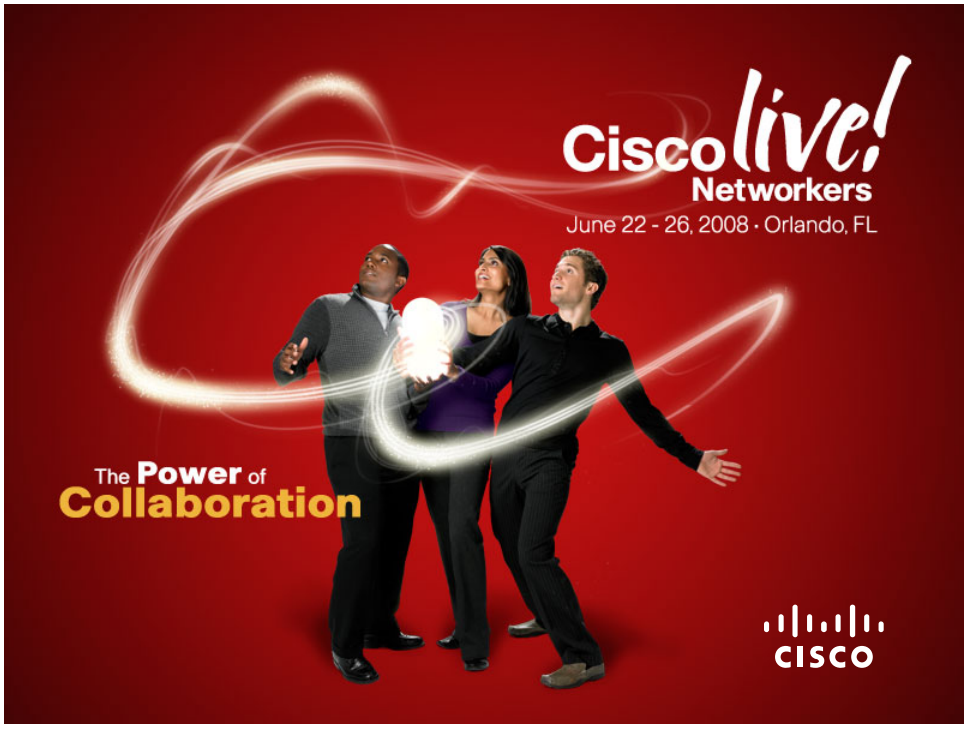
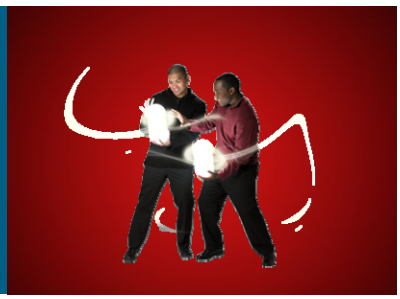


CISCO

Cisco live! Networkers

EIGRP Troubleshooting

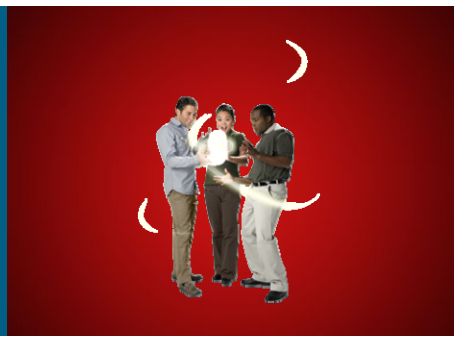


BRKRST-3330

EIGRP Troubleshooting

- Neighbors
- The Active Process
- External Routes
- Aggregation
- Troubleshooting Tools

Neighbors



Troubleshooting EIGRP Neighbors

- Neighbor process
- Checking neighbor status
- Log neighbor changes
- Holding time expired
- Retry limit exceeded
- Manual changes
- Unidirectional links
- Primary/secondary mismatch

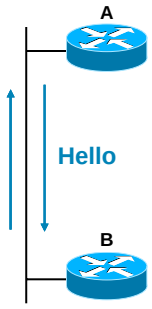
Neighbor Process

- Hello process used for neighbor discovery and maintenance
- Multicast hellos (by default)

  224.0.0.10
  (0100.5e00.000a)
- Hello interval

  60 seconds for low-speed NBMA

  5 seconds for all other interfaces



Neighbor Process

- The hello process has two distinct purposes
- Neighbor discovery

  When a hello packet is seen on an interface with EIGRP enabled, EIGRP looks at the information in the hello to see if it's legal for the sender to become a neighbor. The IP address must be a member of the same subnet, the AS number must match, and the k-values must match. If the sender is legal and isn't currently in the neighbor table, it is put into the neighbor table and the neighbor initialization process is started

- Neighbor maintenance

  Once a neighbor relationship is established, hello packets are sent in order to keep the neighbor relationship alive. Since EIGRP doesn't send periodic updates, the hello process is used to keep track of which neighbors are still functioning. As long as hellos are seen from a neighbor, his routes are considered still valid

Neighbor Process

- Hold timer

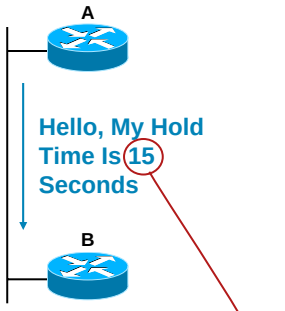
  How long to wait without hearing an EIGRP packet from a neighbor before declaring it dead

  Value contained in hello packets sent by each router

- Defaults to 3 X hello interval

  15 seconds on most interfaces

  180 seconds on low-speed NBMA




```
RTRB#show ip eigrp neighbors
H   Address   Interface Hold
                      (sec)
2   [A]       Et0       15
```


Neighbor Process

- The hold timer defines how long to wait without hearing an EIGRP packet from a neighbor before declaring them down (prior to CSCdi36031, it had to a hello received; that DDTs changed it so that any EIGRP packet will reset the timer, though normally it's still a hello that keeps the neighbor relationship alive)
- Each EIGRP router includes its own hold time in the hellos it sends. This allows it to define how long its neighbors should wait for it. This allows us to have different hello/hold timers on routers on the same subnet, unlike OSPF or IS-IS
- The hold time defaults to 3 X the default hello intervals; therefore, the default hold time for low-speed NBMA networks is 180 seconds (3 X 60 second hellos) and 15 seconds for all other interface types (3 X 5 seconds)
- Note: if you change the hello interval using the "ip hello-interval eigrp 1 <num>" command, it does not automatically change the hold time to 3 X the new value; you must also set the hold time to the value desired
- Note: CSCdr96531 changed the neighbor establishment process to make it much more reliable

Checking Neighbor Status

```
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
```

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

Checking Neighbor Status

- 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

Checking Neighbor Status

- 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

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


Checking Neighbor Status

- 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

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
Neighbor 10.1.1.1 (Ethernet0) is down: route filter changed
Others, but not often
```


Log-Neighbor-Changes Messages

- Peer restarted—the other router reset our neighbor relationship; you need to go to him to see why he thought our relationship had to be bounced
- 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)

What Causes Neighbor Instability?

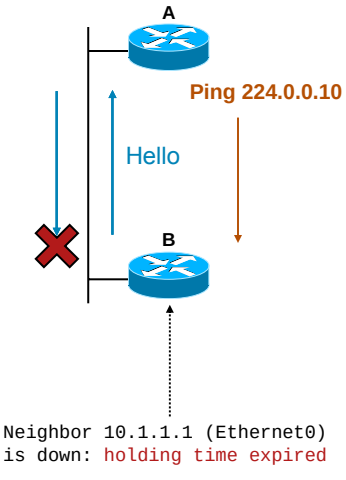
- Holding time expired
- Retry limit exceeded
- Manual changes
- Stuck-in-active routes
- Not as many as there used to be—more on this later

What Causes Neighbor Instability

- A number of network events can cause neighbor instability. With the use of the log-neighbor-changes messages, you can track down many of the causes. In this section, we'll describe what causes some of these messages (and resulting instability) and how to fix them.
- Note that with later code, most manual changes no longer cause neighbors to bounce. After CSCdy20284, many manual changes like summaries and filters now cause a resync between neighbors instead of a full-blown neighbor down. Quite an improvement.

Holding Time Expired

- The holding time expires when an EIGRP packet is not received during hold time
  - Typically caused by congestion or physical errors
- 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 (if you do several show ip eigrp neighbor commands, you should see the hold time value changing)
  - 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?
  - He may be gone (crashed, powered off, disconnected, etc.)
  - He (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 Expiration

```
RouterA# 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

Holding Time Expired

- Another troubleshooting tool available is to do the debug 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

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

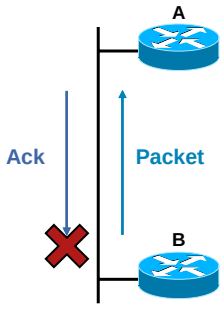
  Reliable packets are retransmitted up to 16 times if not acknowledged

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, he 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.
- You also see this symptom when a link with separate unidirectional connections is having a problem; for example, an ATM or Frame Relay link with a PVC working one direction but not the other; more on this later

Retry Limit Exceeded

- Reliable packets are re-sent after Retransmit Time Out (RTO)
  - Typically 6 x Smooth Round Trip Time (SRTT)
  - Minimum 200 ms
  - Maximum 5000 ms (five seconds)
  - 16 retransmits takes between 50 and 80 seconds
- If a reliable packet is not acknowledged before 16 retransmissions and the hold timer duration has passed, re-initialize the neighbor




```
Neighbor 10.1.1.1 (Ethernet0) is
down: retry limit exceeded
```

Retry Limit Exceeded

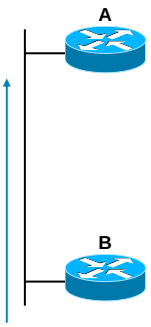
- The Retransmit Timeout (RTO) is used to determine when to retry sending a packet when an Ack has not been received, and is (generally) based on 6 X Smooth Round Trip Time (SRTT). The SRTT is derived from previous measurements of how long it took to get an Ack from this neighbor. The minimum RTO is 200 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 50 seconds (minimum interval of 200 ms with a max interval of 5000 ms). For example, If there isn't an acknowledgement after 200 ms, the packet is retransmitted and we set a timer for 300 ms. If it expires, we send it again and set the timer for 450 ms, then 675 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 1 minute, 20 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

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
  - Unidirectional link
  - Dirty link




```
RtB# 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
....
```


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
- This is normal behavior for older code
  - CSCdy20284 removed many of these neighbor resets
    - Implemented in 12.2S, 12.3T, and 12.4

Manual Changes

- Summary changes

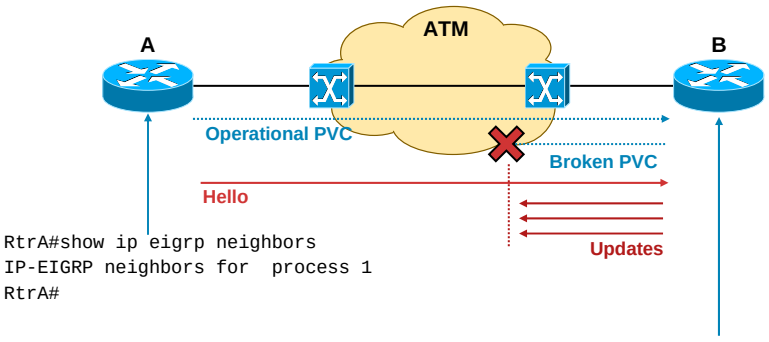
  When a summary changes on an interface, components of the summary may need to be removed from any neighbors reached through that interface; neighbors through that interface are reset to synch up topology entries

- 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

Unidirectional Links




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

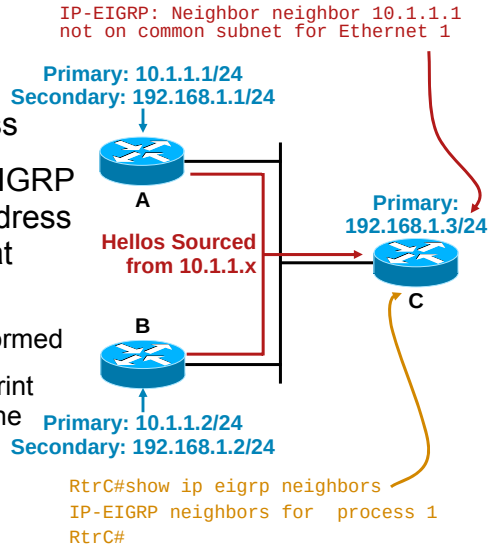
```
RtrB#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H Address      Interface Hold Uptime   SRTT RTO  Q   Seq
                         (sec)         (ms)      Cnt Num
1 10.1.102.2   Et0       14   00:00:15 0    5000 4   0
```

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
- The router on the left doesn't even realize that the router on the right exists. RtrA is sending out his hellos, waiting for a neighbor to show up on the network. What he 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

Primary/Secondary Mismatch

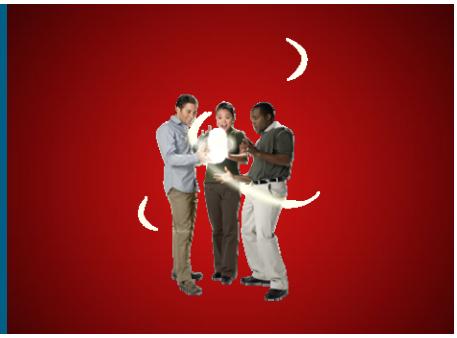
- EIGRP always sources packets from the primary interface address
- If a router receives an EIGRP packet with a source address not on the subnet for that interface:
  - The adjacency won't be formed
  - The receiving router will print log messages indicating the mismatch



Primary/Secondary Mismatch

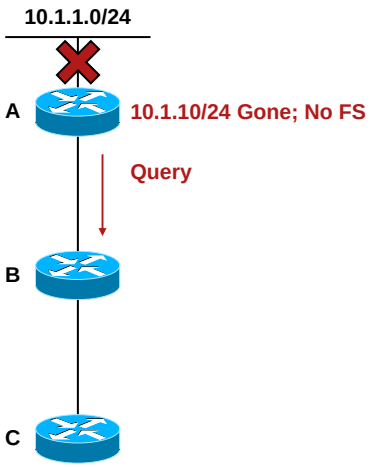
- While secondary addresses are not used as much as they were when routing protocols were not capable of VLSM (Variable Length Subnet Masking), they are still sometimes used
- It is very important to make sure that the primary IP addresses match; i.e., are part of the same subnet
- EIGRP will accept hellos that are sourced from an address that is a member of the secondary subnet; in the example above, rtrA and rtrB will accept the hello from rtrC, since the 192.168.1.3 address falls in the subnet covered by their secondary addresses
- If the source is from an address that doesn't exist on the interface, neighbors will not form; in the example above, the hellos from rtrA and rtrB will be sourced from 10.1.1.1 and 10.1.1.2, and when rtrC evaluates the received hello, it will find that the sources are not on its only subnet on that interface, and the hellos will be rejected

The Active Process



The Active Process

- What happens if there isn't a feasible successor when a router loses its path to a destination?
- A loses its route to 10.1.1.0/24
  - No FS, mark route active
  - Set a 1.5 minute active timer
  - Query all neighbors (B)

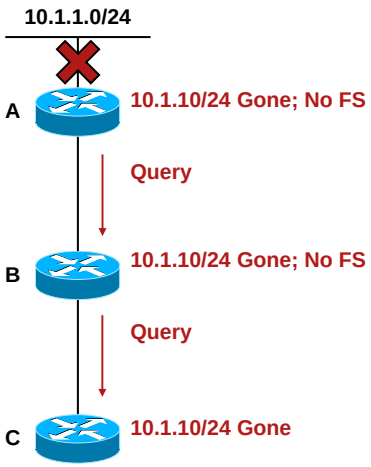


The Active Process

- The normal, converged state of routes in EIGRP is known as passive; when a route goes active, it means that EIGRP no longer has a known-good loop-free path for that destination and must actively look for a path to it
- This happens when the route goes down or increases in metric so that the best path is no longer feasible
  - Note: for a better definition of a feasible route, check out the topology table explanations in the "Troubleshooting Tools" section later in this presentation
- When a route goes active, EIGRP sends queries to all neighbors (possibly limited by split-horizon) and sets a 1.5 minute timer

The Active Process

- B receives A's query
  - No FS, mark route active
  - Set 1.5 minute active timer
  - Query all neighbors (C)
- C receives B's query
  - Examine local topology table
  - No feasible successors
  - No neighbors to query

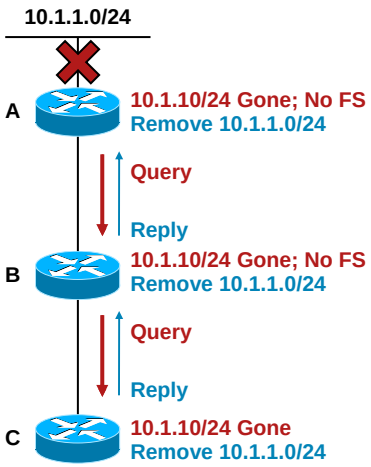


The Active Process

- When B receives the query from A, it looks in its topology table to see if it has a feasible path to that destination; in this example, it doesn't find another path to the destination, so it too goes active on the route
- B will then set its own 1.5 minute timer and send queries to all of its neighbors except the neighbor who it received the query from (its previous successor); in this case, C is the only neighbor to ask
- C receives the query from B and checks its own topology table to see if it has a feasible path to the destination; it doesn't have a feasible route and also doesn't have any other neighbors to ask
- Note that up to this point, B has not replied A, and C has not replied B

The Active Process

- C has no alternate path to 10.1.1.0/24
  - Remove from local tables
  - Reply to querying neighbors
- B receives C's reply
  - No outstanding queries
  - Remove from local tables
  - Reply to querying neighbors
- A receives B's reply
  - No outstanding queries
  - Remove from local tables

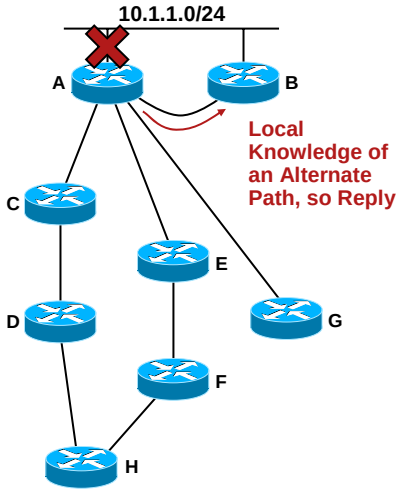


The Active Process

- Since C doesn't have any other neighbors to ask, it deletes the prefix that was active from both the topology table and routing table, then sends a reply to B notifying it that it doesn't have a path to the destination
- When B receives replies to all of its queries (in this case only from C), B will remove the prefix from its topology table and routing table and send a reply to A notifying it that it doesn't have a path to the destination
- When A receives replies to all of its queries (from B) and none of the replies contained an alternative path to the destination, it will remove the prefix from its topology table and routing table
- The network is now converged on the new information

The Active Process

- Where do queries stop?
- Router A loses its connection to 10.1.1.0/24
  - Router A sends B a query
- Router B examines its local tables and finds
  - Its current path (successor) doesn't pass through A
- Router B answers with a valid alternative path
- The query is bounded where there is local knowledge of another loop-free path

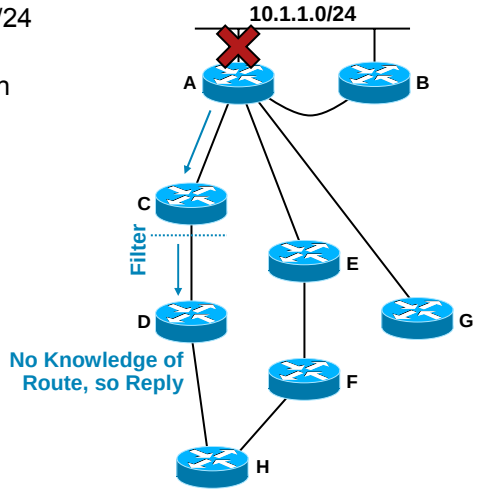


The Active Process

- Since queries are used to find an alternative path to a lost destination; where do these queries stop, or do they just keep circulating through the network?
- One place where a query will stop propagating is where the router receiving the query has an alternative path that it can reply with
- In the example above, B receives the query from A, looks in his topology table and finds that he has a loop free path to the destination
- B will reply to A with the metric he is using for that destination so that A can install the path through him to reach the destination
- There's no need to send the query on when you have a good answer yourself

The Active Process

- Router C is filtering 10.1.1.0/24 towards D
- Router A loses its connection to 10.1.1.0/24
  - Router A sends C a query
- Router C has no FS for 10.1.1.0/24
  - Router C sends D a query
- Router D examines its local tables
  - No information about 10.1.1.0/24, so send a reply
- Query is bounded because D has no information about 10.1.1.0/24

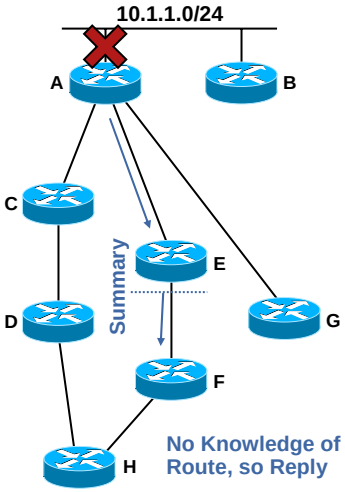


The Active Process

- Another place where a query will stop being propagated is where the router receiving the query doesn't know anything at all about the prefix
- In the example above, router C is using a distribution list to block the sending of 10.1.1.0/24 to router D; when C receives the query from A and doesn't have a feasible path, C will send a query to D
- D looks in his topology table and doesn't find any entries for 10.1.1.0/24 at all; because he doesn't know anything about the prefix, D immediately sends a reply with an infinity metric (unreachable) to C notifying him that the prefix is not reachable through him
- C can then reply to A that the prefix is not reachable through him, either, since the only reply he received declared the prefix unreachable

The Active Process

- Router E is summarizing towards F
- Router A loses its connection to 10.1.1.0/24
  - Router A sends E a query
- Router E has no FS for 10.1.1.0/24
  - Router E sends F a query
- Router F examines its local tables
  - No information about 10.1.1.0/24, so send a reply
- Query is bounded because F has no information about 10.1.1.0/24

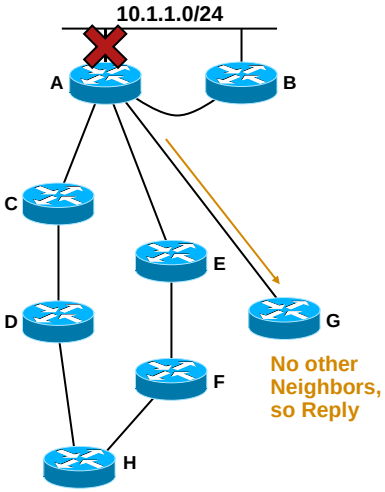


The Active Process

- As already mentioned, a query will stop being propagated when the router receiving the query doesn't know anything about the prefix
- In the example above, router E is using summarization to suppress sending 10.1.1.0/24 to router F; when E receives the query from A and doesn't have a feasible path, E will send a query to F
- F looks in his topology table and doesn't find any entries for 10.1.1.0/24 at all; because he doesn't know anything about the prefix, F immediately sends a reply with an infinity metric (unreachable) to E notifying him that the prefix is not reachable through him
- E can then reply to A that the prefix is not reachable through him either, since the only reply he received declared the prefix unreachable

The Active Process

- Router G has no other neighbors
- Router A loses its connection to 10.1.1.0/24
  - Router A sends G a query
- Router G examines its local tables
  - No FS
  - No other neighbors to query, so send a reply

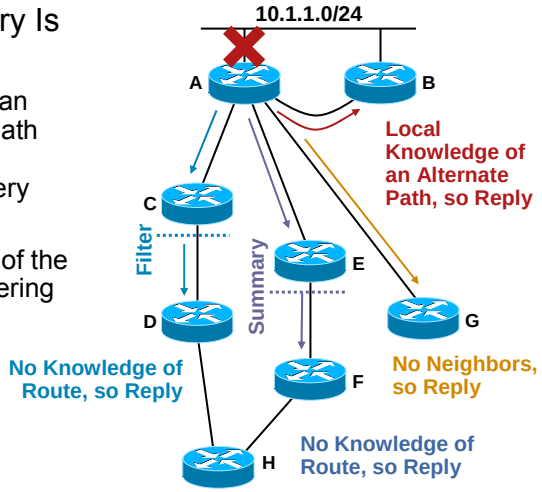


The Active Process

- A query will also stop being propagated when there are no other routers to ask
- In the example above, router A sends a query to router G; router G looks in his topology table and doesn't find an alternative path to the destination
- Since G doesn't have any other routers to send a query to, he will reply to A with a metric of infinity (unreachable)

The Active Process

- In summary, a Query Is Bounded by
  - Local knowledge of an alternate loop-free path not learned through the neighbor the query was received from
  - No local knowledge of the route because of filtering or summarization
  - No more neighbors to query



The Active Process

- So, in summary (no pun intended), a query will stop propagating when
  - A router receiving the query has a loop-free alternative path and replies with that value
  - A router receiving the query doesn't have the prefix in its topology table (no paths at all, much less alternative paths)
  - A router receiving the query has no other routers to ask
- The second point is the most important one in the proper design of a scalable/stable EIGRP network; use information hiding techniques (summarization and filtering) to minimize the query scope in your network

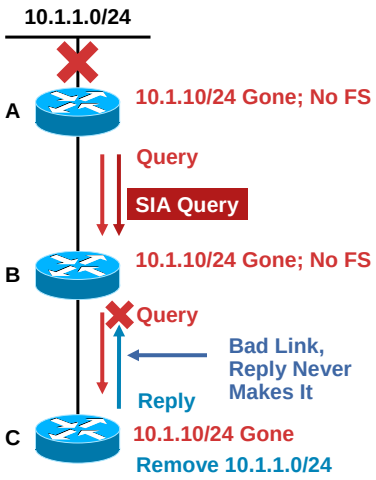
The SIA Query Process

- What if B doesn't reply to A within 1.5 minutes (the active timer)?
- A sends an SIA query

  If B responds to the SIA query with an SIA reply, A resets its timer, and the A/B neighbor relationship stays up
- B's relationship with C will fail at some point

  This clears the query from B's point of view

  B replies to A



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 B doesn't respond to A within 1.5 minutes because it's waiting for a reply from C, A will send an SIA-query to B checking the status. If B is still waiting on answer itself, it will respond to A with an SIA-reply. This resets the SIA timer on A so it will wait another 1.5 minutes
- Eventually, the problem keeping C from responding to B will take the neighbor relationship down between B and C, which will cause B to reply to A, ending the query process

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 still no reply by the end of this process, consider the route 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 we were advertising

Stuck-in-Active Routes (SIA)

```
%DUAL-3-SIA: Route 10.64.5.0 255.255.255.192
stuck-in-active state in IP-EIGRP 100. Cleaning up
```

Indicates at Least Two Problems

- A route went active
- It got stuck

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
  - Flapping link(s)?
  - From the same region of the network?
  - /32s from dial-in?
- 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 they /32 routes created from PPP? This is a common cause of routes going active as dial-in users connect and disconnect.
- 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

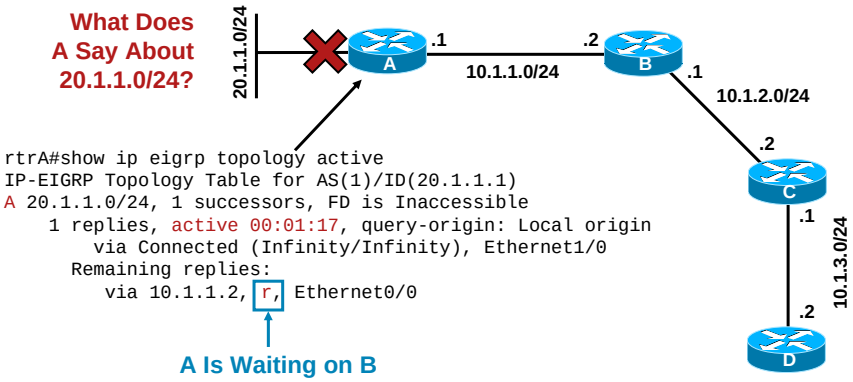
- Show ip eigrp topology active
- Useful only while the problem is occurring
- If problem isn't occurring at the time, it is difficult to find the source of routes getting stuck

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 probably 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 A Reporting SIA Routes? Let's Look at a Problem in Progress

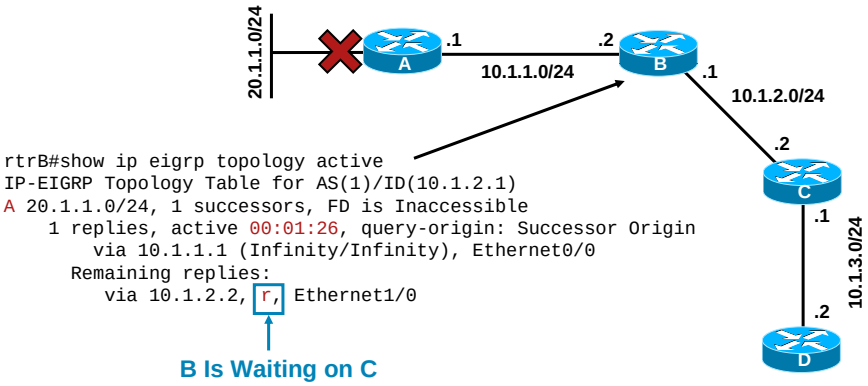


Chasing Active Routes

- 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 20.2.1.0/24 (note the A in the left column) and has been waiting for an answer from 10.1.1.2 (RTRB) for 1 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 (Step 1)

So Why Hasn't B Replied?

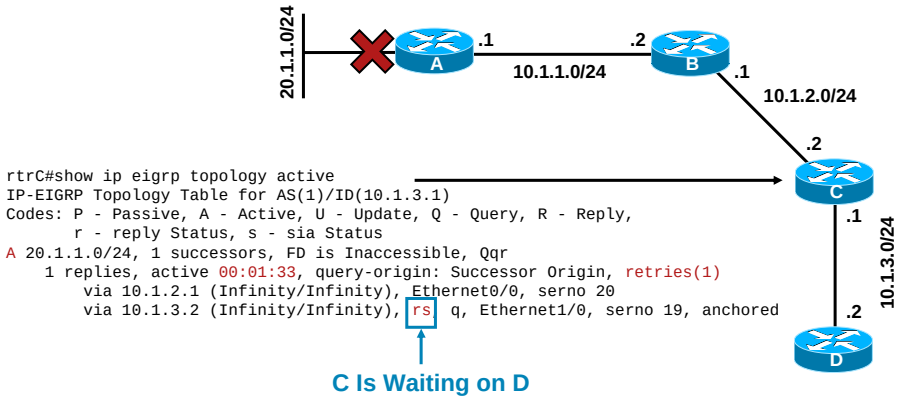


Chasing Active Routes (Step 1)

- 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 he is also waiting on another router downstream to answer his query before he can reply; again, the lower case r beside the IP address of 10.1.2.2 tells us he is the neighbor slow to reply
- We now need to go to 10.1.2.2 (RTRC) and see why he isn't answering RTRB

Chasing Active Routes (Step 2)

What's C's Problem?

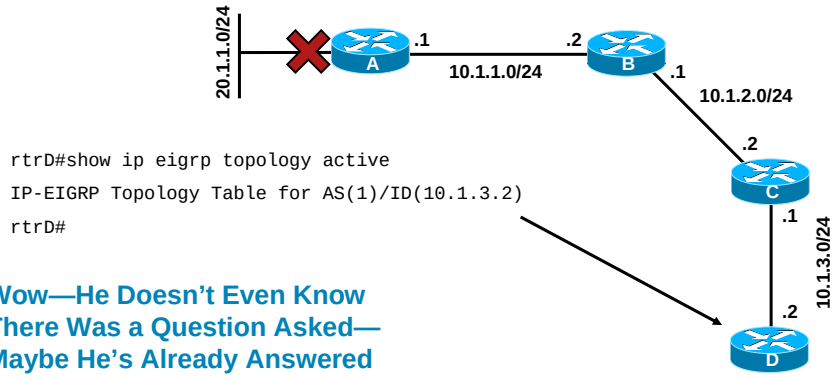


Chasing Active Routes (Step 2)

- On RTRC we repeat the show ip eigrp topology active command and see what he thinks of the route
- Again, he's waiting on another neighbor downstream to answer him before he 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 (Step 3)

Why Isn't D Answering?



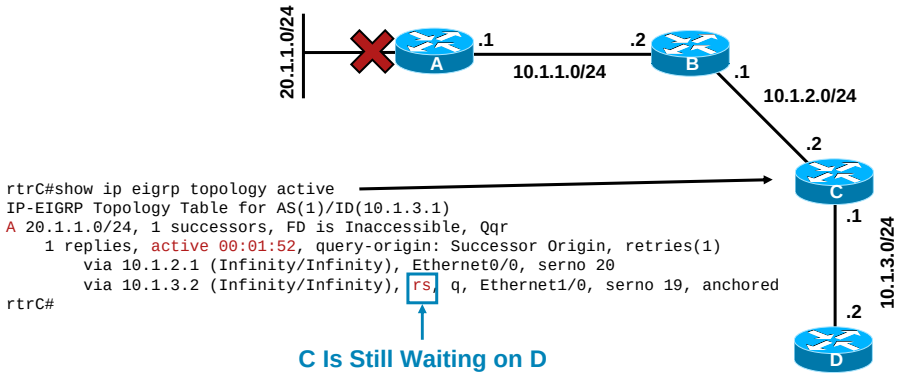
Wow—He Doesn't Even Know There Was a Question Asked—Maybe He's Already Answered

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 he is still active on the route

Chasing Active Routes (Step 4)

No; C Is Still Waiting on D; What's the Deal?

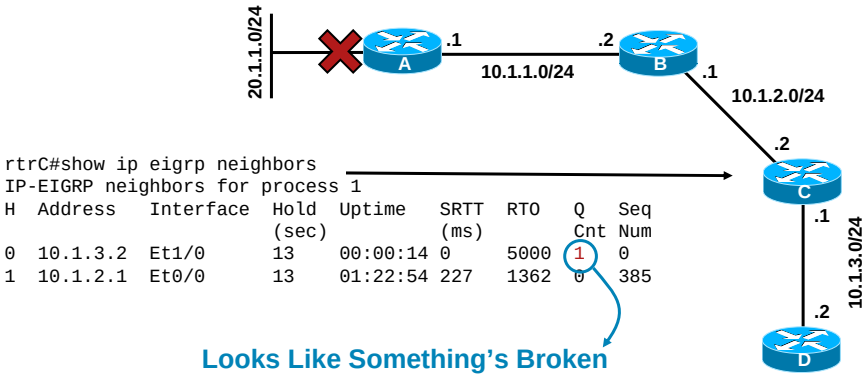


Chasing Active Routes (Step 4)

- Hmmm, RTRC still thinks the route is active and it's gotten even older
- There appears to be a problem, Houston. RTRC thinks he 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 (Step 5)

Let's 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
```

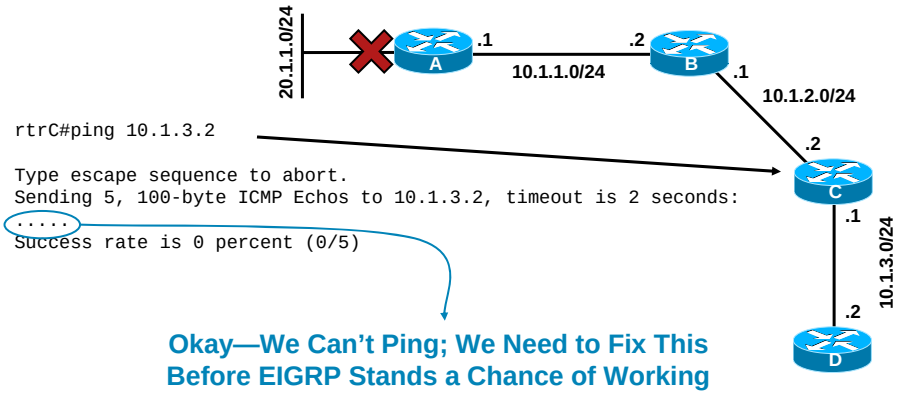
Looks Like Something's Broken Between Rtrc and Rtrd

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 (Step 6)

Let's Check General Reachability



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

  SIA should happen closer to the location of the cause of the problem

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)

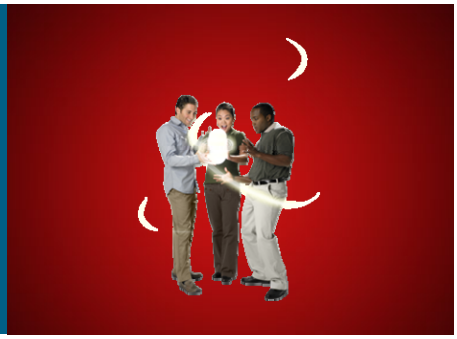
Minimizing SIA Routes

- Decrease query scope (involve fewer routers in the query process)
  - Summarization (manual or auto)
  - Distribute-lists
  - Define spoke/edge routers as stubs
- Run a Cisco IOS which includes CSCdp33034

Minimizing SIA Routes

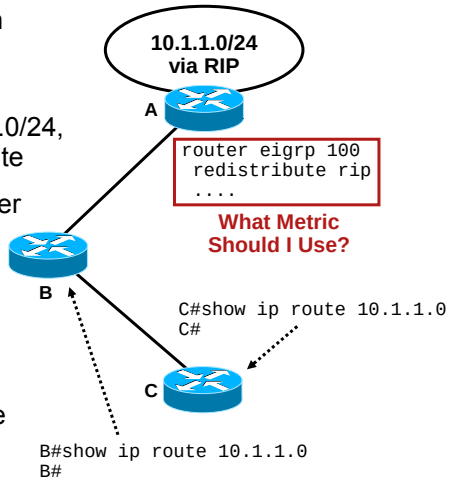
- We've now talked about the impart that SIA routes can cause on your network and how to track down the cause of SIA events; while you may not be able to completely rid your network of SIA routes, there are techniques you can use to minimize your exposure
- Decrease query scope—in our example network, you saw the queries sent to each router in a chain. As explained earlier, if a router received a query on a route that it doesn't have in its topology table, it immediately answers and doesn't send the query onward. This is a very good thing. You do this through:
  - Summarization—auto-summary (seldom used) or manual summary to summarize within a major network or to summarize external routes
  - Distribute-lists—used to limit knowledge of routes; particularly on dual-homed remotes, which tend to reflect all routes back to the other leg of the dual home connection
  - Use hierarchy—if the network doesn't have hierarchy, the two techniques above cannot adequately be used
  - Define spoke/edge routers as stubs so they aren't queried at all
  - Run a Cisco IOS with CSCdp33034 included; It changes the timing to decrease SIAs and push them closer to the source of the problem if they do occur

External Routes



External Routes

- The most common problem with external routes is EIGRP not installing them on some routers
- Router A is redistributing 10.1.1.0/24, but B and C do not have the route
- The first thing to check is whether A has a redistribution metric configured via either
  - Default-metric <metric>
  - Redistribute rip metric <metric>
- EIGRP can't pull metrics out of thin air, so it won't redistribute routes unless it knows what metric to assign to them

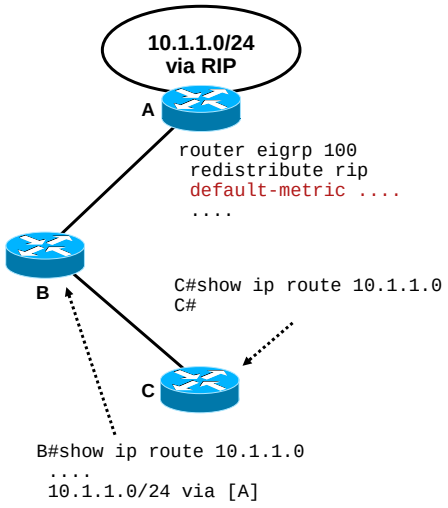


External Routes

- EIGRP will automatically learn the redistribution metrics from:
  - A connected interface for redistribute connected
  - The interface through which a static route is reached for redistribute static (note: this isn't always reliable. You're better off still specifying the redistribution metric!)
  - The metric of an IGRP route in the same AS
  - The metric of an EIGRP route from another AS
- If none are those are true, you **must** supply the metric for redistribution

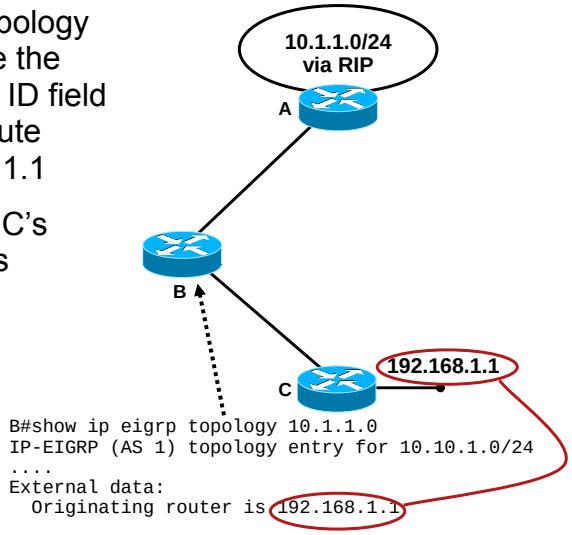
External Routes

- We've resolved the redistribution metric issue on A, and B is now receiving the route, but C isn't—why?



External Routes

- Looking at B's topology table, we can see the originating router ID field in the external route is set to 192.168.1.1
- But, that's router C's loopback address



External Routes

- In the example above, we have a problem where routes are being redistributed just fine, but one router elsewhere in the network is refusing to install the external routes into its topology table or routing table
- As the slide shows, the problem is that the router-id of the redistributing router matches the router-id of the router refusing to install the route
- To block routing loops, a router doing redistribution will not accept a route from a neighbor if he is the one that originated it; this is known by the originating router field in the external data section of the topology table entry
- Since the router-ids are the same on router A and router C, router C thinks router A's external routes originated on router C and he rejects them

External Routes

- In older versions of Cisco IOS Software, the only way to find out a router's router ID was to go to an adjacent router and look for some redistributed route from that router
- If there isn't anything being redistributed, redistribute something

```
router# show ip eigrp topology 10.1.1.0 255.255.255.0
IP-EIGRP (AS 7): topology entry for 10.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2560000256
  Routing Descriptor Blocks:
  10.1.2.1 (Ethernet0/0), via 10.1.2.1, Send flag is 0x0
      Composite metric is (2560000256/0), Route is External
      ....
      External data:
        Originating router is 192.168.1.1
        AS number of route is 0
        External protocol is RIP, external metric is 1
        Administrator tag is 0 (0x00000000)
```


External Routes

- In newer versions of Cisco IOS Software, a router's router ID is listed in the output of show ip eigrp topology

```
router-1# show ip eigrp topology
IP-EIGRP Topology Table for AS(7)/ID(192.168.1.1)
....
```

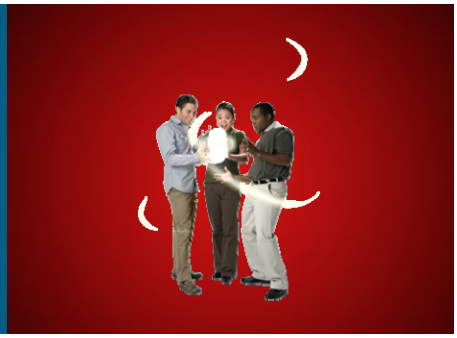
- If your event log is big enough, or things are slow enough, you might see the problem indicated in your even log

```
1    02:30:18.591 Ignored route, metric: 192.168.1.0 2297856
2    02:30:18.591 Ignored route, neighbor info: 10.1.1.0/24 Serial0/3
3    02:30:18.591 Ignored route, dup router: 192.168.1.1
```


External Routes

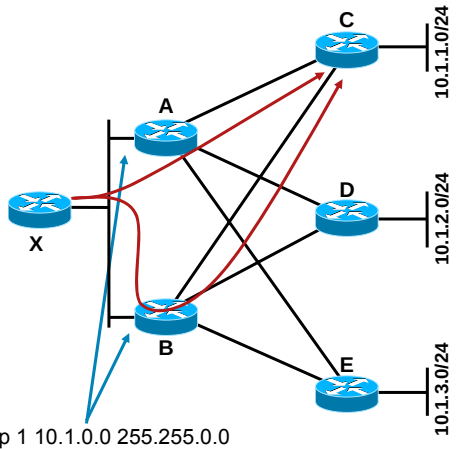
- The EIGRP router ID is derived from
  - The router-id command
  - Lowest numbered loopback interface
  - Highest local interface IP address if no loopbacks
- Once the router ID is set, it won't be changed without manual intervention, even if the interface from which it's taken is removed from the router
- Impact on external routes
  - EIGRP includes the router ID of the originating router in external routing information
  - If a router receives an external route with a router ID matching its own local router ID, it discards the route
  - This prevents routing loops in externally derived routes

Aggregation



Black Hole Summary Routes

- This network implements manual summarization from the distribution routers toward the core
- It normally doesn't matter whether A or B is used to reach a remote from X



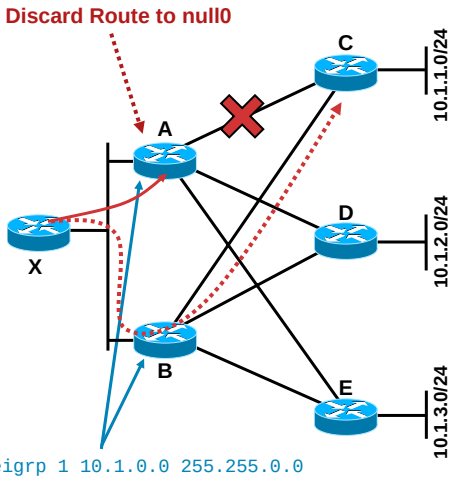
ip summary-address eigrp 1 10.1.0.0 255.255.0.0

Black Hole Summary Routes

- In this example network, the designer implemented manual summarization to hide the specific routes located at the remote sites by summarizing toward the core. On each of the interfaces of router A and router B toward router X is the summary statement ip summary-address eigrp 1 10.1.0.0 255.255.0.0. This blocks the specific prefixes from being advertised to X, and instead only installs the 10.1.0.0/16 prefix there.
- Normally, this works great. Minimal info is known at the core and proper routing takes place just fine. But what happens if a problem occurs?

Black Hole Summary Routes

- What happens if the A to C link fails?
  - X is still receiving the summary from both A and B and may chose A as its path for packets going to 10.1.1.0/24
  - A builds a discard route to null0 with an administrative distance of 5 when the summary is configured
  - The traffic will be dropped at A!




```
ip summary-address eigrp 1 10.1.0.0 255.255.0.0
```

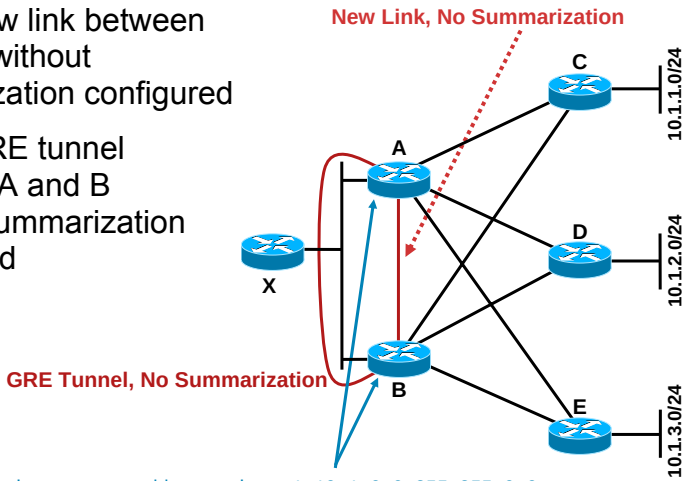

Black Hole Summary Routes

- Another problem we sometimes encounter when network designers listen to our recommendations is with summary black holes. The problem is that a summary will be issued if any component of the summary exists. If a router doing summarization loses access to one of the components of the summary, it will still advertise reachability to the entire summary, even though packets destined to the missing component cannot be delivered.
- This isn't a problem if the summarizing router is the only path the lost network, but often it's not. In the diagram above, both rtrA and rtrB have access to the remotes and are summarizing them toward the core of the network. If rtrA loses access to 10.1.1.0/24, routers downstream (like rtrX) could send packets to rtrA that he cannot deliver. If the packets went to rtrB, however, they would have been delivered successfully.

Black Hole Summary Routes

Possible Solutions

- Add a new link between A and B without summarization configured
- Add a GRE tunnel between A and B without summarization configured




```
ip summary-address eigrp 1 10.1.0.0 255.255.0.0
```


Black Hole Summary Routes

- The normal method to avoid/resolve this problem is to have another link between summarizing routers (another Fast/Gig Ethernet, PVC, etc.) and not summarize across this link; in that way, rtrA would be getting component routes from rtrB and would know how to deliver packets to 10.1.1.0 through rtrB
- Another approach used if the cost of another link is too high is to put a GRE tunnel between rtrA and rtrB and allow all component routes to be advertised across this tunnel
- There's also some discussion inside of EIGRP development of ways to solve this problem dynamically. A sys-wish bug has been filed against it (CSCdw68502) but we haven't started the work yet. We're still discussing the best way to solve it.

Troubleshooting Tools

- The event log
- Debugs
- Show commands
- Cisco's technical support web site

EIGRP Troubleshooting Tools

- Debugs Versus the EIGRP Event Log
- On a busy, unstable network debugs can be hazardous to your network's health
- Event log is non-disruptive—already running

EIGRP Troubleshooting Tools

- Two 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 much safer; just display it and see what's been happening lately

Event Log

- Always running (unless manually disabled)
- Default 500 lines (configurable)

```
EIGRP event-log-size <number of lines>
```

- Most recent events at top of log

  Read bottom to top

Event Log

- Separate event log is kept for each AS
- 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 ##`

  ## is the number of lines

  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 at the top of the log, so time flows from bottom to top

Event Log

- Three Different Event Types Can Be Logged

  `EIGRP log-event-type [dual][xmit][transport]`
- Default is dual—normally most useful
- Any combination of the three can be on at the same time
- Work in progress to add additional debug info to event log

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
```


Debugs

- Remember—debugs can be dangerous
  - Use only in the lab or if advised by the TAC
- To make a little safer
  - Logging buffered <size>
  - No logging console

Debugs

- By enabling logging buffered and shutting off the console log, you improve your odds of not killing your router when you do a debug; still no guarantees
- We often change the scheduler interval when we do debugs in the TAC, as well; this command is version dependent, so I'm not going to give you syntax here

Debugs

- Use Modifiers to Limit Scope of Route Events or Packet Debugs
- Limit to a particular neighbor

```
debug ip eigrp neighbor AS address
```

- Limit to a particular route

```
debug ip eigrp AS network mask
```


Debugs

- Both packet debugs and route event debugs create so much output that you would have a hard time sorting through it for the pieces you care about; the two modifier commands above allow you to limit what the debug output will show
- Debug ip eigrp AS <network><mask> will limit the output to only those entries that pertain to the route identified
- Debug ip eigrp neighbor AS address will limit the output to those entries pertaining to a particular neighbor
- Unfortunately, you have to enable the debug (packet or route events) prior to putting the modifier on, so you could kill your router before you are able to get the limits placed on the output; sorry, but that's the way it is
- Examples of route events with modifiers will be in the next few slides

Debugs

Debug IP EIGRP Route Events

```
RTRA#debug ip eigrp
IP-EIGRP Route Events debugging is on
RTRA#debug ip eigrp neighbor 1 10.1.6.2
IP Neighbor target enabled on AS 1 for 10.1.6.2
IP-EIGRP Neighbor Target Events debugging is on
RTRA#clear ip eigrp neighbor
RTRA#
IP-EIGRP: 10.1.8.0/24 - do advertise out Serial1/2
IP-EIGRP: Int 10.1.8.0/24 metric 28160 - 256002560
IP-EIGRP: 10.1.7.0/24 - do advertise out Serial1/2
IP-EIGRP: 10.1.1.0/24 - do advertise out Serial1/2
IP-EIGRP: Int 10.1.1.0/24 metric 28160 - 25600256
IP-EIGRP: Processing incoming UPDATE packet
IP-EIGRP: 10.1.6.0/24 - do advertise out Serial1/1
```


Debugs

Debug IP EIGRP Route Events

- In this debug, we are looking at route events recorded when neighbors are cleared (in reality, the debugs produced were far, far more. This is only a snapshot of the debug). A modifier was included to limit the output to only the events related to a single EIGRP neighbor, 10.1.6.2.
- Notice that the debug output doesn't identify which neighbors are involved in any of the events; without knowing the address used in the modifier command, you really can't tell which neighbors you are interacting with in the debug output
- This output is often useful when trying to determine what EIGRP thinks is happening when there are route changes in the network

Debugs

Debug IP EIGRP Route Events

```
RTRA#debug ip eigrp
IP-EIGRP Route Events debugging is on
RTRA#debug ip eigrp 1 10.1.7.0 255.255.255.0
IP Target enabled on AS 1 for 10.1.7.0/24
IP-EIGRP AS Target Events debugging is on
RTRA#clear ip eigrp neighbor
IP-EIGRP: 10.1.7.0/24 - do advertise out Serial1/2
IP-EIGRP: 10.1.7.0/24 - do advertise out Serial1/1
IP-EIGRP: Int 10.1.7.0/24 metric 20512000 20000000 512000
IP-EIGRP: 10.1.7.0/24 - do advertise out Serial1/2
IP-EIGRP: Processing incoming UPDATE packet
IP-EIGRP: 10.1.7.0/24 - do advertise out Serial1/1
```


Debugs

Debug IP EIGRP Route Events

- Again, this is the output of debugging routing events, this time modified to only display output related to a single route in the network. This modifier can be very useful when trying to troubleshoot a single route (or representative route)

Debugs

Debug EIGRP Packet

```
RTRA#debug eigrp packet ?
  ack          EIGRP ack packets
  hello        EIGRP hello packets
  ipxsap       EIGRP ipxsap packets
  probe        EIGRP probe packets
  query        EIGRP query packets
  reply        EIGRP reply packets
  request      EIGRP request packets
  stub         EIGRP stub packets
  retry        EIGRP retransmissions
  terse        Display all EIGRP packets except Hellos
  update       EIGRP update packets
  verbose      Display all EIGRP packet
```


Debugs

Debug EIGRP Packet

- This debug is used in a variety of problems and circumstances; Debug eigrp packet hello is used to troubleshoot neighbor establishment/maintenance problems
- Debug eigrp packet query, reply, update, etc., are also often used to try to determine the process occurring when a problem occurs. Be careful. I've crashed/hung more than one router by doing a debug on a router that was too busy.
- Probably the most commonly used debug eigrp packet option is terse, which includes all of the above except hellos; an example follows on the next page

Debugs

Debug EIGRP Packet Terse

```
RtrA#debug eigrp packet terse
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, IPXSAP, PROBE, ACK, STUB)
EIGRP: Sending UPDATE on Serial1/0 nbr 10.1.1.2
   AS 1, Flags 0x0, Seq 2831/1329 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno 19707-19707
EIGRP: Sending UPDATE on Serial1/1 nbr 10.1.2.2
   AS 1, Flags 0x0, Seq 2832/1708 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno 19707-19707
EIGRP: Sending UPDATE on Serial1/2 nbr 10.1.3.2
   AS 1, Flags 0x0, Seq 2833/1680 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno 19707-19707
EIGRP: Received ACK on Serial1/0 nbr 10.1.1.2
   AS 1, Flags 0x0, Seq 0/2831 idbQ 0/0 iidbQ un/rly 0/0 peerQ un/rely 0/1
EIGRP: Serial1/0 multicast flow blocking cleared
EIGRP: Received ACK on Serial1/1 nbr 10.1.2.2
  AS 1, Flags 0x0, Seq 0/2832 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1
EIGRP: Serial1/1 multicast flow blocking cleared
```


Debugs

Debug IP EIGRP Notifications

- Debug ip eigrp notifications is used to troubleshoot problems with redistribution into EIGRP; the callbacks describe what is happening between EIGRP and the routing table as routes are redistributed

```
RtrA#debug ip eigrp notifications
IP-EIGRP Event notification debugging is on
RtrA#clear ip route *
RtrA#
IP-EIGRP: Callback: reload_iptable
IP-EIGRP: iptable_redistribute into eigrp AS 1
IP-EIGRP: Callback: redist frm static AS 0 100.100.100.0/24
           into: eigrp AS 1  event: 1
IP-EIGRP: Callback: redist frm static AS 0 200.200.200.0/24
           into: eigrp AS 1  event: 1
```


Debugs

Debug EIGRP FSM (Finite State Machine)

```
RTRA#debug eigrp fsm
EIGRP FSM Events/Actions debugging is on
RTRA#clear ip route *
RTRA#
DUAL: Find FS for dest 10.1.8.0/24. FD is 28160, RD is 28160
DUAL:     0.0.0.0 metric 28160/0 found Dmin is 28160
DUAL: Find FS for dest 10.1.3.0/24. FD is 21024000, RD is 21024000
DUAL:     10.1.6.2 metric 21024000/2169856 found Dmin is 21024000
DUAL: RT installed 10.1.3.0/24 via 10.1.6.2
DUAL: Find FS for dest 10.1.2.0/24. FD is 21536000, RD is 21536000
```

Debugs

Debug EIGRP FSM

- Debug eigrp fsm is very, very similar to dual event log; since the dual event log is non-disruptive and this debug could certainly cause problems, I rarely use this debug
- FSM stands for Finite State Machine, which describes the behavior of DUAL, the path selection part of EIGRP

Topology Table

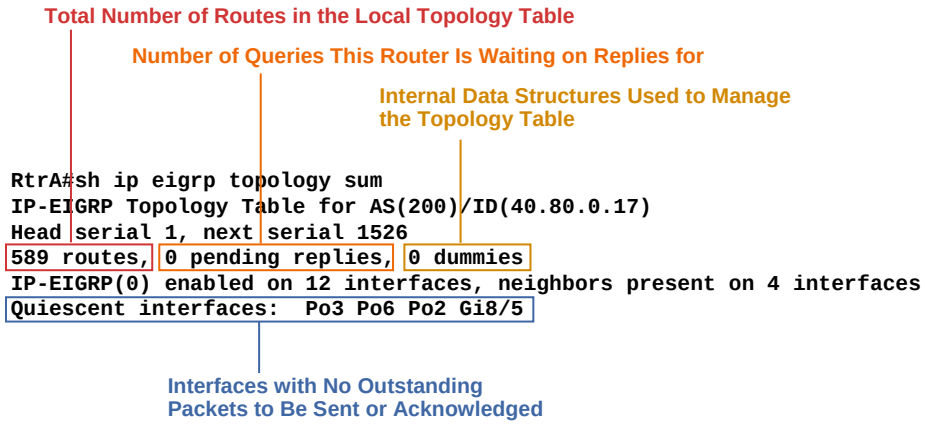
- The topology table is probably the most critical structure in EIGRP
  - Contains building blocks used by DUAL
  - Used to create updates for neighbors/populate routing table
- Understanding the topology table contents is extremely important in troubleshooting EIGRP problems

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.
- 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

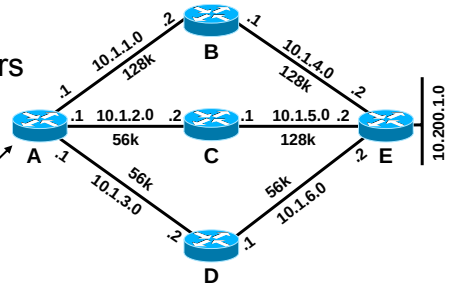
Show IP EIGRP Topology Summary



Topology Table

Show IP EIGRP Topology

- 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

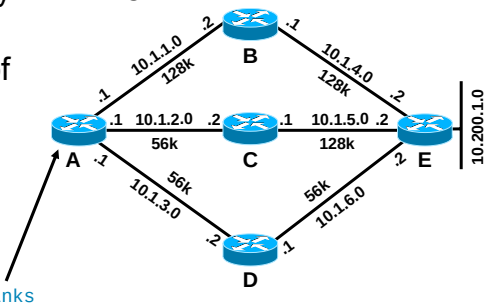
Show IP EIGRP Topology

- 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 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.
- 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.
- As you can see, 10.1.2.2 is a feasible successor because his reported distance (21514560) is less than our current feasible distance (21026560). (remember the feasibility condition?)

Topology Table

Show IP EIGRP Topology All-Links

- 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
```

Successor
Feasible Successor
Possible Successor

RD

Topology Table

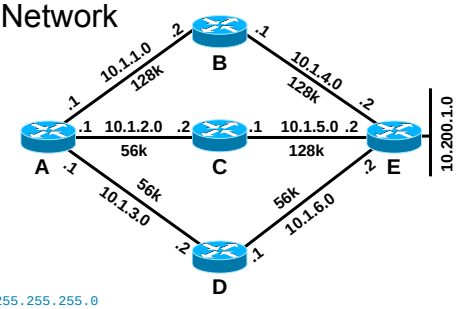
Show IP EIGRP Topology All-Links

- If you want to display all of the information 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 he 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 Network

- Show ip eigrp topology <network><mask> displays detailed information for all paths received for that destination




```
RtrA#show ip eigrp topology 10.200.1.0 255.255.255.0
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

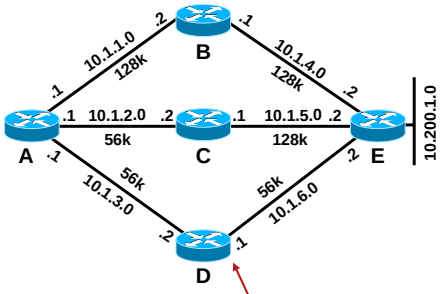
Show IP EIGRP Topology Network

- 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>
- 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

Show IP EIGRP Topology Network

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




```
RtrA#show ip eigrp topology 30.1.1.0 255.255.255.0
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)
```

Static Route to 30.1.1.0/24 Is Redistributed into EIGRP

Topology Table

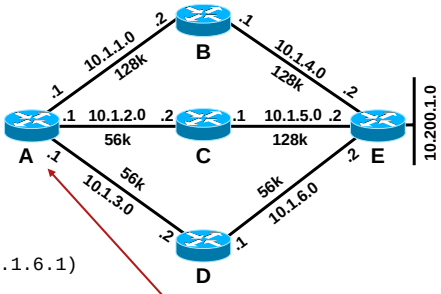
Show IP EIGRP Topology Network

- 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 loopback interfaces, 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, he 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
```

Static Route to 10.200.1.0/24 Through RtrB

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

Other Show Commands

Show IP EIGRP Neighbor Detail

- Show ip eigrp neighbor detail is the big brother of the command shown earlier show ip eigrp neighbor; 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
```


Other Show Commands

Show IP EIGRP 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
```

Other Show Commands

Show IP EIGRP 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

Other Show Commands

Show IP EIGRP Traffic

```
rtrB#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 1
  Hellos sent/received: 574/558
  Updates sent/received: 5/7
  Queries sent/received: 2/2
  Replies sent/received: 2/2
  Acks sent/received: 11/7
  Input queue high water mark 2, 0 drops
  SIA-Queries sent/received: 1/1
  SIA-Replies sent/received: 1/1
  Hello Process ID: 64
  PDM Process ID: 63
```


Other Show Commands

Show IP EIGRP Traffic

- Show ip eigrp traffic can be very useful to see what kind of activity has been occurring on your network; some of the most interesting information include:

  Input queue high water mark—this shows how many packets have been queued inside of the router to be processed. When packets are received from the IP layer, EIGRP accepts the packets and queues them up for processing. If the router is so busy that the queue isn't getting serviced, the queue could build up. Unless there are drops, there is nothing to worry about, but it can give you an indication of how hard EIGRP is working

  SIA-queries sent/received—this is useful to determine how often the router has stayed active for at least 1 1/2 minutes (as mentioned in the earlier section on stuck-in-active routes. This number should be relatively low. If it's not, it's taking a bit of time for replies to be received for queries, and it might be worth exploring why

Other Show Commands

Show IP Protocol

```
rtrA#show ip protocol
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 200"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 200
  EIGRP NSF-aware route hold timer is 240s
  EIGRP NSF enabled
     NSF signal timer is 20s
     NSF converge timer is 120s
```

Other Show Commands

Show IP Protocol

- There are many fields in the show ip protocol which are useful in the troubleshooting process
- Some of the most interesting include:

  Outgoing and incoming filter lists

  Variance setting

  Redistribution configured (note that if a router is not redistributing other protocols, it still shows that it is redistributing itself)

  NSF configuration and timers

Other Show Commands

Show IP Protocol

```
Automatic network summarization is not in effect
Address Summarization:
  40.0.0.0/8 for Vlan301
    Summarizing with metric 1536
Maximum path: 4
Routing for Networks:
  40.80.0.0/16
  192.168.107.0
Routing Information Sources:
  Gateway         Distance      Last Update
  (this router)         90      00:01:10
  40.80.24.33           90      01:13:47
  40.80.12.19           90      01:13:47
  40.80.23.31           90      01:13:48
  40.80.8.13            90      01:13:48
Distance: internal 90 external 170
```

Other Show Commands

Show IP Protocol

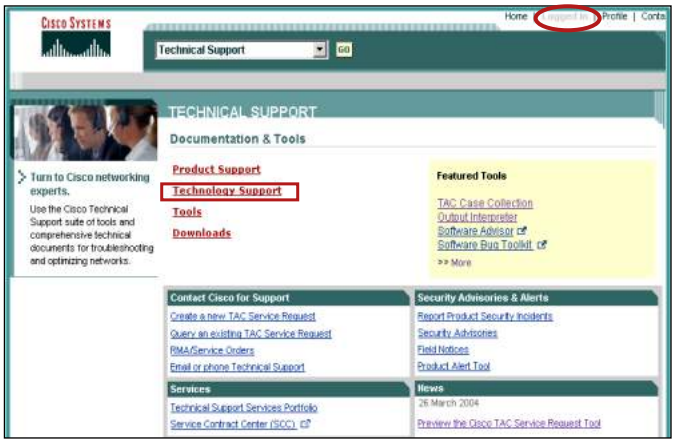
- More show ip protocol stuff
  - Summarization defined (both auto and manual) along with the metric associated with each summary
  - Max-path setting
  - Network statements
  - Distance settings
- Note that the Routing Information Sources section is really useless for EIGRP, since it doesn't use periodic update; we didn't rip it out of the display, but there isn't much useful information for EIGRP in this section

EIGRP Troubleshooting Summary

- Most problems seen in EIGRP networks are caused by factors outside of EIGRP itself (congestion, lack of summarization, etc.)
- There are many tools and techniques available for troubleshooting problems in EIGRP networks

Cisco Technical Support Website

http://www.cisco.com/techsupport

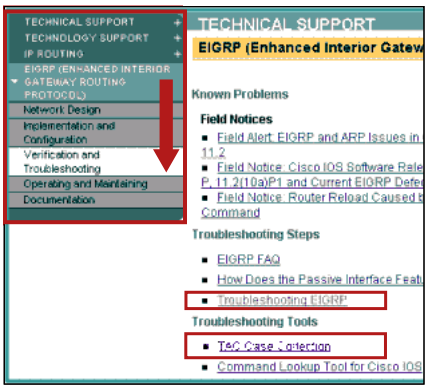


Cisco Technical Support Web Site

- This is the Cisco Technical Support Website. To access the range of useful support tools and documentation on the site at: www.cisco.com/techsupport, please be sure to log in with your registered cisco.com user name and password. If you are not a registered user and would like information about what is needed to register at Cisco.com (e.g., a service contract), please click the "Register" tab at the top of the page.
- The Technical Support site is divided into modules. The first two modules, Product Support and Technology Support, under Documentation and Tools, contain all the technical content on the Cisco Technical Support Website. These two modules provide an intuitive and easy way to navigate to all the technical documents and tools on the site.
- Also in this screen capture are additional modules that provide fast access to specific types of information on the site; for example, the "Tools" section links to useful online support tools, and in the upper right of the home page, we now have links to some of our top tools under "Featured Tools"
- You'll also see on the home page a link to software downloads called "Downloads" and a section to "Contact Cisco for Support" that include links to the TAC Service Request Tool—to create a new service request or to query an existing service request
- In addition, under "Communities and Training," you'll find a link, "Technical Support Training Resources" for training information
- I'd also like to point out a new section, "Security and Advisories and Alerts" that contains links to security advisories, field notices and so on
- We're interested today specifically in troubleshooting EIGRP, so we'd click the link for Technology Support → IP Routing → EIGRP → Verification and Troubleshooting section (for EIGRP)—as we'll see in the next slide

Troubleshooting Support for EIGRP

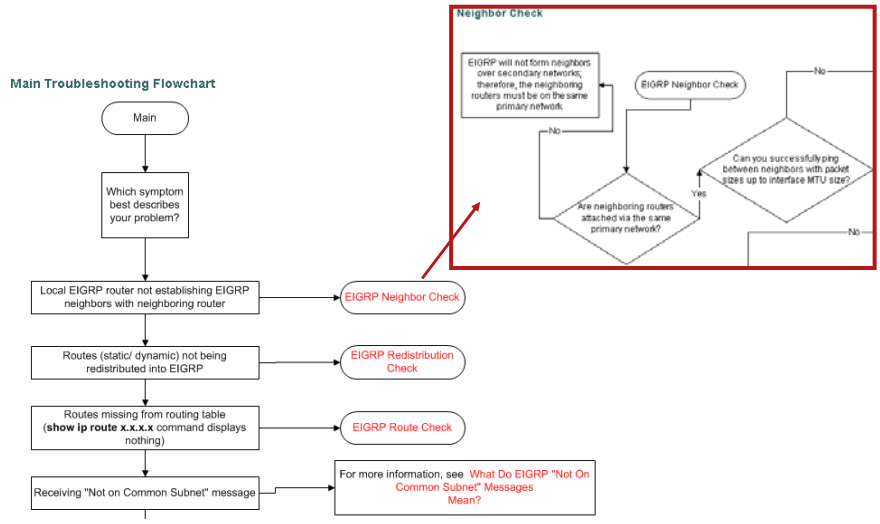
- Known problems documented as field notices
- Troubleshooting steps
- Troubleshooting tools



Troubleshooting Support for EIGRP

- Here we are in the “Verification and Troubleshooting” section for EIGRP under IP Routing
- The troubleshooting section contains a sub-section for known problems such as field notices. There are also sub-sections for troubleshooting steps and tools. We’re going to take a quick look at a document under troubleshooting steps, “Troubleshooting EIGRP,” and at one of our useful troubleshooting tools, TAC Case Collection, using an EIGRP example.

Troubleshooting EIGRP Documentation

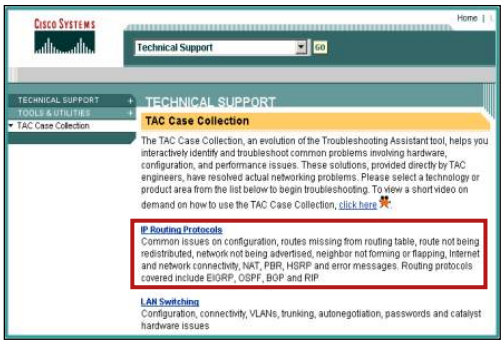


Troubleshooting EIGRP Documentation

- Having clicked the link for the “Troubleshooting EIGRP” document shown in the previous slide, we arrive here, where we have a document that outlines steps of a flowchart for troubleshooting common issues with EIGRP. The document starts off by asking you what symptom best describes your problem. For example, you may respond that the local EIGRP router is not establishing EIGRP neighbors with the neighboring router, and then, click the command box in red to be taken to the next flowchart, as we see in this slide, and so on.
- By the way, this is one of our interactive documents that offers customized analysis of your Cisco device, so if you have the output of a show interface serial, show ip eigrp neighbor, show tech-support, or a show ip eigrp topology x.x.x.x command from your Cisco device, you can use the Output Interpreter tool to display potential issues and fixes. To use the Output Interpreter, you must be a registered customer, be logged in, and have JavaScript enabled
- And speaking of useful technical support website tools, why don’t we take a look at one of our most popular ones— the TAC Case Collection—in order to troubleshoot an EIGRP issue

TAC Case Collection

- Free text query
- Guided search
- Search by solution number
- View all solutions
- Suggests solutions that come from actual service requests taken by TAC engineers

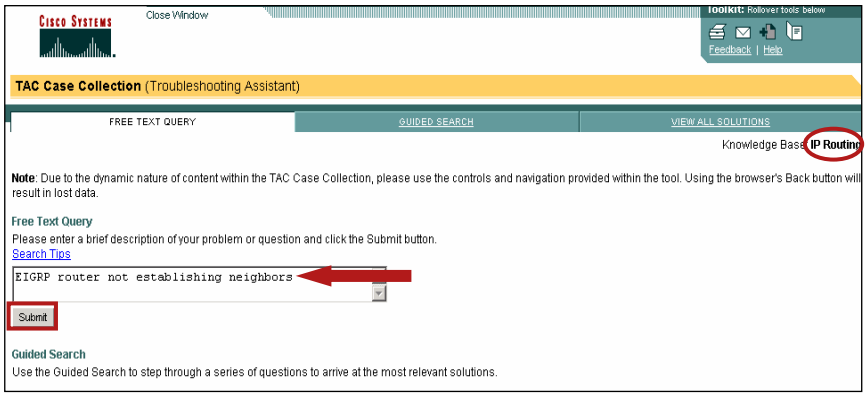


TAC Case Collection

- The TAC Case Collection is an evolution of the troubleshooting assistant tool and has been enhanced to include new knowledge bases, including for IP routing protocols; we'll be looking at an EIGRP example today
- The tool enables you to interactively identify and troubleshoot common problems involving hardware, configuration, and performance issues. These solutions, provided directly by TAC engineers, have resolved actual networking problems. You'd select a technology or product area in which you are experiencing a problem or have a question in order to begin the troubleshooting process
- If you would like to view a short video on demand about how to use the TAC Case Collection, please click the link on the tool's launch page
- This tool offers four different methods of searching for solutions:
- Free text query—allows you to interact with the application through natural language. This method will be the focus of our coverage of the tool today
- Guided search—allows you to perform a step-by-step search by answering a series of questions that lead to the most relevant solutions based on the symptoms selected
- Search by solution #—in instances where the solution number is known, for example, if you have been provided the URL to the tool along with a solution number by a TAC engineer, you'd simply type the solution number in the free text field and be provided directly with the details of that solution; these solution numbers begin with the letter K followed by a series of numeric digits
- View all solutions—will link you to a list of all the solution titles in the solution base, hyperlinked to their respective solution details page, and categorized by problem type
- So, guided search, search by solution # and view all solutions would yield the same solutions via those other routes as via free text query; I invite you to check out these three methods on your own
- A quick note on the tool: due to the dynamic nature of the application, the supported browser versions are IE 5.5 and above or Netscape 6.x. Also, please note that you must use the controls and navigation provided within the tool. If you use your browser's navigation controls (for example, the back button), search data may be lost

TAC Case Collection

Free Text Query

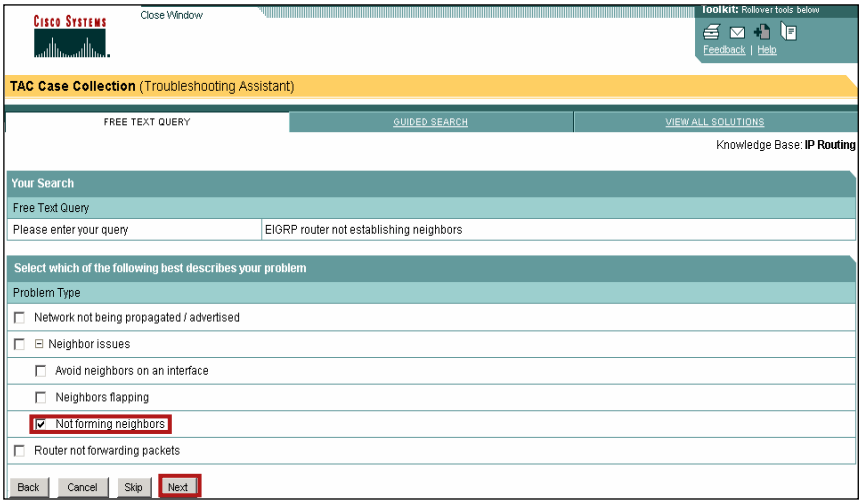


TAC Case Collection

- With that background, let's look at how to use the tool. Clicking the IP Routing Protocols link from the previous jump page, brings me here to a new child window. In the upper right of the page, you'd see the knowledge base selected—in our example, IP Routing
- First, let's focus our attention on how to perform a free text query highlighted in the top white tab. You will notice that Guided Search and View All solutions are the next tabs highlighted in green. So basically, the free text field is where you'd type in a description of the problem or question: "EIGRP router not establishing neighbors," and then click the "Submit" button to begin the search. The application then parses the information in the query and in some instances, will prompt you for further information via refining questions. The tool continues to ask questions until the search results are narrowed to three or fewer at 100% similarity and then will present a search results table—as we work through the tool to get to our solutions.
- I'd like to point out that while you can continue to narrow your search by clicking the "next" button, you can also view search results at anytime by selecting the "submit" button
- Also, for a free text query, the tool will switch to Guided Search if additional information is required to narrow the search or if the tool did not understand the entry. Otherwise, if your query is understood by the tool from the outset, you could end up saving time because you will have started from a later stage in the search process as you work toward reaching the solutions.
- So, let's run the search for our example now

TAC Case Collection

Free Text Query (Cont.)—Problem Type

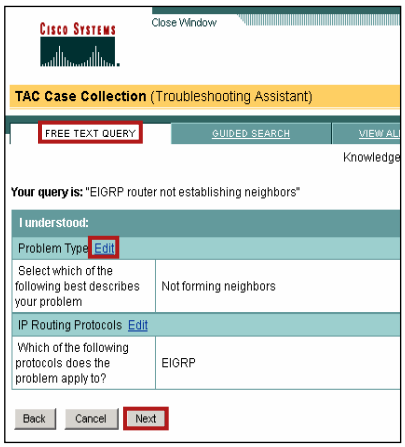


TAC Case Collection

- The tool understands the query correctly “EIGRP Router not establishing neighbors” and selects “not forming neighbors” as the problem type; we click “next” to get the following screen as we work through the tool

TAC Case Collection

- Tool will clarify the problem area
- Edit problem type if query has been misunderstood

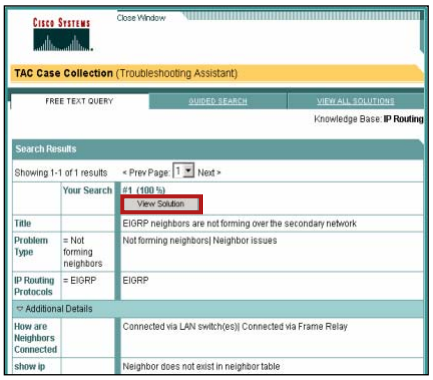


TAC Case Collection

- On this query confirmation page here, the tool will clarify the problem area; you also have the opportunity to edit the problem type if your query has been misunderstood
- In our example, though, the tool correctly understood my problem and identified EIGRP as the IP routing protocol associated with my problem
- Upon clicking the “next” button, the search is submitted to the TAC Case Collection solution database

TAC Case Collection

- Search results table
- Click “View Solution” links to be taken to solution(s)

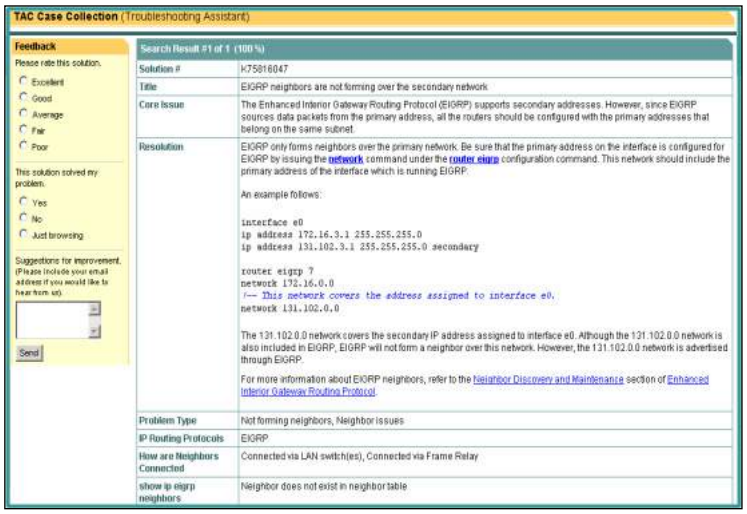


TAC Case Collection

- And we get to the search results table, which shows one solution at 100% similarity
- By clicking the “View Solution” link, I’ll be taken to my solution

TAC Case Collection

Detailed Solution Page

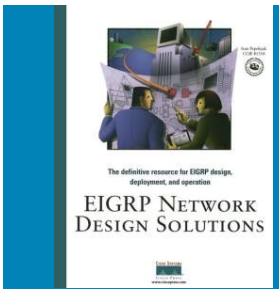


TAC Case Collection

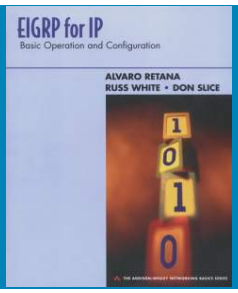
- And here the tool has provided us with a detailed solution page—where we see the solution number, title, core issue, resolution, along with links to other resources
- As a reminder, I'd reiterate that the solutions presented here are from actual service requests taken by TAC engineers
- Before leaving the TAC Case Collection, I'd like to mention that there is a feedback form on the left-hand side of the page beside the detailed solution; we invite you to please take a few moments to rate the solution, let us know if it resolved your problem, and provide us with any additional feedback

Q and A

Recommended Reading



ASIN: 1578701651



ISBN: 0201657732

ISBN: 1-58705-019-6

- Continue your Networkers at Cisco Live learning experience with further reading for this session from Cisco Press
- Check the Recommended Reading flyer for suggested books

Available Onsite at the Cisco Company Store

Complete Your Online Session Evaluation

- Give us your feedback and you could win fabulous prizes. Winners announced daily.
- Receive 20 Passport points for each session evaluation you complete.
- Complete your session evaluation online now (open a browser through our wireless network to access our portal) or visit one of the Internet stations throughout the Convention Center.

Don't forget to activate your Cisco Live virtual account for access to all session material on-demand and return for our live virtual event in October 2008.

Go to the Collaboration Zone in World of Solutions or visit www.cisco-live.com.