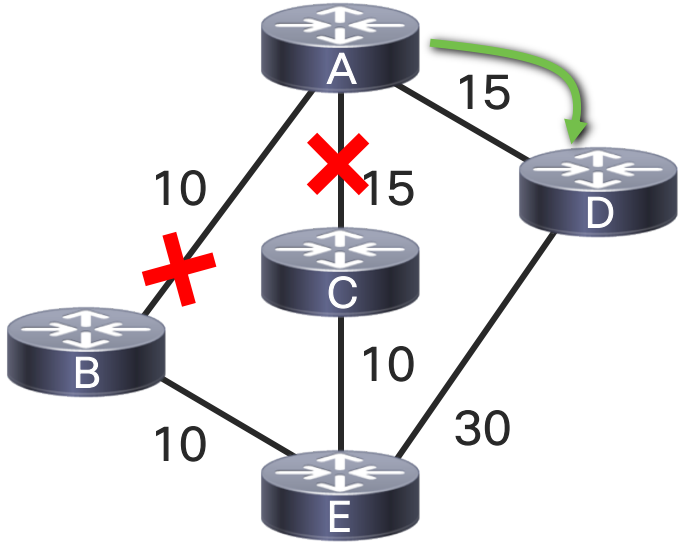
- Indicated in the Event Log, local computation
- Extremely fast, and linear convergence based on prefix count.




```
RtrA#show ip eigrp event
…
97    11:12:06.124 Metric set: 10.1.4.0/24 metric(20480)
98    11:12:06.124 Route installing: 10.1.4.0/24 10.1.2.2
99    11:12:06.124 Route installed: 10.1.4.0/24 10.1.1.2
100   11:12:06.124 Route installing: 10.1.4.0/24 10.1.1.2
101   11:12:06.124 RDB delete: 10.1.4.0/24 10.1.3.2
102   11:12:06.124 FC sat rdbmet/succmet: metric(20480) metric(20224)
103   11:12:06.124 FC sat nh/ndbmet: 10.1.1.2 metric(20480)
104   11:12:06.124 Find FS: 10.1.4.0/24 metric(20480)
105   11:12:06.124 Rcv update met/succmet: metric(Infinity) metric(Infinity)
106   11:12:06.124 Rcv update dest/nh: 10.1.4.0/24 10.1.3.2
107   11:12:06.123 Send reply: 10.1.4.0/24 10.1.2.2
108   11:12:06.123 Rcv query met/succ met: metric(Infinity) metric(Infinity)
109   11:12:06.123 Rcv query dest/nh: 10.1.4.0/24 10.1.2.2
110   11:12:06.123 Ignored route, hopcount: 10.1.4.0 255
```

Failover to a Non-Feasible Successor

- What if the path through C now fails?
- A examines its topology information, and finds it has no loop free path to E
- However, it does have a peer, and that peer might have a loop free path
- So, it places E in active state and Queries D

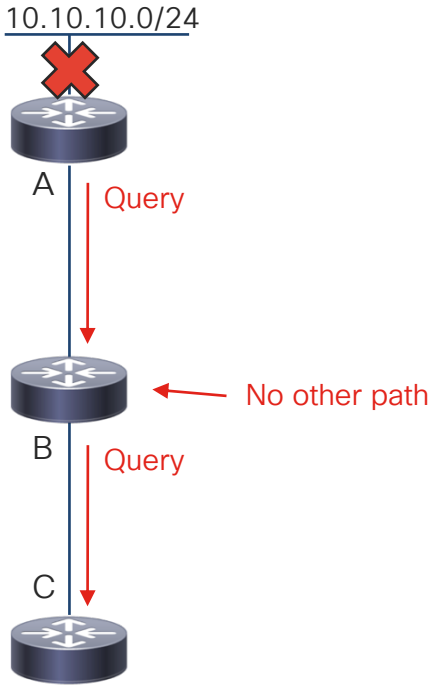


EIGRP Convergence – Without Feasible Successor

- Distance Vector Protocol
  - Doesn't see the entire network like OSPF
- Based on QUERY and ACK messages for convergence
  - QUERY sent to determine best path for failed route
  - ACK sent when alternative path found or no other paths
- DUAL algorithm determines best path
  - Runs as soon as all outstanding QUERIES are received
- Query domain size can effect convergence time

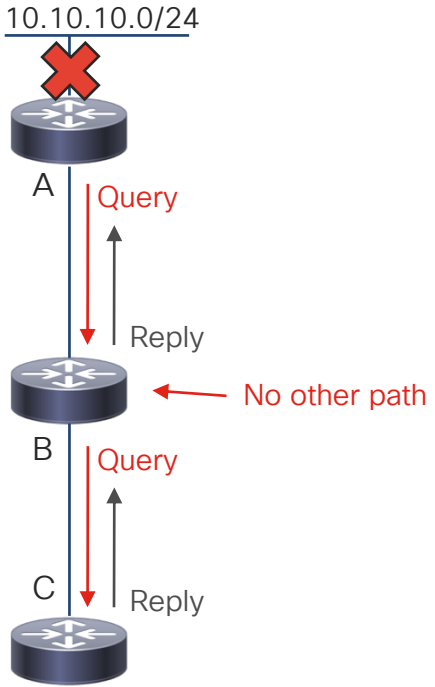
The Active Process

- RtrA loses its route to 10.10.10.0/24
- RtrA has no other path to this destination, so it marks the route as Active and sends a Query to RtrB
- RtrB receives this Query from its successor and has no other paths to reach the destination
- RtrB marks 10.10.10.0/24 as Active, and sends a Query to RtrC



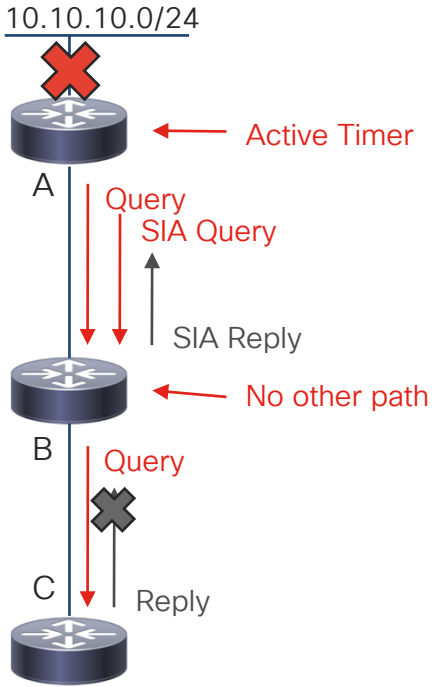
The Active Process

- RtrC receives the Query and has no more neighbors to Query and no alternate paths to 10.10.10.0/24
- RtrC marks the route as unreachable, and sends a Reply to RtrB
- RtrB receives the Reply, marks 10.10.10.0/24 as unreachable, and sends a Reply to RtrA
- RtrA receives the Reply and since it didn't learn any viable paths to reach 10.10.10.0/24, it deletes the route from the topology and routing tables



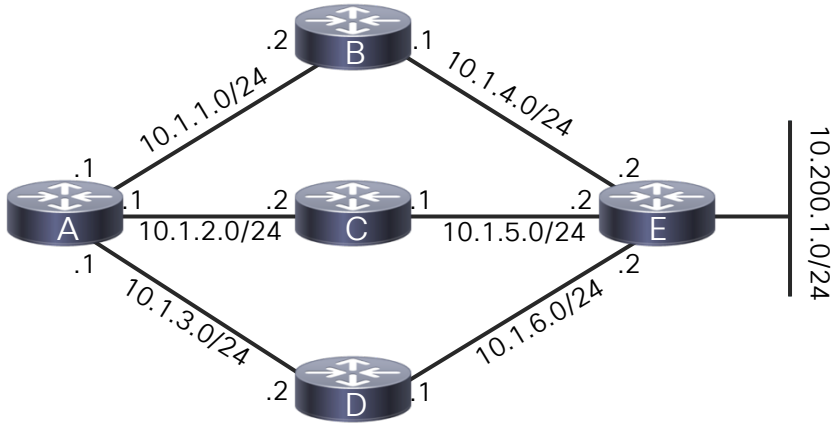
The Active Process

- What happens if RtrC ’s Reply isn’t sent, or doesn’t make it toRtrB?
- While RtrC is trying to send the Reply, RtrA ’s Active timer is running
- After 90 seconds, RtrA sends an SIA query to RtrB
- If RtrB is still waiting on RtrC , it sends an SIA reply to RtrA



Convergence of a Non-Feasible Successor

- Not as fast as Feasible Successor
- Requires co-operative processing with peers: active, query, reply.




```
RtrA#show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(1)/ID(10.1.6.1)
…..snip…..
P 10.200.1.0/24, 1 successors, FD is 21026560
         via 10.1.1.2 (21026560/20514560), Serial1/0
         via 10.1.2.2 (46740736/20514560), Serial1/1
         via 10.1.3.2 (46740736/46228736), Serial1/2
```

Feasible distance

Successor

Feasible successor

Possible successor

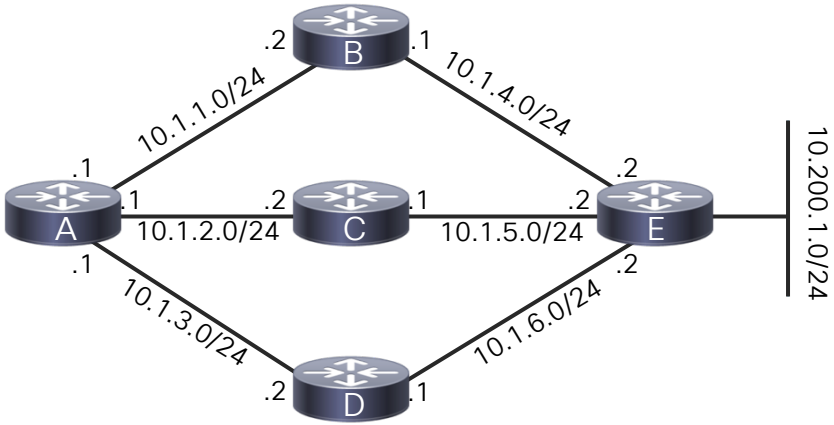
Computed distance

Reported distance

RD (46228736) is not < FD (21026560); NO FS

Convergence of a Non-Feasible Successor

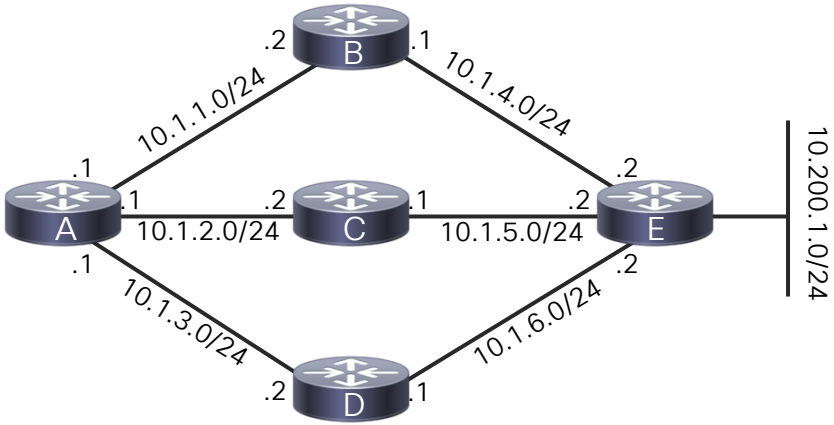
- Show ip eigrp event
- Look for FC not sat, transition to Active




```
RtrA#show ip eigrp event
169   12:04:09.627 State change: Local origin Successor Origin
170   12:04:09.627 Metric set: 10.1.4.0/24 metric(Infinity)
171   12:04:09.627 Active net/peers: 10.1.4.0/24 2
172   12:04:09.627 FC not sat Dmin/met: metric(47360) metric(20480)
173   12:04:09.627 Find FS: 10.1.4.0/24 metric(20480)
174   12:04:09.627 Rcv query met/succ met: metric(Infinity) metric(Infinity)
175   12:04:09.627 Rcv query dest/nh: 10.1.4.0/24 10.1.1.2
```

Convergence of a Non-Feasible Successor

- Show ip eigrp event, cont'd.
- Look for Rcv reply, and Route installed




```
RtrA#show ip eigrp event
…
110   12:04:09.659 Route installing: 10.1.4.0/24 10.1.2.2
111   12:04:09.659 Route installed: 10.1.4.0/24 10.1.3.2
112   12:04:09.659 Route installing: 10.1.4.0/24 10.1.3.2
113   12:04:09.659 Route installing: 10.1.4.0/24 10.1.1.2
114   12:04:09.659 Send reply: 10.1.4.0/24 10.1.1.2
115   12:04:09.659 Find FS: 10.1.4.0/24 metric(Infinity)
116   12:04:09.659 Free reply status: 10.1.4.0/24
117   12:04:09.659 Clr handle num/bits: 2 0x0
118   12:04:09.659 Clr handle dest/cnt: 10.1.4.0/24 0
119   12:04:09.659 Rcv reply met/succ met: metric(48640) metric(22784)
120   12:04:09.659 Rcv reply dest/nh: 10.1.4.0/24 10.1.2.2
```

Active Process – When things don't go as planned

```
%DUAL-3-SIA: Route 10.1.1.0 255.255.255.0 stuck-in-active state in IP-EIGRP 100. Cleaning up
```

- If you reach this point, at least two problems have occurred:
  - A route went active – there's a network that went missing somewhere
  - It got stuck ☹
- Both the 'stuck' and the 'active' have already occurred prior to the message being logged!
  - Event-logs are your friend

The SIA Query Process

- SIA-queries are sent to a neighbor up to three times
  - May attempt to get a reply from a neighbor for a total of six minutes
  - If a Reply is not received by the end of this process, the route is considered stuck through this neighbor
- On the router that doesn't get a reply after three SIA-queries
  - Reinitializes neighbor(s) who didn't answer
  - Goes active on all routes known through bounced neighbor(s)
  - Re-advertises to bounced neighbor all routes that were previously advertised

The SIA Query Process

- Sometimes the active process doesn't complete normally; this can be due to a number of different problems which are covered later in this presentation… what happens when things go wrong?
- If RtrB doesn't respond to RtrA within 1.5 minutes because it's still waiting for a Reply from RtrC, RtrA will send an SIA-query to RtrB checking the status—if RtrB is still waiting for a Reply itself, it will respond to RtrA with an SIA-reply; this resets the SIA timer on RtrA so it will wait another 1.5 minutes
- Eventually, the problem keeping RtrC from responding to RtrB will take the neighbor relationship down between RtrB and RtrC, which will cause RtrB to reply to A, ending the Query process

Troubleshooting SIAs

- Two (probably) unrelated causes of the problem – stuck and active
- Need to troubleshoot both parts
  - Cause of active often easier to find
  - Cause of stuck more important to find

Troubleshooting SIAs

- If routes never went active in the network, we would never have to worry about any getting stuck; unfortunately, in a real network there are often link failures and other situations that will cause routes to go active—one of our jobs is to minimize them, however
- If there are routes that regularly go active in the network, you should absolutely try to understand why they are not stable; while you cannot ensure that routes will never go active on the network, a network manager should work to minimize the number of routes going active by finding and resolving the causes
- Even if you reduce the number of routes going active to the minimum possible, if you don't eliminate the reasons that they get stuck you haven't fixed the most important part of the problem; the next time you get an active route, you could again get stuck
- The direct impact of an active route is small; the possible impact of a stuck-in-active route can be far greater

Troubleshooting the Active Part of SIAs

- Determine what is common to routes going active
  - Known network problems?
  - Flapping link(s)?
  - From the same region of the network?
- Resolve whatever is causing them to go active (if possible)

Troubleshooting the Active Part of SIAs

- The syslog may tell you which routes are going active, causing you to get stuck. Since the SIA message reports the route that was stuck, it seems rather straight forward to determine which routes are going active. This is only partially true—once SIAs are occurring in the network, many routes will go active due to the reaction to the SIA; you need to determine which routes went active early in the process in order to determine the trigger
- Additionally, you can do show ip eigrp topology active on the network when SIAs are not occurring and see if you regularly catch the same set of routes going active
- If you are able to determine which routes are regularly going active, determine what is common to those routes—are links flapping (bouncing up and down) causing the routes (and everything behind it) to regularly go active?
- Are most or all of the routes coming from the same area of the network? If so, you need to determine what is common in the topology to them so that you can determine why they are not stable

Troubleshooting the Stuck Part of SIAs

- **Show ip eigrp topology active**
- Useful only while the problem is occurring
- If the problem isn't occurring at the time, it is very difficult to find the reason the routes are getting stuck

Troubleshooting the Stuck Part of SIAs

- Our best weapon to use to find the cause of routes getting stuck-in-active is the command show ip eigrp topology active; it provides invaluable information about routes that are in transition—examples of the output of this command and how to evaluate it will be in the next several slides
- Unfortunately, this command only shows routes that are currently in transition; it isn't useful after the fact when you are trying to determine what happened earlier—if you aren't chasing it while the problem is occurring, there aren't really any tools that will help you find the cause

Chasing Active Routes

Why Is RtrA Reporting SIA Routes?
Let's Look at a Problem in Progress

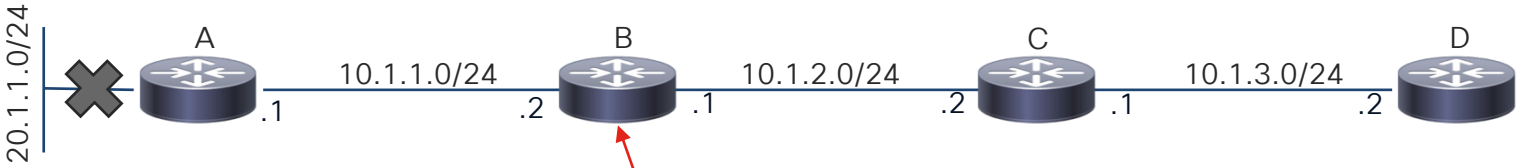
```
RtrA#show ip eigrp topology active
IP-EIGRP Topology Table for AS(1)/ID(20.1.1.1)
A 20.1.1.0/24, 1 successors, FD is Inaccessible
    1 replies, active 00:01:17, query-origin: Local origin
       via Connected (Infinity/Infinity), Ethernet1/0
      Remaining replies:
         via 10.1.1.2, r, Ethernet0/0
```

RtrA Is Waiting on RtrB

Chasing Active Routes

- In our example network, we've noticed dual-3-sia messages in the log of RtrA and we know the trigger is an unstable network off of this router; instead of just shutting down the unstable link, we decide to try to determine the cause of the stuck part of stuck-in-active
- In the above output, we see that RtrA is active on the route 20.1.1.0/24 (note the "A" in the left column) and has been waiting for an answer from 10.1.1.2 (RtrB) for one minute and 17 seconds—we know that we are waiting on RtrB because of the lower case r after the IP address; sometimes, the lower case r comes after the metric in the upper part of the output (not under remaining replies)—don't be fooled—the lower case r is the key, not whether it's under the remaining replies are or not
- Since we know why we are staying active on the route because RtrB hasn't answered us, we need to go to him (RtrB) to see why he's taking so long to answer

Chasing Active Routes



So Why Hasn't RtrB Replied?

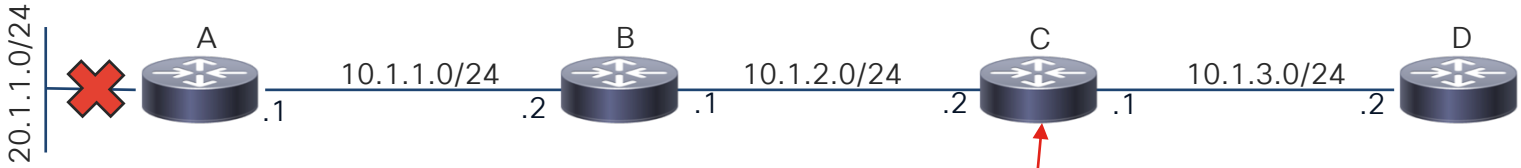
```
RtrB#show ip eigrp topology active
IP-EIGRP Topology Table for AS(1)/ID(10.1.2.1)
A 20.1.1.0/24, 1 successors, FD is Inaccessible
    1 replies, active 00:01:26, query-origin: Successor Origin
        via 10.1.1.1 (Infinity/Infinity), Ethernet0/0
      Remaining replies:
         via 10.1.2.2, r, Ethernet1/0
```

RtrB is Waiting on RtrC

Chasing Active Routes (Step 1)

- We repeat the show ip eigrp topology active command on RtrB and we get the results seen above
- We see that RtrB probably isn't the cause of our stuck-in-active routes, since it is also waiting on another router downstream to answer his query before it can reply; again, the lower case r beside the IP address of 10.1.2.2 tells us it is the neighbor slow to reply
- We now need to go to 10.1.2.2 (RtrC) and see why it isn't answering RtrB

Chasing Active Routes



What's RtrC's Problem?

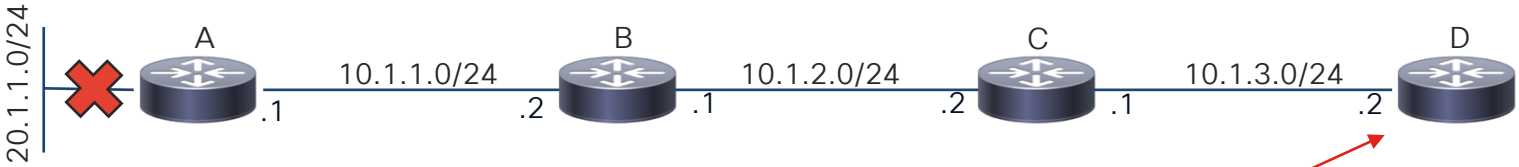
```
RtrC#show ip eigrp topology active
IP-EIGRP Topology Table for AS(1)/ID(10.1.3.1)
A 20.1.1.0/24, 1 successors, FD is Inaccessible, Qqr
    1 replies, active 00:01:33, query-origin: Successor Origin, retries(1)
        via 10.1.2.1 (Infinity/Infinity), Ethernet0/0, serno 20
        via 10.1.3.2 (Infinity/Infinity), rs, q, Ethernet1/0, serno 19, anchored
```

RtrC Is Waiting on RtrD

Chasing Active Routes (Step 2)

- On RtrC we repeat the show ip eigrp topology active command and see what it thinks of the route
- Again, he's waiting on another neighbor downstream to answer him before it can answer RtrB… you are probably getting the idea of how exciting this process can be; of course, in a real network you probably have users/managers breathing down your neck making it a bit more interesting
- As I'm sure you suspect our next step should be to see why 10.1.3.2 (RtrD) isn't answering RtrC's query

Chasing Active Routes



Why Isn't RtrD Answering?

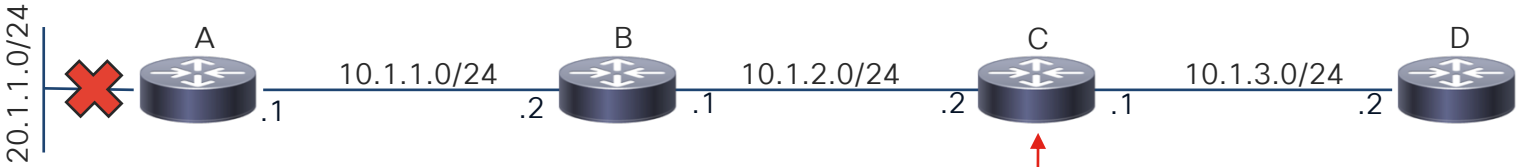
```
RtrD#show ip eigrp topology active
IP-EIGRP Topology Table for AS(1)/ID(10.1.3.2)
RtrD#
```

No active routes... back to RtrC!

Chasing Active Routes (Step 3)

- And again, we look at the active topology table entries, this time on RtrD
- Wait… RtrD isn't waiting on anyone for any routes; did the replies finally get returned and the route is no longer active? We need to go back to RtrC and see if it is still active on the route

Chasing Active Routes



No; RtrC Is Still Waiting on RtrD; What's the Deal?

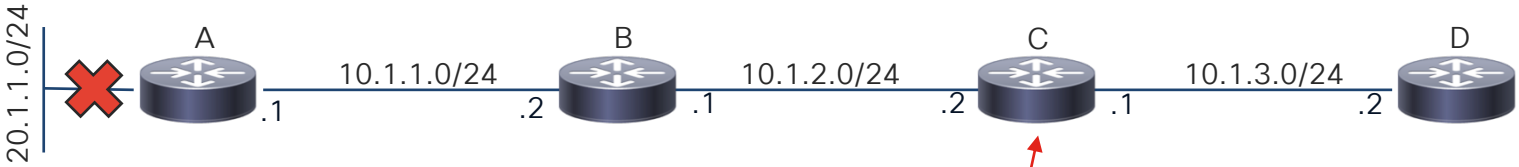
```
RtrC#show ip eigrp topology active
IP-EIGRP Topology Table for AS(1)/ID(10.1.3.1)
A 20.1.1.0/24, 1 successors, FD is Inaccessible, Qqr
    1 replies, active 00:01:52, query-origin: Successor Origin, retries(1)
        via 10.1.2.1 (Infinity/Infinity), Ethernet0/0, serno 20
        via 10.1.3.2 (Infinity/Infinity), rs, q, Ethernet1/0, serno 19, anchored
```

RtrC is waiting on RtrD

Chasing Active Routes (Step 4)

- Hmmm… RtrC still thinks the route is active and it's gotten even older
- There appears to be a problem, Houston. RtrC thinks it needs a reply from RtrD, yet RtrD isn't active on the route; we need to take a look at the neighbor relationship between these two routers to try to identify what is going wrong

Chasing Active Routes



We need to investigate and see why they don't seem to agree about the active route

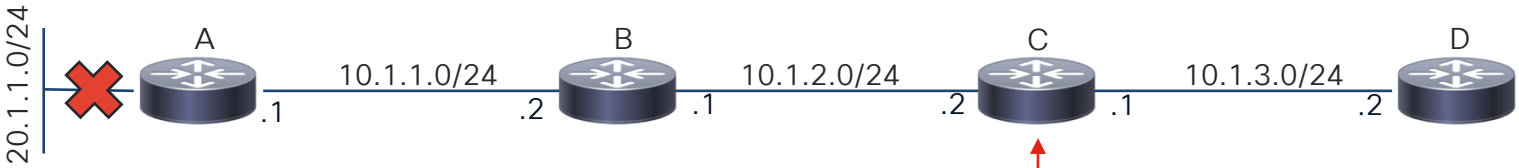
```
RtrC#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address      Interface Hold   Uptime    SRTT   RTO    Q   Seq
                                   (sec)     (ms)          Cnt Num
0   10.1.3.2     Et1/0     13   00:00:14     0    5000   1     0
1   10.1.2.1     Et0/0     13   01:22:54   227    1362   0     385
```

Looks like something's broken between RtrC and RtrD

Chasing Active Routes (Step 5)

- It appears that RtrC is having a bit of a problem communicating with RtrD—the neighbor relationship isn't even making it completely up based on the Q count on RtrC; we also notice in the log that the neighbor keeps bouncing due to retry limit exceeded
- Now we need to use our normal troubleshooting methodology to determine why these two routers can't talk to each other properly

Chasing Active Routes




```
RtrC#ping 10.1.3.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Okay—we can't ping; we need to fix this before EIGRP stands a chance of working

Chasing Active Routes (Step 6)

- How does basic connectivity look? A ping between RtrC and RtrD isn't succeeding either; we'll need to find out why they can't talk to each other
- Whatever is causing them to not talk to each other is undoubtedly a contributing factor to the SIAs we're seeing in the network; we need to find and fix the problem with this link and remove the cause of the SIA routes

Troubleshooting the Stuck Part of SIAs

- It's not always this easy to find the cause of an SIA
- Sometimes you chase the waiting neighbors in a circle
  - If so, summarize and simplify
- Easier after CSCdp33034 (circa 2000)
  - SIA should happen closer to the location of the cause of the problem
- CSCul80747 – introduces a new 'soft reset' for the SIA condition. Graceful Resync of the peer can be enabled by the soft-sia cli command.

Troubleshooting the Stuck Part of SIAs

- Our example of chasing SIA routes was intentionally made very easy in order to demonstrate the tools and techniques—in a real event on a network, there would probably be many more routes active, and many more neighbors replying; this can make chasing the waiting neighbors significantly more challenging
- Usually, you will be able to succeed at tracking the waiting neighbors back to the source of the problem—occasionally, you can't—on highly redundant networks, in particular, you can find yourself chasing neighbors in circles without reaching an endpoint cause of the waiting; if you run into this case, you may need to temporarily reduce the redundancy in order to simplify the network for troubleshooting and convergence

Likely Causes for Stuck-in-Active

- Bad or congested links
- Query range is “too long”
- Excessive redundancy
- Overloaded router (high CPU)
- Router memory shortage
- Software defects (seldom)

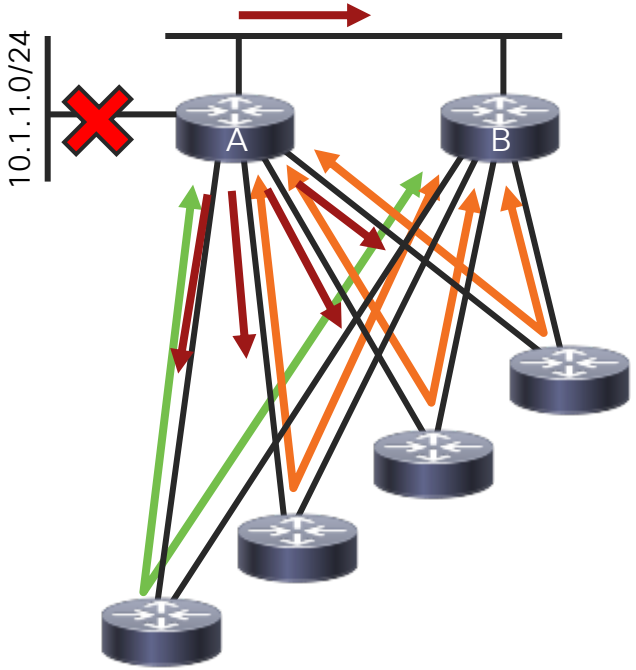
Likely Causes for SIAs

- Remember that the cause of the SIA route could be a different location than where the SIA message and bounced neighbors happened; this is particularly true with code older than CSCdp33034
- Some of the possible causes of SIAs are:
  - Links that are either experiencing high CRC or other physical errors or are congested to the point of dropping a significant number of frames—queries, replies, or acknowledgements could be lost
  - The time it takes for a query to go from one end of the network to the other is too long and the active timer expires before the query process completes; I don't think I've ever seen a network where this is true, by the way
  - The complexity in the network is so great due to excessive redundancy that EIGRP is required to work so hard at sending and replying to queries that it cannot complete them in time
  - A router is low on memory so that it is able to send hellos, which are very small, but be unable to send queries or replies
- There have occasionally been software defects that caused SIAs (CSCdi83660, CSCdv85419, CSCtc31545)

EIGRP Stub

EIGRP Stub

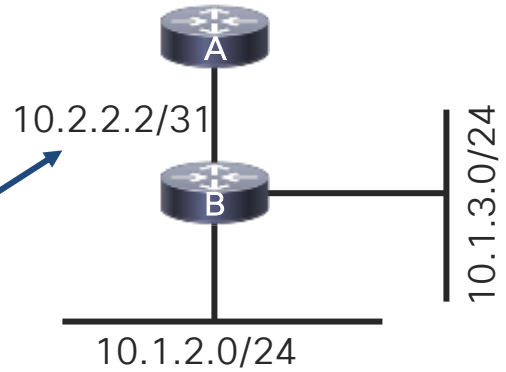
- Significant feature for EIGRP scalability and stability, one reason it is the best option for Hub and Spoke
- Marks a router (not an area) as the edge of the network, and as a non-transit
- Stub Rules:
  - Does not impact what Non-Stubs advertise to Stubs
  - Stubs may not advertise 'learned' routes from other neighbors
  - By default, configuring Stub means a router will advertise Connected and Summary routes
  - Stubs may Query non-Stub routers
  - Stubs should not be Queried
- Newer versions allow for a collection of devices at a branch to be marked as a Stub-site, relaxing some of the rules



Hub and Spoke (STUBs)

- At A, you can tell B is a stub using:

show ip eigrp neighbor detail



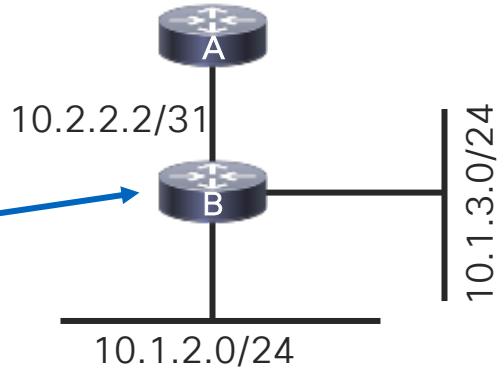

```
router-a#show ip eigrp neighbor detail
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.2.2.3                Se0               13 00:00:15    9   200  0  9
   Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 1
   Stub Peer Advertising ( CONNECTED ) Routes
   Suppressing queries
```

Hub and Spoke (STUBs)

- At B, you can see that the EIGRP process for AS 100 is running as a stub:

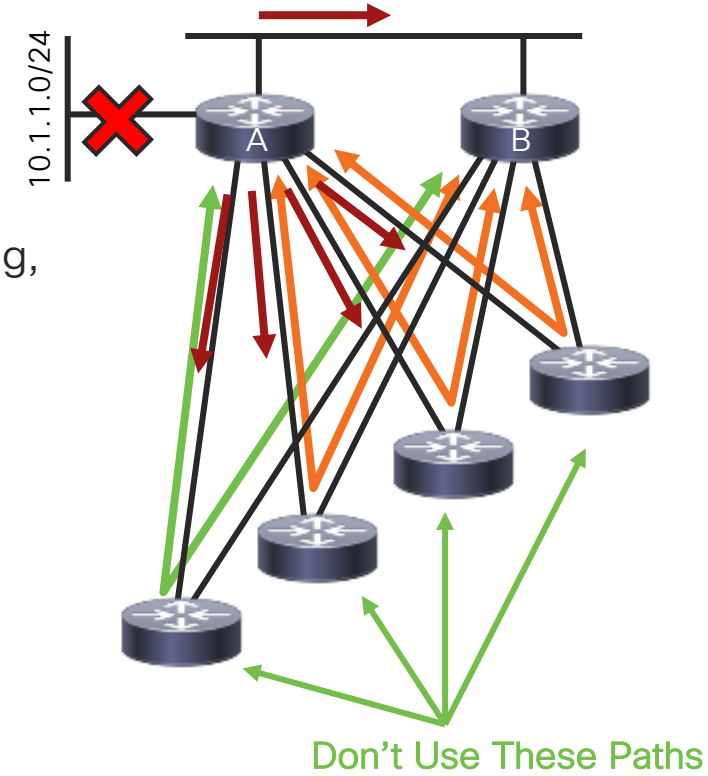
show ip protocols

```
router-b#show ip protocols
Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  EIGRP stub, connected
  Redistributing: static, eigrp 100
  .
  .
```



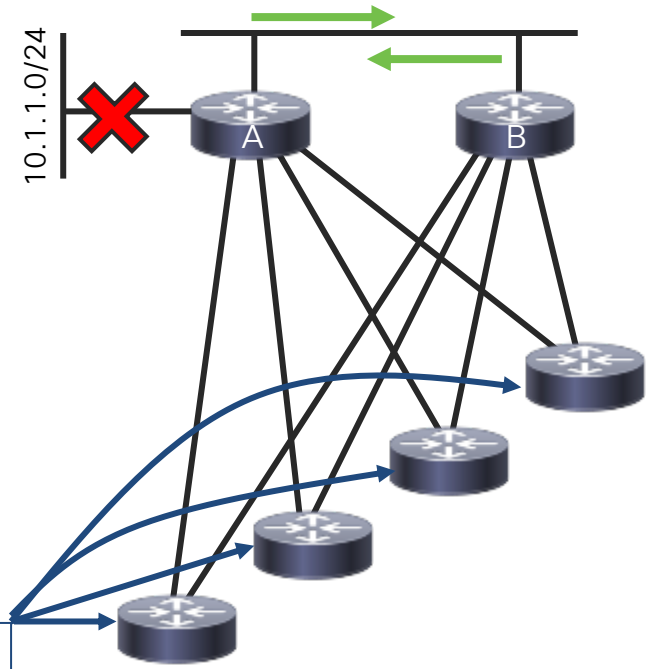
EIGRP Stub – How does it work?

- If these spokes are remote sites, they have two connections for resiliency, not so they can transit traffic between A and B
- A should never use the spokes as a path to anything, so there's no reason to learn about, or query for, routes through these spokes
- What happens when a route or link is lost?
  - EIGRP query's ALL neighbors
  - Each neighbors using it to reach the destination will also query their neighbors



EIGRP Stub – How does it work?

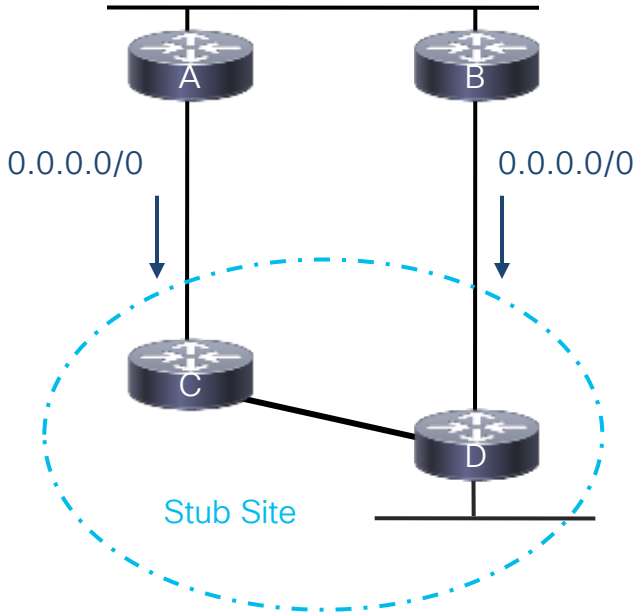
- Marking spokes as stubs allows the STUBs to signal A and B they are not valid transit paths
- A will not query stubs, reducing the total number of queries in this example to one
- Marking the remotes as stubs also reduces the complexity of this topology
- Router B now believes it only has one path to 10.1.1.0/24 (through A), rather than five




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

EIGRP Stub – Stub Site

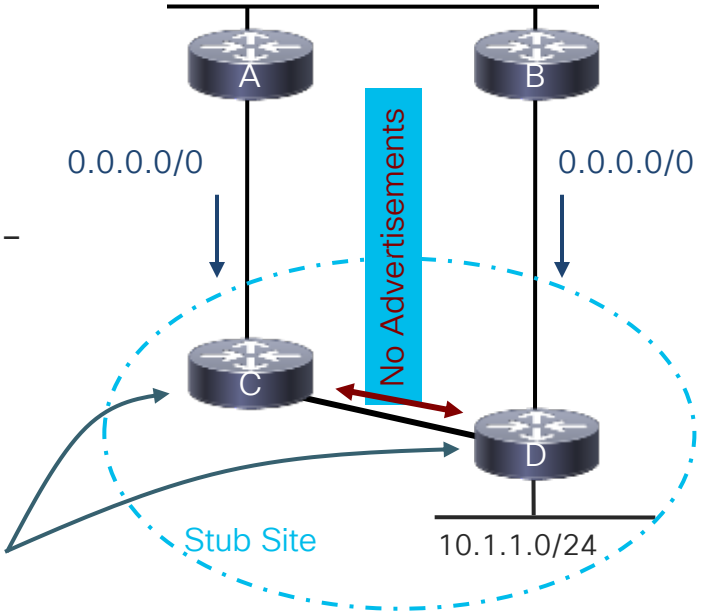
- Traditional stub behavior is designed for a single device
- Some deployments have remote sites with two routers and we want to mark the entire site as a “stub site”
- Stub-site was new functionality introduced in conjunction with IWAN Architecture
  - Simplifies branch site design
  - Easy to configure
  - Alternative to stub-leaking
- Intended to bring the benefits of Stub, Stub Leaking, and Loop Prevention to a branch near you!



EIGRP Stub – Stub Site

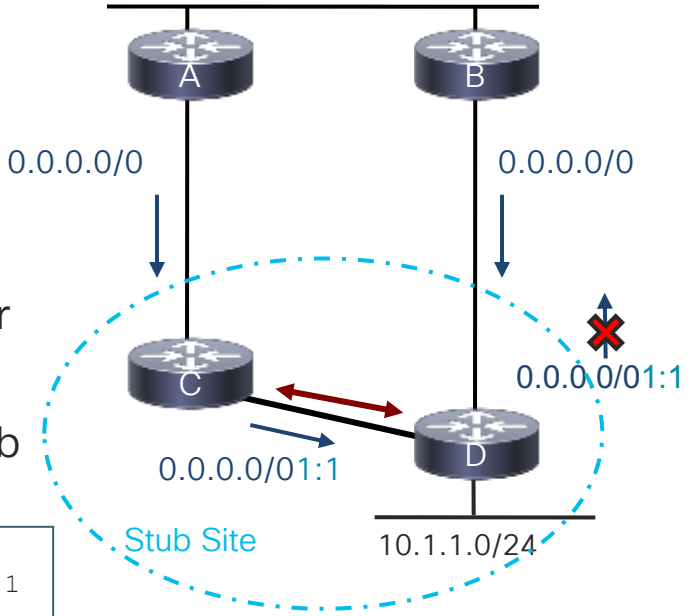
- Both routers at a location are configured with the SAME stub-site ‘site-id’ to create the stub-site
- *Normally* stubs C and D won’t advertise learned routes to each other, to override this, add the “stub-site” configuration

```
router eigrp ROCKS
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   stub-site wan-interface
  exit-af-interface
 eigrp stub-site 1:1
```



EIGRP Stub – Stub Site

- Routes learned INBOUND on the wan-interface are tagged with EXTCOMM value of the site-id.
- Routes with *any* site-id are automatically filtered OUTBOUND on any configured wan-interface.
- C and D will now exchange routes between each other in a normal manner, relaxing normal stub restrictions.
- Wan-interfaces will be marked as stub towards the hub routers, A and B and behave as stubs.

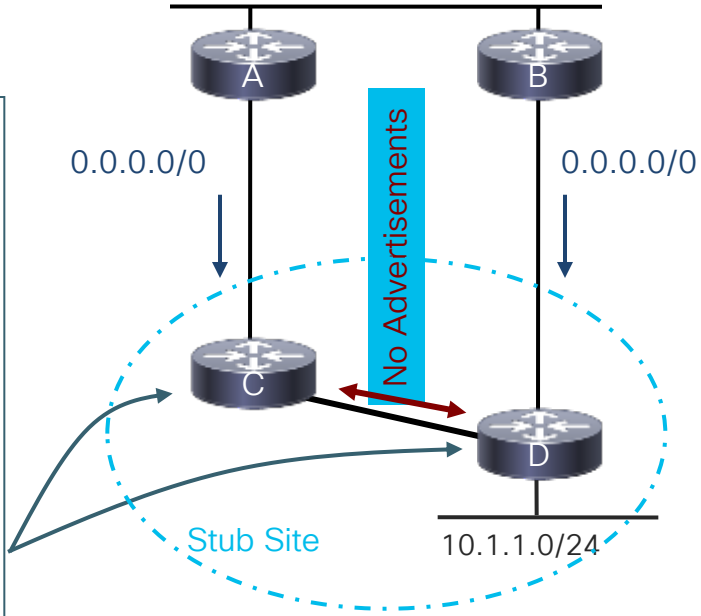



```
router eigrp ROCKS
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   stub-site wan-interface
  exit-af-interface
 eigrp stub-site 1:1
```

Wan Simplification with STUB-SITE

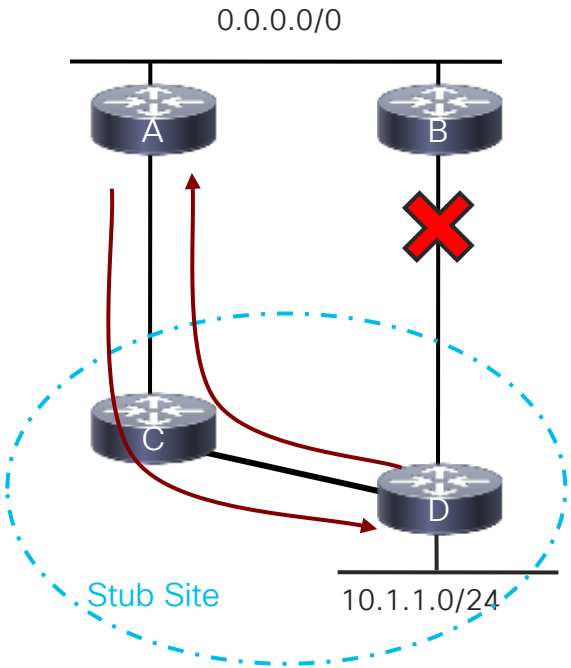
- Example branch configuration:

```
router eigrp ROCKS
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 10.0.0.0
  eigrp router-id 10.1.1.1
  eigrp stub-site 1:1
 exit-address-family
```



EIGRP Stub – Stub Site

- If the Router B to Router D link fails–
- 10.1.1.0/24 can now be reached from Router A
  - Since Router D is a stub-site, Router D will advertise 10.1.1.0/24 to Router C, who will advertise it to A
- Router D can now reach Router A, or anything behind Router A
  - Since Router C is a stub-site, Router C will advertise the default to Router D
- A and B should still not query C or D!



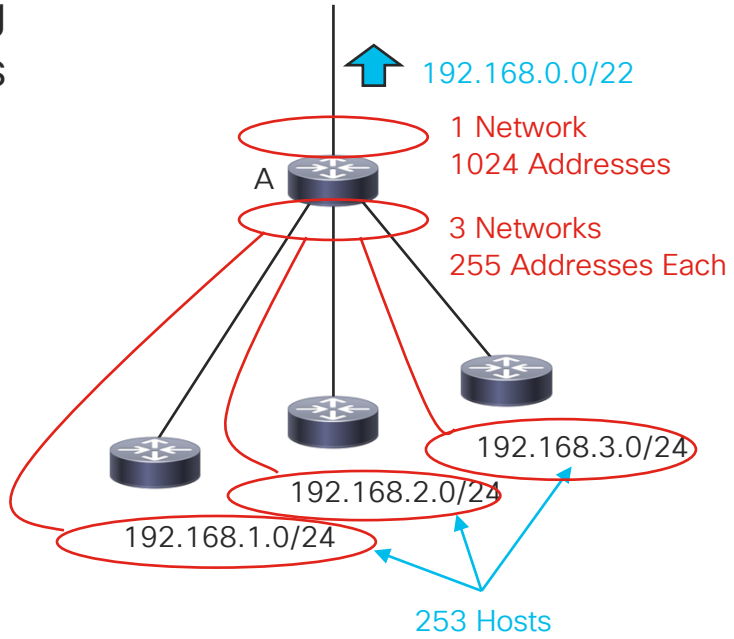
EIGRP Stub – Stub Site

- Some deployments have a single remote site with two routers and we want to mark the entire site as a “stub site”
- *Normally* stubs C and D won’t advertise learned routes to each other, to override this, add the “stub-site” configuration
- http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/15-mt/ire-15-mt-book/ire-iwan-simpl.html

EIGRP Summarization

EIGRP Summarization

- Summarization is an information hiding technique to send less-specific routes to represent block of prefixes
  - 192.168.1.0/24, 192.168.2.0/24, and 192.168.3.0/24 can be aggregated to 192.168.0.0/22
  - Rather than advertising three networks with each representing 255 addresses (253 hosts), RtrA advertises a single network, representing 1024 addresses
  - Also hides component route changes – usually!

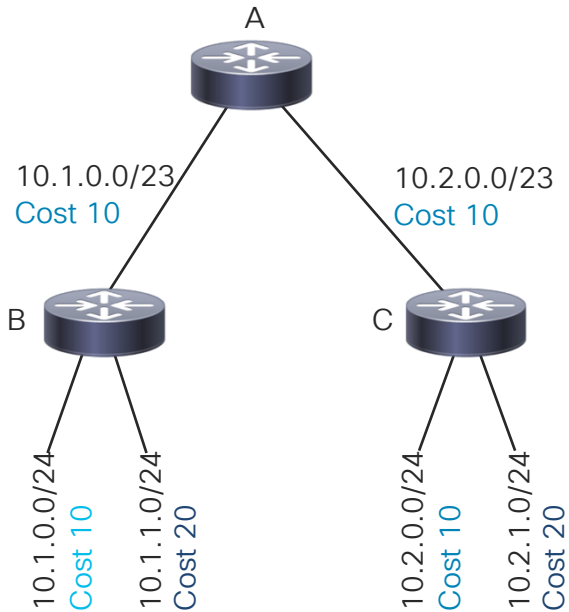


EIGRP Summary – Basics

- Summarization is an information-hiding technique used to minimize the number of prefixes advertised while still maintaining full reachability—summarization will be most effective if the network is designed in a hierarchical way so that multiple prefixes can be represented at some point in the network by a single, less specific prefix; one typical place of summarization is from distribution routers toward spokes that only need to know a default route (or at least some subset of total routes) in order to reach the remainder of the network
- When summarization is used in EIGRP networks, scalability is greatly enhanced both because of the fewer number of prefixes known throughout the network as well as the decreased query scope that summarization brings; the query scope aspect will be explained in more detail later in this presentation

EIGRP Summarization

- In EIGRP, the metric of a summary is based on the metrics of its components
- EIGRP chooses the metric of the lowest cost component route as the metric of the summary

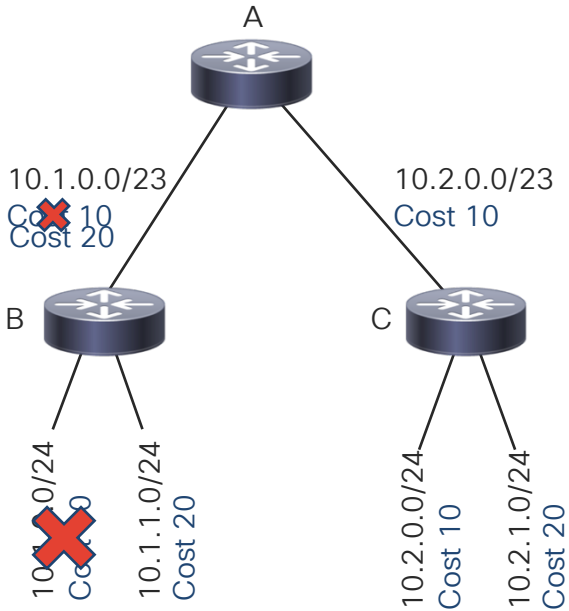


EIGRP – Summary Metrics

- When EIGRP creates a summary route, it has to determine the metric to include with the route advertisement—EIGRP examines every entry in the database (topology table) looking for components of the summary that will be suppressed (thus represented by) the summary; EIGRP finds the component with the best composite metric and then copies the metric details from it (bandwidth, delay, etc.) into the summary topology table entry
- Note that it does not take the best delay, best bandwidth, etc., but takes the best composite metric and grabs the attributes from it.
- This works fine except for the fact that components of the summary may come and go, which means EIGRP has to continually make sure the summary is still using the lowest metric contained in a summary component

EIGRP Summarization

- If the component from which the metric was derived flaps, then summary updates are required as well!
- The summary is used to hide reachability information, yet changes to the metric information causes the routers beyond the summary to perform work to keep up with the metric changes
- There is also processing overhead for EIGRP to recalculate the summary metric each time a component changes

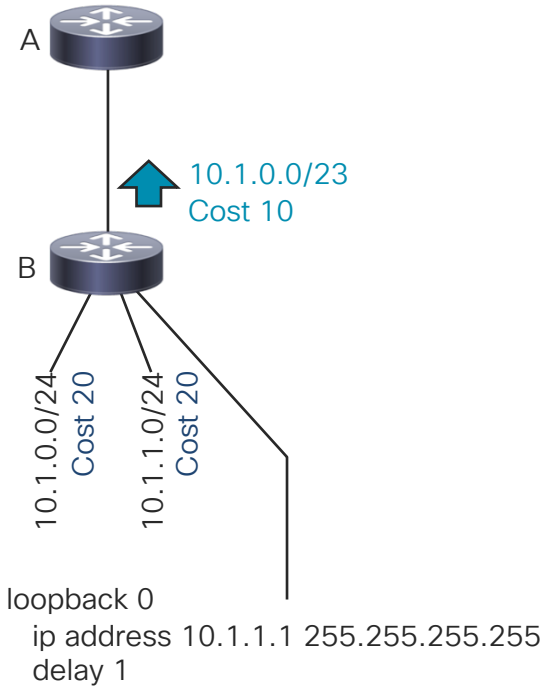


EIGRP Summary – Metrics

- This recalculation of the summary metric when components change causes two significant things to happen:
  - Every time the component with the best metric changes, the summary needs to be re-advertised to all of it's peers—thus the desire to hide topology changes behind the summary is only partially functional; while it hides the changes for each component prefix, it still causes updates and processing if the best component is the one that changed
  - Even if the best component isn't the one that changed, EIGRP internally has to look at every topology table entry to make sure the summary metric wasn't affected; with large numbers of components or large numbers of summaries, this can be significant processing

EIGRP Summarization – Solutions

- Use a loopback interface to force the metric to remain constant
  - Create a loopback interface within the summary range with a lower metric than any other component
  - Generally best to use a /32 for the prefix and use delay to force the metric value
  - The summary will use the metric of the loopback, which will never go down

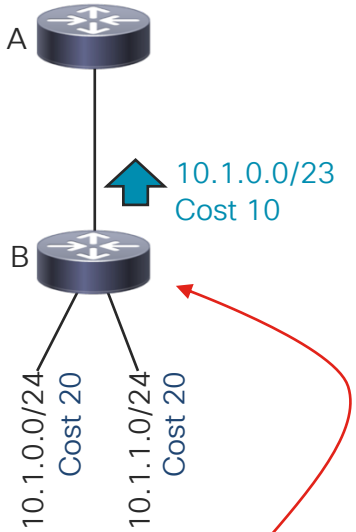


EIGRP Summary – Metrics

- One way to minimize/remove the first problem (metric changing downstream due to component changes) is to create a loopback on the router doing the summarization and ensure that it has the best metric of any component of the summary; since it will remain up unless administratively shut down, the metric of the summary will not change in its updates to upstream peers
- Note that this approach does nothing to change the second summary metric issue; i.e., router cpu processing required to recalculate on the router doing the summarization—that's next

EIGRP Summarization – Solution

- In recent EIGRP code, you can define the "summary-metric" command in router mode in order to specify the metric to be used on the summary, regardless of the metrics of the component routes
  - This is similar to defining the metric on redistribution statements in router mode
  - This eliminates metric churn downstream as well as local processing




```
router eigrp ROCKS
   address-family ipv4 autonomous-system 1
      network 10.0.0.0
      topology base
         summary-metric 10.1.0.0 255.255.254.0 10000 100 255 1 1500
```

Summary Metrics

- The recent implementations of EIGRP (release five and newer) contain the new “summary-metric” command under the router prompt which allows you to specify the metric to use on the summary so that learning the metric from summary components is unnecessary; since the metric is fixed, both the route churn problem for downstream peers and local database searching processing are removed
- This new command will greatly improve scalability in networks using summarization with large topology tables, which is where summarization is most useful!

Summary

Summary: What Have We Learned?

- EIGRP uses a Topology Table to store information about all network paths it knows
  - Successors are ‘best’ paths in use
  - Feasible Successors are pre-computed loop free alternate paths.
- EIGRP uses DUAL and the Active process to resolve loop free alternate paths when no Feasible Successor exists.
- Stub marks a device as the end of the path, thus “do not query”. Scalability!
- Summarization is essential to minimize info and unnecessary updates, maximizing scalability and stability.
- What’s next: Basic EIGRP Design Concepts, EIGRP Event-log, Redistribution
- Check out CiscoLive online for past EIGRP sessions for more details
  - BRKRST-2331 for EIGRP Troubleshooting
  - BRKRST-2336 for EIGRP Design

Agenda

- Section 1
  - Intro to Distance Vector
  - Basic EIGRP Configuration
- Section 2
  - Neighbors
  - Packets
  - Metrics
- Section 3
  - Topology Table
  - Convergence
  - Stub
  - Summarization
- Section 4
  - Basic Design
  - Event Log
  - EIGRP Redistribution

Agenda

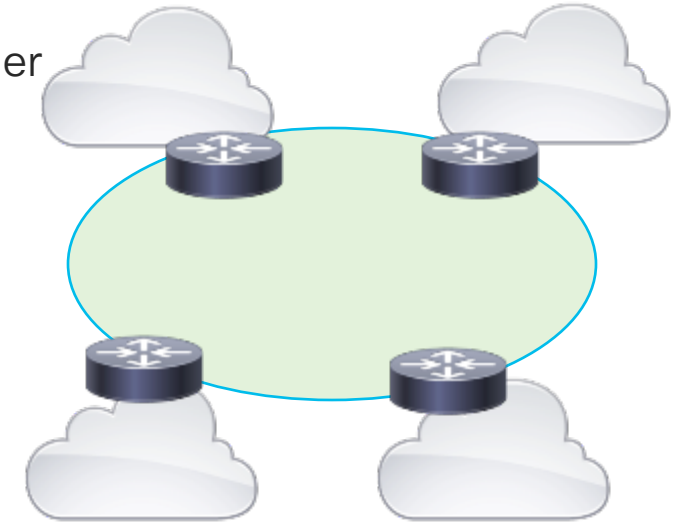
- Basic EIGRP Design
  - Core
  - Distribution/Access
  - WAN/Branch
- EIGRP Event-Log
- Redistribution
- Conclusion

EIGRP Design Goals

- Typical enterprise network is built upon multiple levels of routers and switches deployed in three general layers: access (to include WAN Aggregation), distribution, and core
- Universal Principle of Design: *Form follows Function!*
- Core:
  - Provides high speed connectivity between aggregation layers – move traffic from one area of the network to another.
- Distribution:
  - Provides aggregation of traffic flows from multiple Access layers to the Core. Traffic filtering and packet policies are typically implemented here. The distribution layer should be the blocking point for Queries
- Access:
  - Provide connectivity to user attachment points for servers, end stations, storage devices, and other IP devices. Consider use of EIGRP STUBS
- WAN Aggregation:
  - Provides connectivity to or through the internet and/or remote sites/offices.

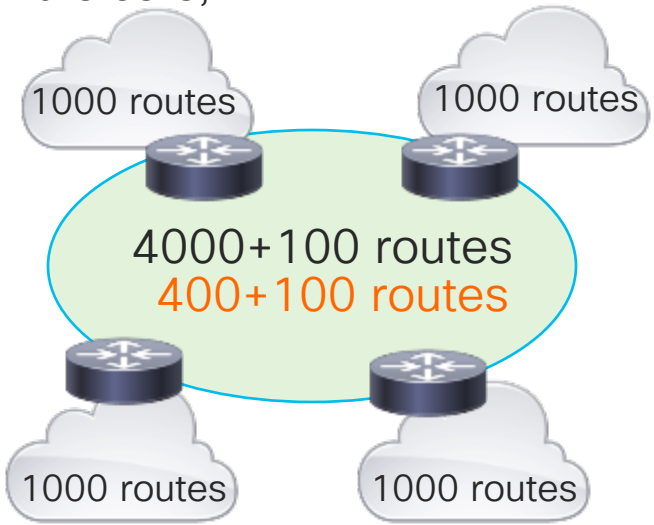
Core – Design Considerations

- Simplicity!
- Move traffic from one area of the network to another
- Segmentation and Domains
- Hierarchy
  - 2 Layer
  - 3 Layer
  - More
- Reliability vs Speed
  - Failure Detection
  - Graceful Restart(GR)
  - Non-Stop Forwarding(NSF)



Impact of Hierarchy to Core

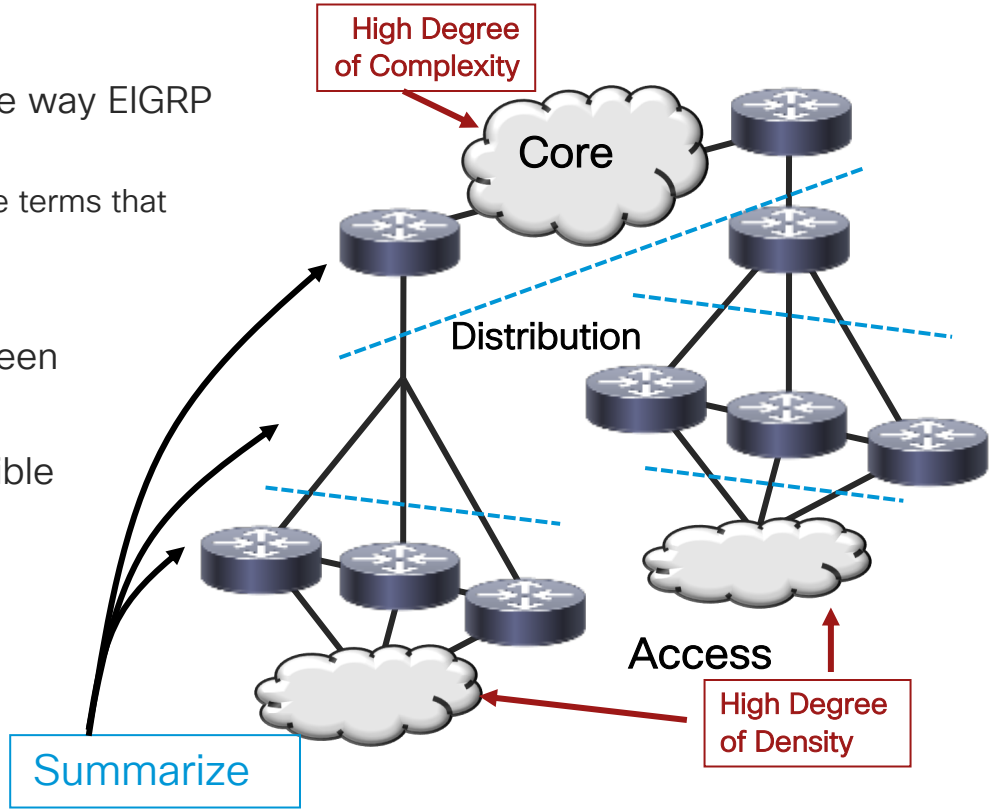
- As an example, let look at the impact of hierarchy on the core; Consider the following topology and assume;
- 4000 routes, each failing once/month means
  4100/30 = 136.7
  route changes per day in the core of this network
- Summarizing each 1000 route zone into 100 summary routes reduces the core to 500, rather than 4100 routes
- Summarization hides individual route changes, so we only see the 100 “core” routes change:
  100/30 = 3.3
  state changes per day in the core of this network



Hierarchy and the Core

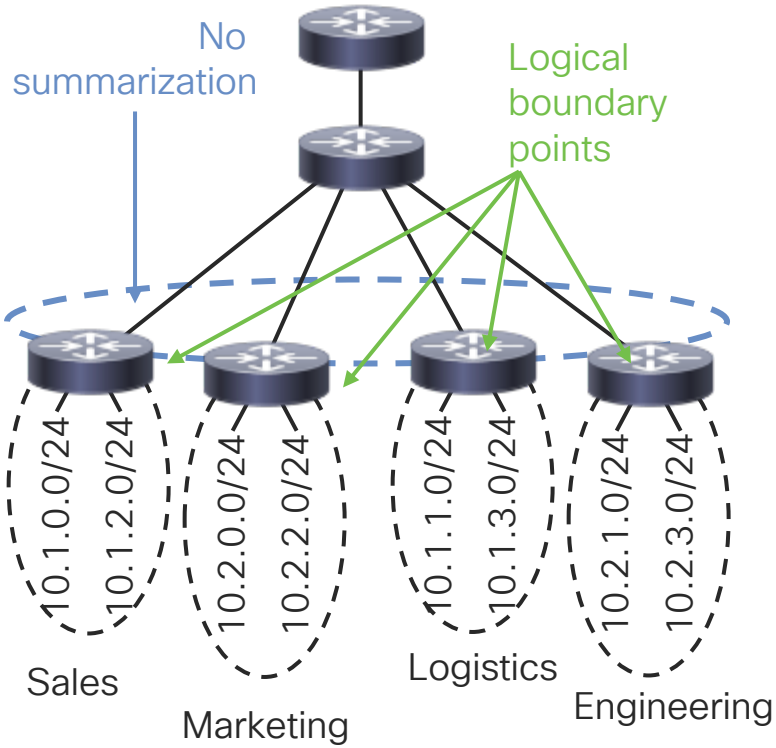
- The depth of the hierarchy doesn't alter the way EIGRP is deployed; there are no "hard edges"
  - "Core", "Distribution", and "Access" are flexible terms that may, or may not, fit your topology
  - EIGRP does not force these boundaries
- Core divides and isolates complexity between functional areas via summarization points
- Summarize at every boundary where possible
  - Aggregate reachability information
  - Aggregate topology information
  - Aggregate traffic flows
- Summary Points:

  Generally a good place to apply traffic policy



Hierarchical Design

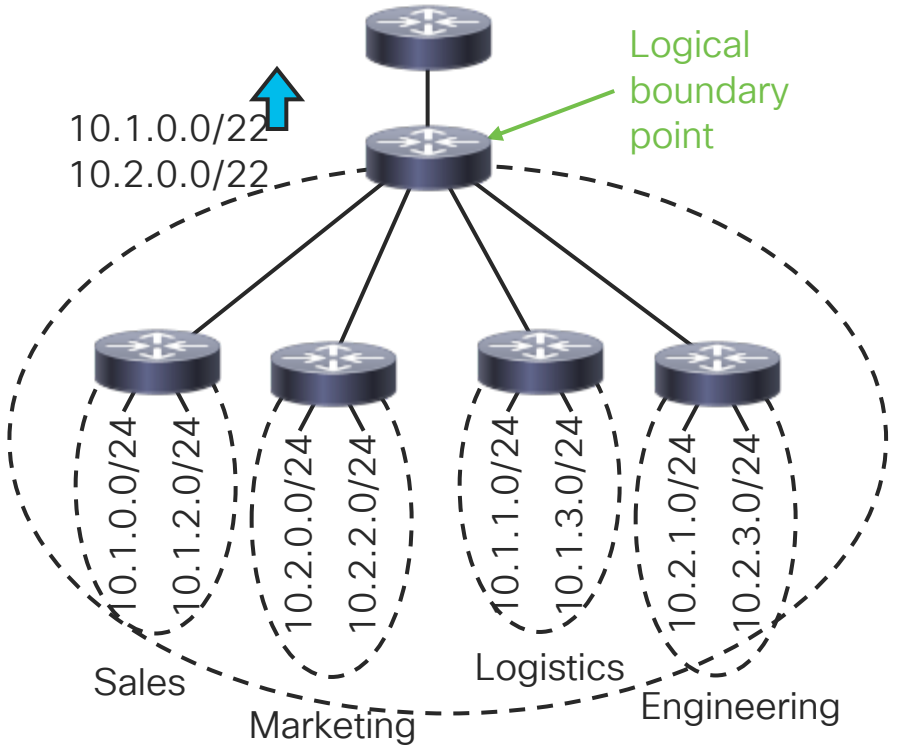
- No imposed limit on levels of hierarchy; a key design advantage.
- No “areas” or other restrictions on dividing a network
- Topology information can be hidden at any hop in the network anyway
- Hierarchy is created through summarization, rather than through a “protocol defined” boundary
- Proper addressing is a must to insure you can summarize



Hierarchical Design

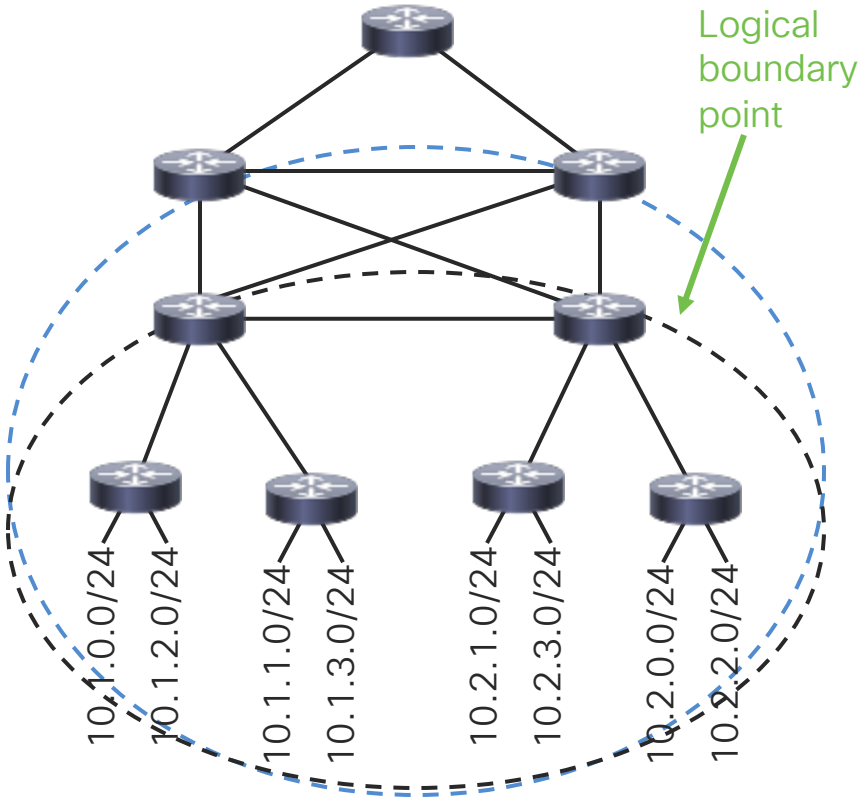
- The logical network structure no longer follows the corporate departments
- We now have a point at which we can summarize routes!

What Happens if We Move the Logical Boundary Point Up One Layer?



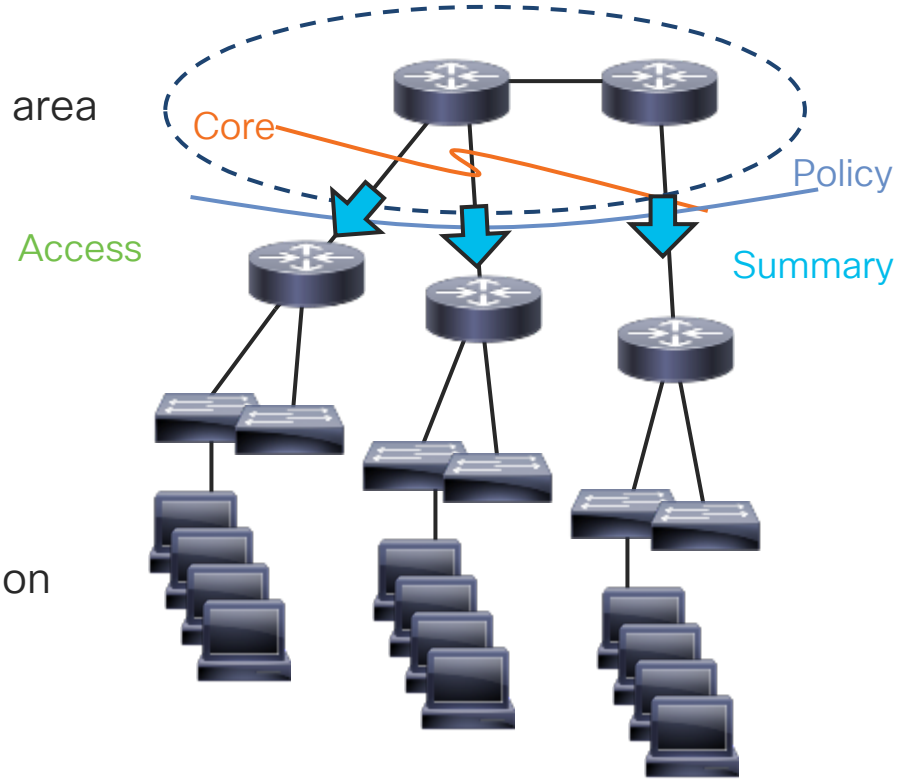
Hierarchical Design

- In this case, moving the logical boundary point down one layer can be used to improve summarization
- For EIGRP, it's just a matter of configuring summaries in the best possible locations



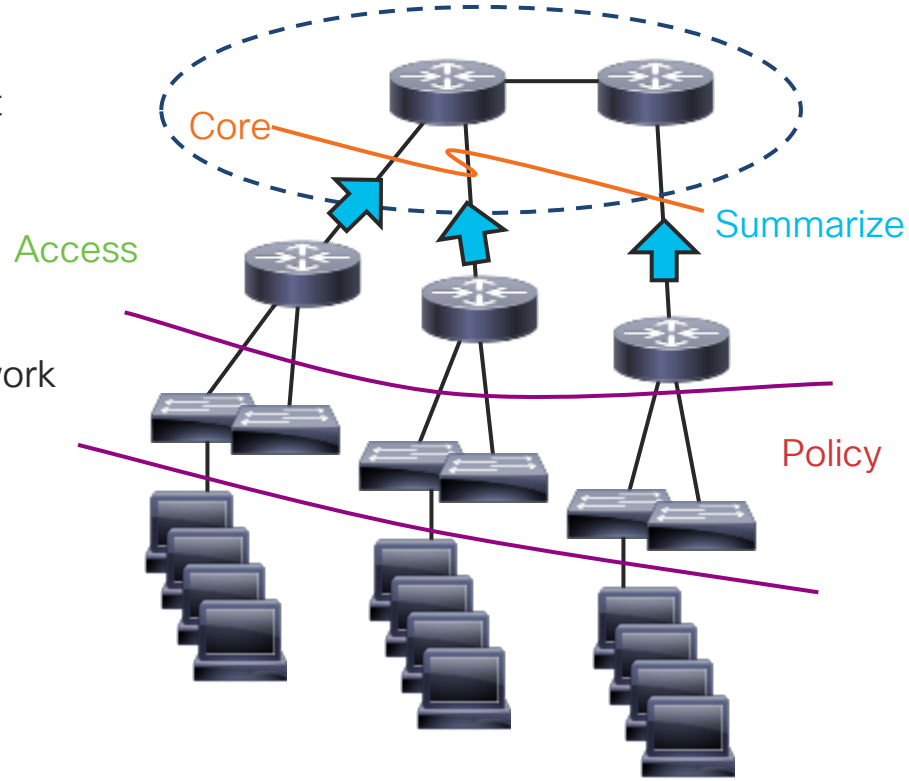
Two Layer Hierarchy

- The core gets traffic from one topological area of the network to another
  - High Speed Switching is the focus
- Within the core, **avoid:**
  - Policy configuration or enforcement
  - Reachability and topology aggregation (summarization)
- Core routers should summarize routing information towards the access/aggregation layers
- Routing policy may also be implemented at the core edge



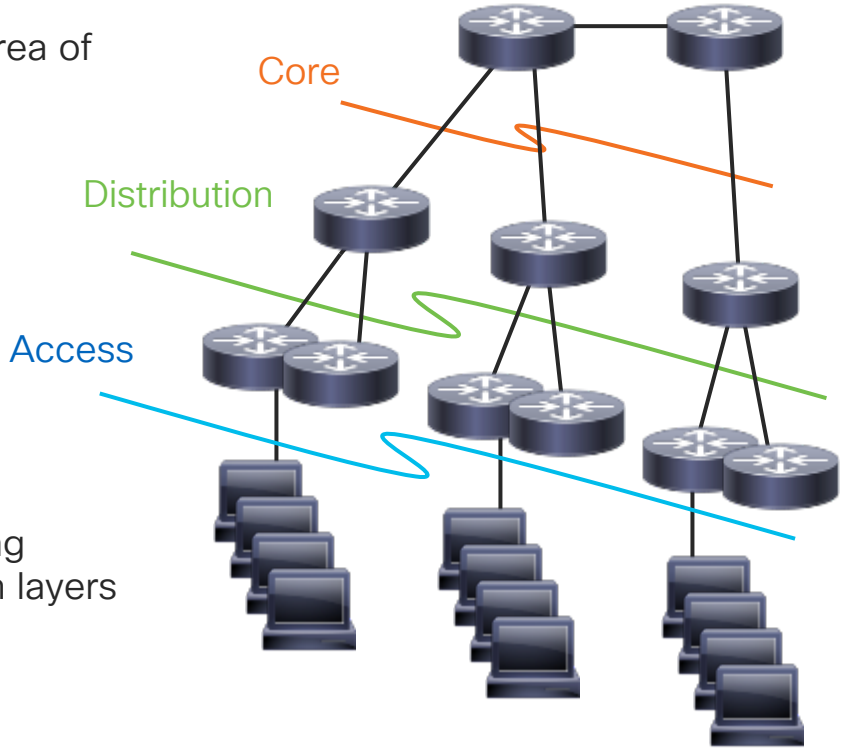
Two Layer Hierarchy

- The aggregation layer provides user attachment points
  - Information hiding (summarization)
  - Edge routes should be ‘hidden’ from the core
  - Summarize routes towards the core
- Policy should be placed at the edge of the network
  - Traffic acceptance (based on load and traffic type)
  - Filtering unwanted traffic
  - Security policy
- Layer 2 and Layer 3 filters apply at the edge



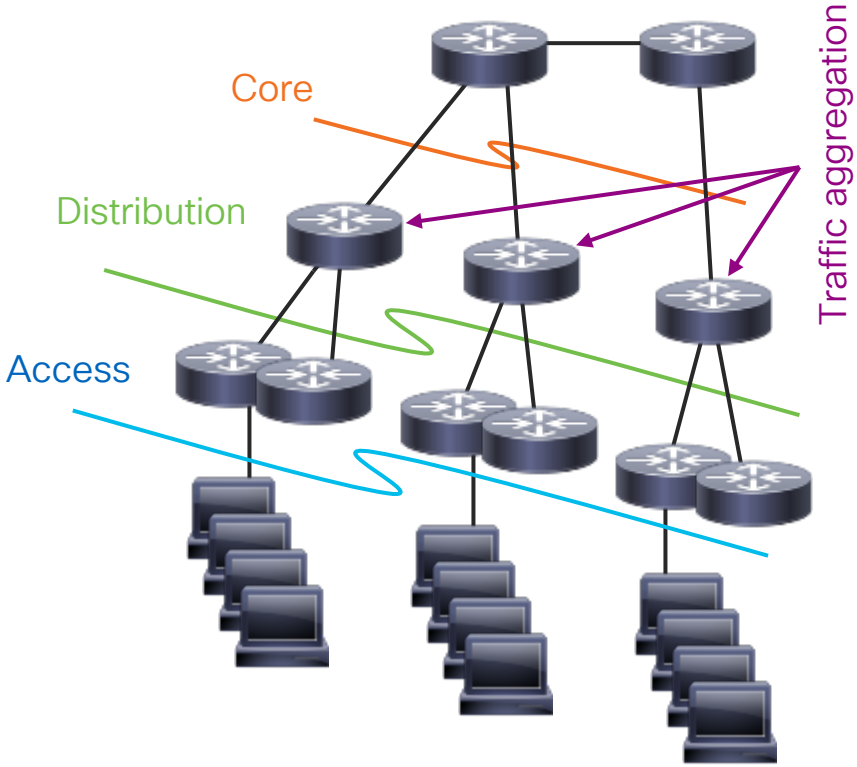
Three Layer Hierarchy

- The core gets traffic from one topological area of the network to another
  - High Speed Switching is the focus
- Within the core, **avoid:**
  - Policy configuration or enforcement
  - Reachability and topology aggregation (summarization)
- Core routers should summarize routing information towards the distribution layers
- Distribution routers should summarize routing information towards the access/aggregation layers
- Deeper hierarchy does not change EIGRP's fundamental design concepts



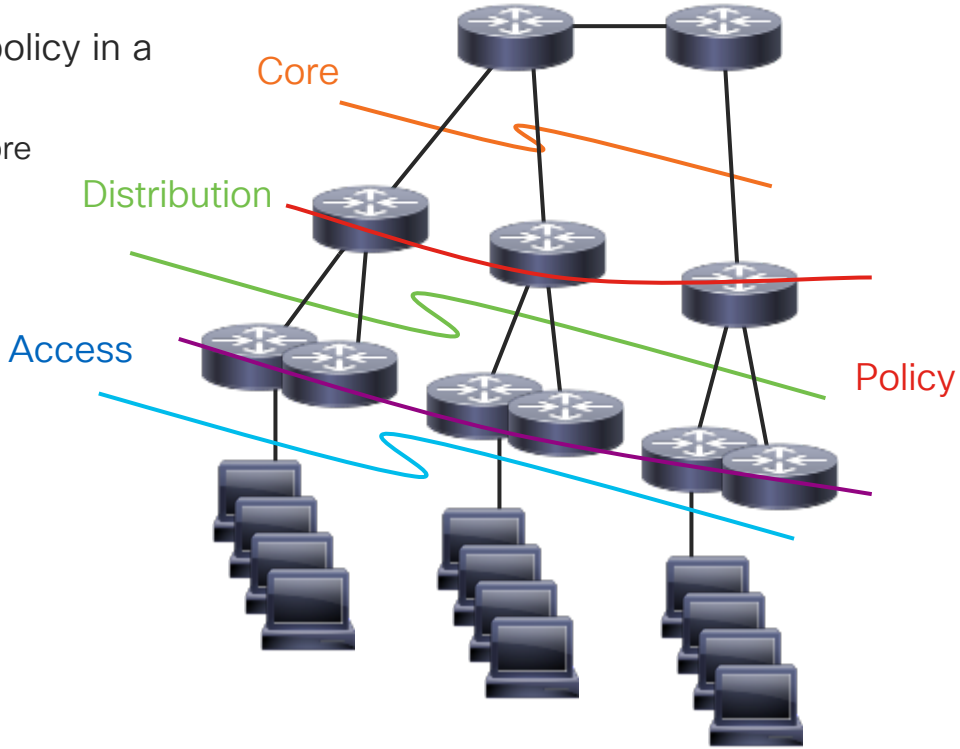
Three Layer Hierarchy

- Address summarization and aggregation occur at the distribution layer
- Address Summarization
  - At the distribution layer edge and the core
  - At the distribution layer edge and the access layer
  - At both edges of the distribution layer
- The distribution layer should be a blocking point for Queries
  - Provide minimal information toward the core
  - Provide minimal information toward the access
- Access layer routers should be considered for configuration as “stubs”



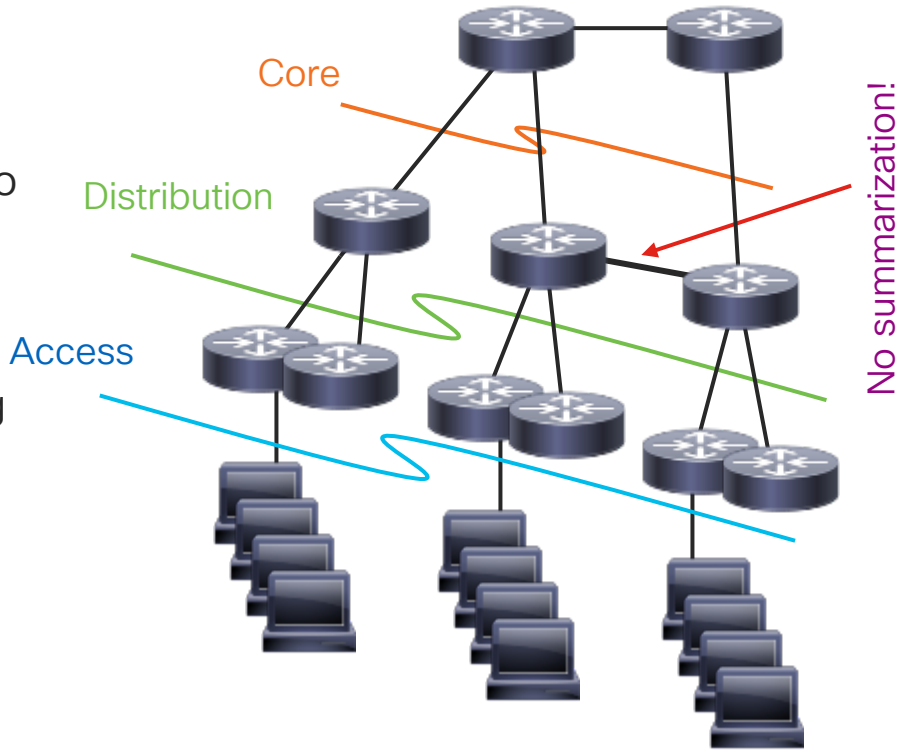
Three Layer Hierarchy

- The distribution layer is where most of the policy in a three layer network should reside
  - Should take all the policy load off the network core
- Traffic Engineering
  - Directing traffic into the best core entry point
  - Access layer failover
  - Traffic filters
- Routing Policy
  - Routes accepted from the access layer
  - Routes will be passed from the core into the access layer
  - Filtering unwanted traffic at Layer 2 and Layer 3
  - Security policy



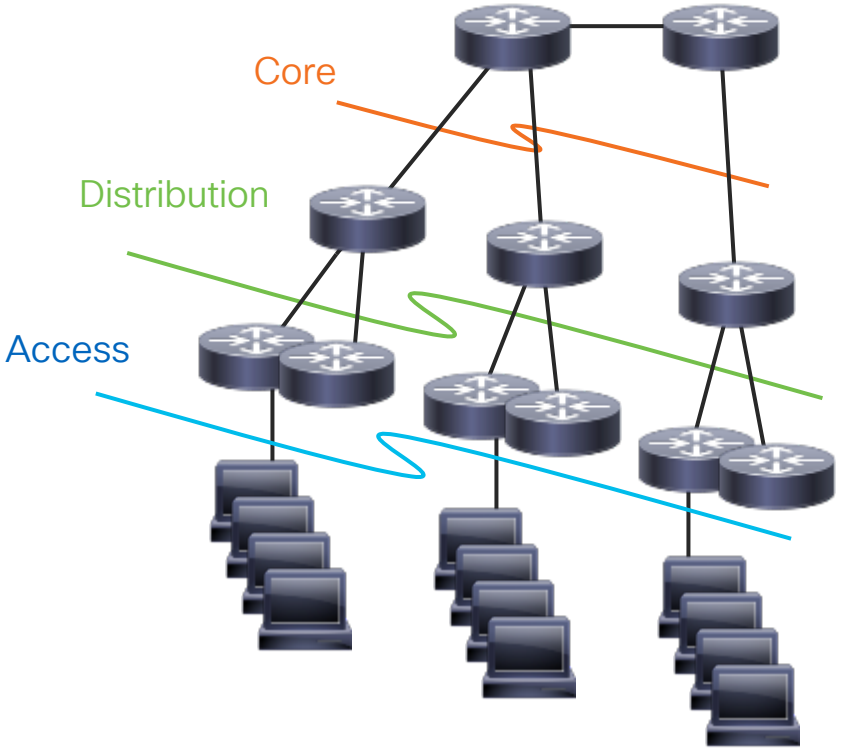
Three Layer Hierarchy

- Summarization should be avoided between distribution layer routers!
- This can cause a lot of odd and hard to troubleshoot problems within the network
- Focus summarization and policy up and down the layers, rather than along the layers



Distribution and Access – Design Considerations

- Aggregation!
- Distribution (aggregation point for access)
  - Summarization
    - Summary Metrics
    - Summary Leak-maps
  - Filtering
    - Route Map Support
    - Route Tag Enhancement
- Access (STUB and edge features)
  - Managing alternate paths
    - Passive interfaces
  - Hub and Spoke
    - Scaling
    - Enhancements
    - Stub-Site



Filtering and Route-Map Support

EIGRP supports Enhanced Route-Maps

- Enhanced support of route maps allows EIGRP to use a route map to prefer one path over another
- Route-maps can now be applied on the distribute-list in/out statement
- Filters can be applied even before the prefix hits the topology table

```
route-map setmetric permit 10
 match interface serial 0/0
 set metric 1000 1 255 1 1500
route-map setmetric permit 20
  match interface serial 0/1
 set metric 2000 1 255 1 1500
....
router eigrp ROCKS
  address-family ipv4 auto 4453
    topology base
    distribute-list route-map setmetric in
```

EIGRP Enhanced Route Map Support

match tag 100	matches against tags on internal routes
match tag external 100	matches against tags on external routes
match metric external 1000	matches against the external metric of an external route
match metric 1000 deviation 100	matches routes with metrics from 900 to 1100
match route-type external route-type bgp 65000	matches routes sourced from BGP autonomous system 65500
match route-type external route-type bgp 65000	matches routes sourced from BGP autonomous system 65500

EIGRP Enhanced Route Map Support

match ip next-hop 10.1.1.1	matches against the next hop listed in the route
match interface serial 0/0	matches against the interface the route was learned through
set metric 1000 1 255 1 1500	sets the component metrics for a route
set ip next-hop 10.1.1.1	sets the next hop listed in the route
set tag 100	sets the tag on internal routes (range limited to 1-255)
set tag external 100	sets the tag on external routes

Enhanced Routing Tagging

Enhanced Route Tags

- EIGRP has been extended to support a more flexible route tag method
  - Dotted-Decimal notation easer to read
  - Support mask for multiple tag matching
  - Supports IPv4 and IPv6

Classic Route Tag

```
route-map current-route-tag-usage permit 10
 match tag 451580 451597 451614 451631
 set metric 10000 10 200 5 1500
!
Router# show ip route tag
```

Enhanced Route Tag

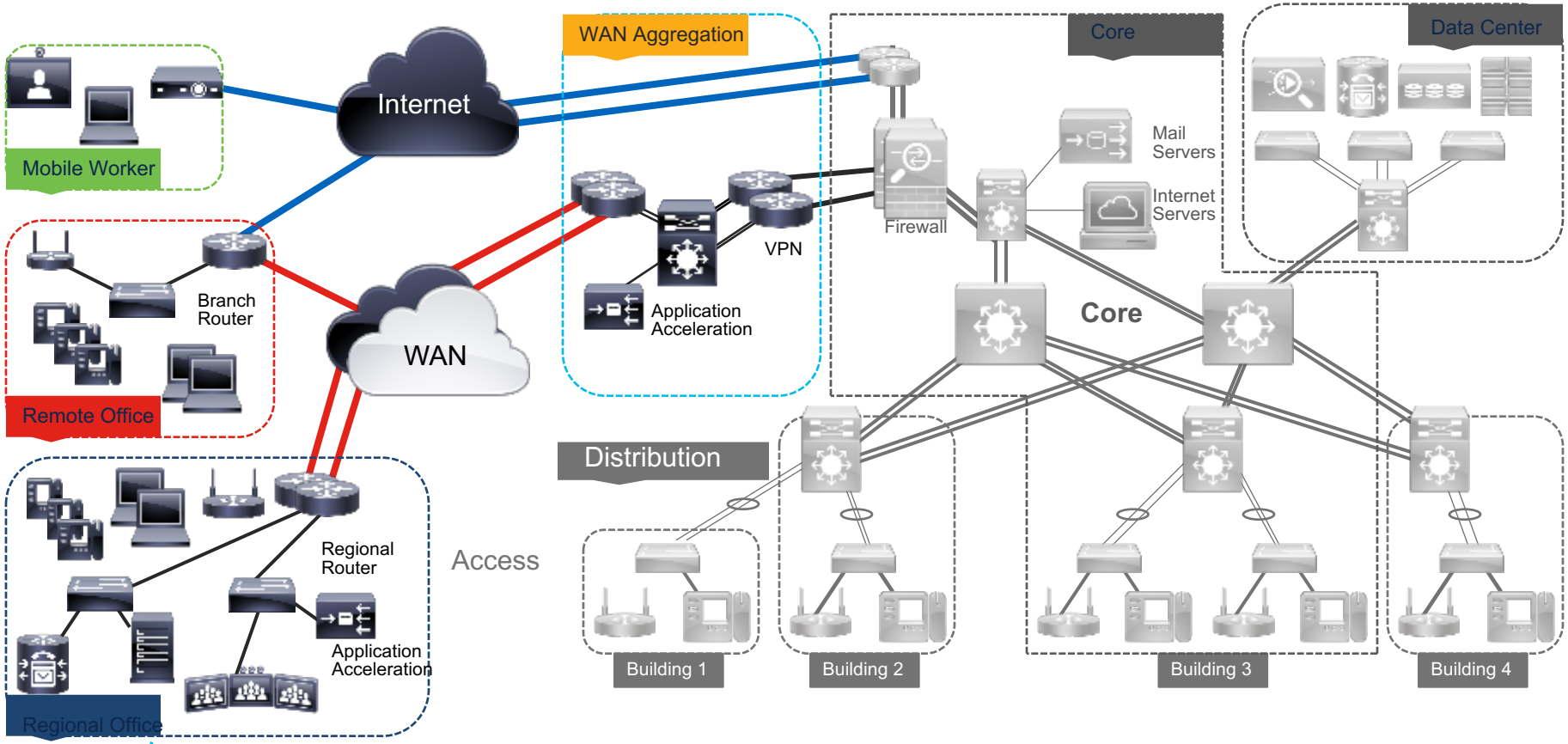
```
route-tag list enhanced-route-tag-usage
  permit 10.10.10.0 0.0.0.7
!
route-map OSPF-to-EIGRP
  match tag list enhanced-route-tag-usage
  set metric 10000 10 200 5 1500
!
router eigrp ROCKS
  address-family ipv4 vrf tagit autonomous-system 4452
    topology base
      redistribute ospf 2 route-map OSPF-to-EIGRP
```

Assigning routes a default tag

```
router eigrp ROCKS
 address-family ipv4 vrf tagit autonomous-system 4452
   topology base
    eigrp default-route-tag 10.10.10.10
```

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xe-3s/ire-xe-3s-book/ire-en-rou-tags.html

WAN



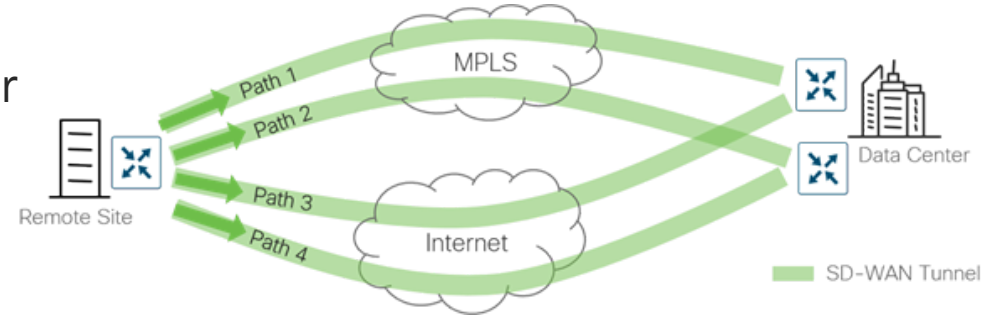
WAN Aggregation

Design Considerations

- Connecting Large Areas of the Network
- Optimal Path Selection with Security in mind
- Main Techniques
  - PE-CE (MPLS VPN)
  - SD-WAN (utilizing MPLS/Internet services)
    - Dual Home
    - Scaling
  - WAN Transparency – OTP
    - Attractive Alternative to PE-CE
    - Point-to-Point
    - Route Reflector

SD-WAN: EIGRP Support

- IOS XE SD-WAN devices only at this time
- vManage versions 19.1 and higher
- EIGRP routes not automatically redistributed into OMP, must be manually configured
- Uses an “SD-WAN Down Bit” to for loop prevention, as opposed to EIGRP SoO or tagging.



https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/cisco-sdwan-design-guide.html

EIGRP Design Goals

- Typical enterprise network is built upon multiple levels of routers and switches deployed in three general layers: access (to include WAN Aggregation), distribution, and core
- Universal Principle of Design: *Form follows Function!*
- Core:
  - Provides high speed connectivity between aggregation layers – move traffic from one area of the network to another.
- Distribution:
  - Provides aggregation of traffic flows from multiple Access layers to the Core. Traffic filtering and packet policies are typically implemented here. The distribution layer should be the blocking point for Queries
- Access:
  - Provide connectivity to user attachment points for servers, end stations, storage devices, and other IP devices. Consider use of EIGRP STUBS
- WAN Aggregation:
  - Provides connectivity to or through the internet and/or remote sites/offices.

EIGRP Design in Modern Networks

Action Plan – Parting Thoughts

- What sections of your network have identified with excessive complexity?
- Where can you summarize? (with a summary metric!)
- What changes can you consider to improve upon scalability?
- What changes can you consider to improve upon convergence?
- When will you add EIGRP RFC 7868 to your ‘Must Read’ list?
- What can you consider in your EIGRP architecture to continue to extend business and network capabilities?
- Where can you learn more about EIGRP Deployment and Design?

Event Log

Event Log

- EIGRP keeps a log of common events for each AS, 500 lines by default, rotating.
- 500 lines are not very much; on a network where there is significant instability or activity, 500 lines may only be a second or two (or less) – you can change the size of the event log (if needed) by the command
  - eigrp event-log-size <number of lines>
    - IOS limits to half of available memory
  - If number of lines set to 0, it disables the log
- You can clear the event log by typing
  - clear ip eigrp event
- Most recent events are at the top of the log by default, so time flows from bottom to top. The <reverse> keyword lets you display it from oldest to newest.

Event Log

- Three different event types can be logged
  - EIGRP log-event-type [dual][xmit][transport]
- Default is DUAL—normally most useful
  - DUAL is the EIGRP FSM (decisions in finite state machine)
  - xmit and transport are different aspects of actually sending packets to peers
- Any combination of the three can be on at the same time
- Work is in progress to add additional debug information to event log

EIGRP Event Log

- The two primary weapons at your disposal are debugs and the event log; realize that the output of both debugs and the event log are cryptic and probably not tremendously useful to you (so why am I telling you about them?)
- There are times when the output of debugs or the event log is enough to lead you in a direction, even if you don't really understand all that it is telling you; don't expect to be an expert at EIGRP through the use of debugs or the event log, but they can help
- Don't forget, debugs can kill your router—don't do a debug if you don't know how heavy the overhead is; I may tell you below about some debugs, but don't consider this approval from Cisco to run them on your production network
- The event log is non-disruptive, so it is much safer; just display it and see what's been happening lately

Event Log

- New parameters available for showing the event log

```
Rtr2#show ip eigrp event ?
  <1-4294967295>  Starting event number
  errmsg          Show Events being logged
  reverse         Show most recent event last
  sia             Show Events being logged
  type            Show Events being logged
  |               Output modifiers
  <cr>
```

Event Log

```
RtrA#show ip eigrp events
Event information for AS 1:
1    01:52:51.223 NDB delete: 30.1.1.0/24 1
2    01:52:51.223 RDB delete: 30.1.1.0/24 10.1.3.2
3    01:52:51.191 Metric set: 30.1.1.0/24 4294967295
4    01:52:51.191 Poison squashed: 30.1.1.0/24 lost if
5    01:52:51.191 Poison squashed: 30.1.1.0/24 metric chg
6    01:52:51.191 Send reply: 30.1.1.0/24 10.1.3.2
7    01:52:51.187 Not active net/1=SH: 30.1.1.0/24 1
8    01:52:51.187 FC not sat Dmin/met: 4294967295 46738176
9    01:52:51.187 Find FS: 30.1.1.0/24 46738176
10   01:52:51.187 Rcv query met/succ met: 4294967295 4294967295
11   01:52:51.187 Rcv query dest/nh: 30.1.1.0/24 10.1.3.2
12   01:52:36.771 Change queue emptied, entries: 1
13   01:52:36.771 Metric set: 30.1.1.0/24 46738176
```

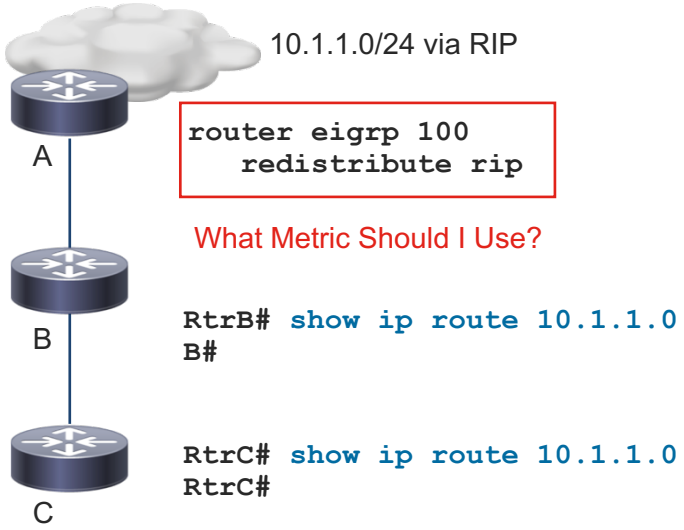
EIGRP Redistribution

EIGRP Redistribution

- Redistribution is the means of bringing routes – from the routing table, owned by another protocol instance, into another protocol.
  - Important: Only routes in the routing table can be redistributed!
  - Important: Metrics, for redistribution, are critical, to achieve proper routing and avoid loops.
  - Redistributed routes are marked as external and carry additional information to identify the point of redistribution.
- Route-maps are crucial, highly recommended, to explicitly define and control what routes are selected for redistribution. Design and Configure with Intent!

Redistribution Metrics

- RtrA is redistributing 10.1.1.0/24 from RIP, but RtrB and RtrC do not see the route
- EIGRP topology table on RtrA also does not show the route although it is learned in RIB
- Check whether RtrA has a redistribution metric configured via either
  - `default-metric` <metric>
  - `redistribute rip metric` <metric>
  - route-map used in `redistribute rip` statement
- EIGRP, in general, will not invent the metric components if they have not been specified explicitly (exceptions apply)



Redistribution Metrics

- More precisely, EIGRP can derive the redistribution metrics itself for:
  - **`redistribute connected`** – the interface metrics are taken
  - **`redistribute static`** – but only if the route points out an egress interface
  - **`redistribute eigrp`** – 1:1 copy of the other EIGRP process metrics
  - **`redistribute bgp`** – in MPLS L3VPNs if the original routes come from another site of the same VPN and were learned from EIGRP
- Otherwise, it is mandatory to specify the redistribution metric explicitly
  - On the **`redistribute`** statement, either via **`metric`** keyword, or in a route-map
  - Or with the **`default-metric`** statement
- Better safe than sorry – always specify the metrics explicitly!

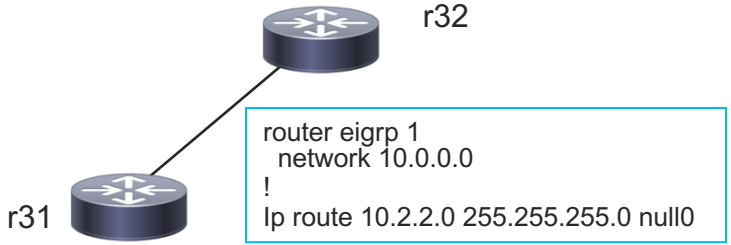
Redistribution Metrics

- Always use sensible redistribution metrics
  - `redistribute rip metric 1 1 1 1 1` will result in a very high initial EIGRP metric that will further grow as the route is advertised
  - Such metric is in risk of growing to a value that is considered an infinity
  - Beyond a certain hop, the route would apparently stop being learned
- The particular value of the redistribution metric has meaning only if there are multiple redistribution points to indicate preference
- No harm done if the metric is always specified using
  - BW = 1000000 (1Gbps), D = 1, RLY = 255, L = 1, MTU = 1500

Static Route to Connected Interface

- Another surprise you could hit is not really a “problem” but could be unexpected
- A static route defined only with the egress interface may be automatically redistributed
- This happens if the destination network is covered by a **`network`** statement

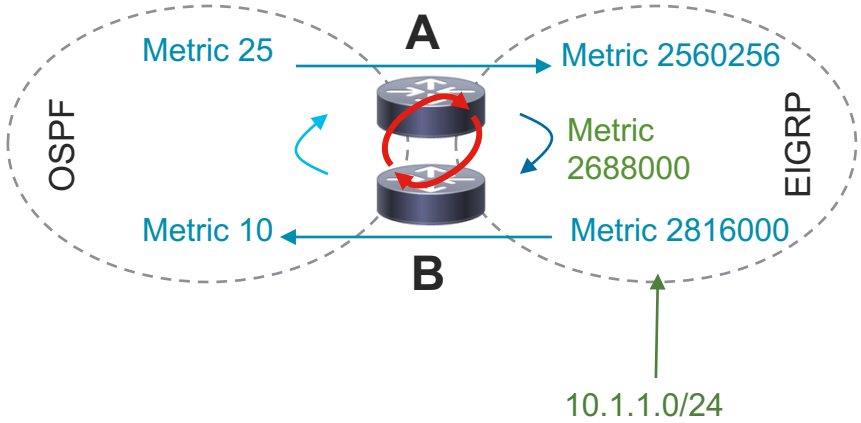
```
r32#show ip route
     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.2.0 is directly connected, Ethernet0/0
D       10.2.2.0 [90/281600] via 10.1.2.31, 00:19:20, Ethernet0/0
D       10.1.1.0 [90/307200] via 10.1.2.31, 00:24:19, Ethernet0/0
```




```
r31#show ip eigrp topology 10.2.2.0/24
IP-EIGRP (AS 1): Topology entry for 10.2.2.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 256
  Routing Descriptor Blocks:
  0.0.0.0, from Rstatic, Send flag is 0x0
      Composite metric is (256/0), Route is Internal
      Vector metric:
        Minimum bandwidth is 10000000 Kbit
        Total delay is 0 microseconds
        Reliability is 0/255
        Load is 0/255
        Minimum MTU is 1500
        Hop count is 0
```

Multiple Points of Redistribution

- A route is injected into EIGRP as an external; this route is redistributed into OSPF by RtrB
- The route is transmitted through OSPF to RtrA, who redistributes it back into EIGRP
- Depending on the manually set metrics, RtrB may prefer this redistributed route, building a routing loop
- Depending on the timing, the loop can be persistent or transient. Either way, a bad thing!



Redistribution Design

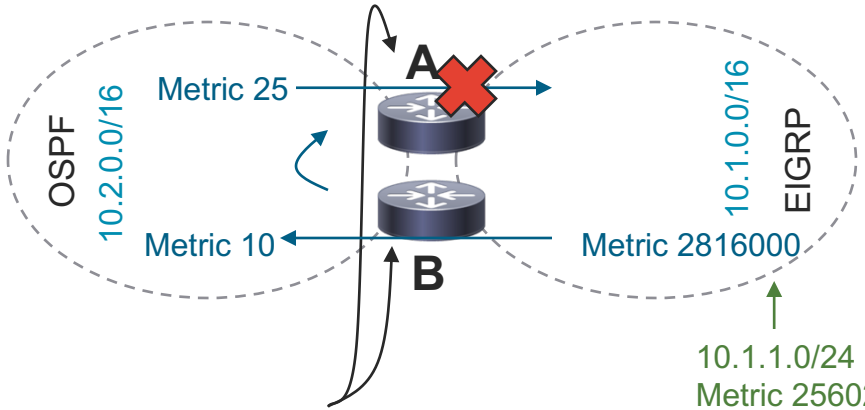
- There are three primary methods used to prevent this routing loop:
  - Redistributing live routing information in only one direction
  - Filtering routes based on their prefixes
  - Filtering routes using route tags
- The underlying goal is always the same
  - Prevent routes from being injected into where they originally came from
- Using route tags is typically the most flexible approach

Multiple Points of Redistribution

Filtering Based on Tags

- Set route tags when redistributing between the protocols; deny tagged routes at the redistribution point
- The route is injected into EIGRP as an external; it is redistributed into OSPF by RtrB and a tag is set
- The route is transmitted to RtrA through OSPF
- The route is blocked from being redistributed into EIGRP because of the route tag

```
route-map usetags deny 10
 match tag 1000
route-map usetags permit 20
 set tag 1000
```




```
router ospf 100
 redistribute eigrp 100 metric 10 route-map usetags

router eigrp 100
 redistribute ospf 100 metric 1000 1 255 1 1500 route-map usetags
```

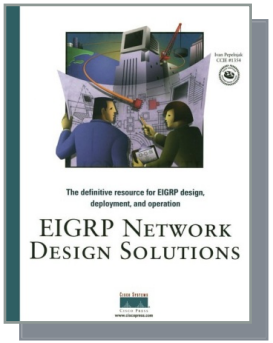
Summary

Summary: What Have We Learned?

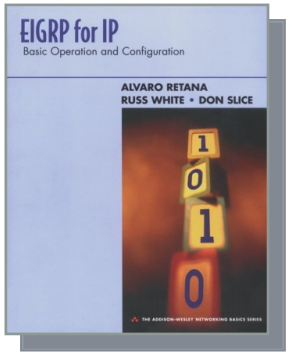
- EIGRP is no longer proprietary! RFC 7868
- EIGRP design – form follows function:
  - Core
  - Access/Distribution
  - Use summarization for information hiding, to help segment and scale.
- EIGRP's event-log knowledge is essential for operating a network with EIGRP
- Redistribution: Metrics, tags, route-maps – essential!
- Check out CiscoLive online for past EIGRP sessions for more details
  - BRKRST-2331 for EIGRP Troubleshooting
  - BRKRST-2336 for EIGRP Design & Deployment

Recommended Reading

EIGRP Specific Reading



ASIN: 1578701651

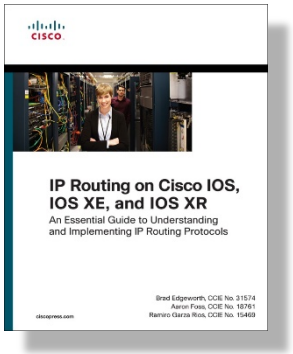


ISBN: 0201657732

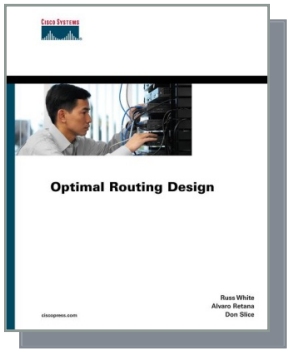
Open-EIGRP:
RFC 7868

Recommended Reading

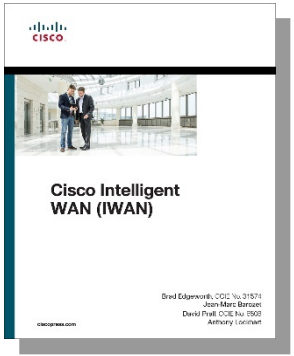
General Routing Reading, Including EIGRP



ISBN-13: 978-1587144233



ISBN 1587051877



ISBN-13: 978-1587144639
ISBN-10: 1587144638

Thank you

Possibilities

#CiscoLive