



Enhanced Interior Gateway Routing Protocol

ROUTE Module 2

Agenda

- **Introduction**
- **Basic Principles and Terminology**
- **Configuration**
- **Securing EIGRP**
- **Advanced features**
 - Stub Networks
 - SIA-Query
 - Graceful Shutdown
- **EIGRP in NBMA**

Enhanced Interior Gateway Routing Protocol ①

- [Document ID: 16406 \(Enhanced Interior Gateway Routing Protocol\)](#)
- Cisco proprietary protocol invented with support of SRI International (prof. J.J. Garcia-Aceveda)
- Enhanced **distance-vector** protocol uses
 - DUAL algorithm
 - Loop-free control
 - RTP (even for multicast)
 - Neighbor detection
 - Partial bounded updates
- Currently it is only routing protocol which **guarantees** (when configured appropriately) **loop-free topology**

Enhanced Interior Gateway Routing Protocol ②

- EIGRP has modular structure independent on routed protocol (L3 protocol)
 - IPv4, IPv6, IPX, AppleTalk
- Classless (VLSM), automatic and manual summarization, authentication, stub routing
 - Composite metric based on multiple factors
- IPv4 networks – 224.0.0.10 multicast address
- IPv6 networks – ff02::a multicast address
- Administrative distance
 - Internal EIGRP routes: 90
 - External EIGRP routes: 170
 - Summary routes (discard routes): 5

EIGRP Main Properties

- **Neighbor Detection**

- Every router has its own **neighbor table** where it stores information about directly connected neighbors

- **Reliable Transport Protocol (RTP)**

- Transport protocol independent on L3 protocol – protocol number 88
- Guarantees delivery of unicast and multicast communication

- **Protocol-dependent modules (PDMs)**

- **DUAL Finite-state Automata**

- It directs whole best route selection mechanism

- **Loop-free Topology Protection**

- Guarantees that each used next-hop doesn't cause routing loop in topology

Basic EIGRP Terminology and Principles

EIGRP Terms ①

- **Successor** – next-hop
 - Successor is next-hop router for destination network
 - Route through successor is loop less with shortest distance
- **Feasible successor** – backup next-hop
 - Potential candidate for next-hop for destination network
 - Route through FS is loop less but not with shortest distance
- **Reported distance** (RD, a.k.a. **advertised distance**)
 - The current best distance of the particular neighbor to the destination
- **Computed Distance** (CD)
 - The total metric of reaching the destination over the particular neighbor
- **Feasible distance** (FD)
 - A *historical* record, or a *historical* copy, of the smallest known CD toward a particular destination
- **Feasibility condition** (FC)
 - Condition that MUST be satisfied for target next-hop router
 - Tells whether direction to destination network is loopless

EIGRP Terms ②

- **Neighbor table**

- Table with information about neighbors

- **Topology table**

- Table with information about destination networks – state, CDs, FDs, RDs, next-hop, etc.
 - Table itself doesn't contain topology description, only list of destination networks

- **Passive state**

- State of destination network when successor is known and route is converged and usable

- **Active state**

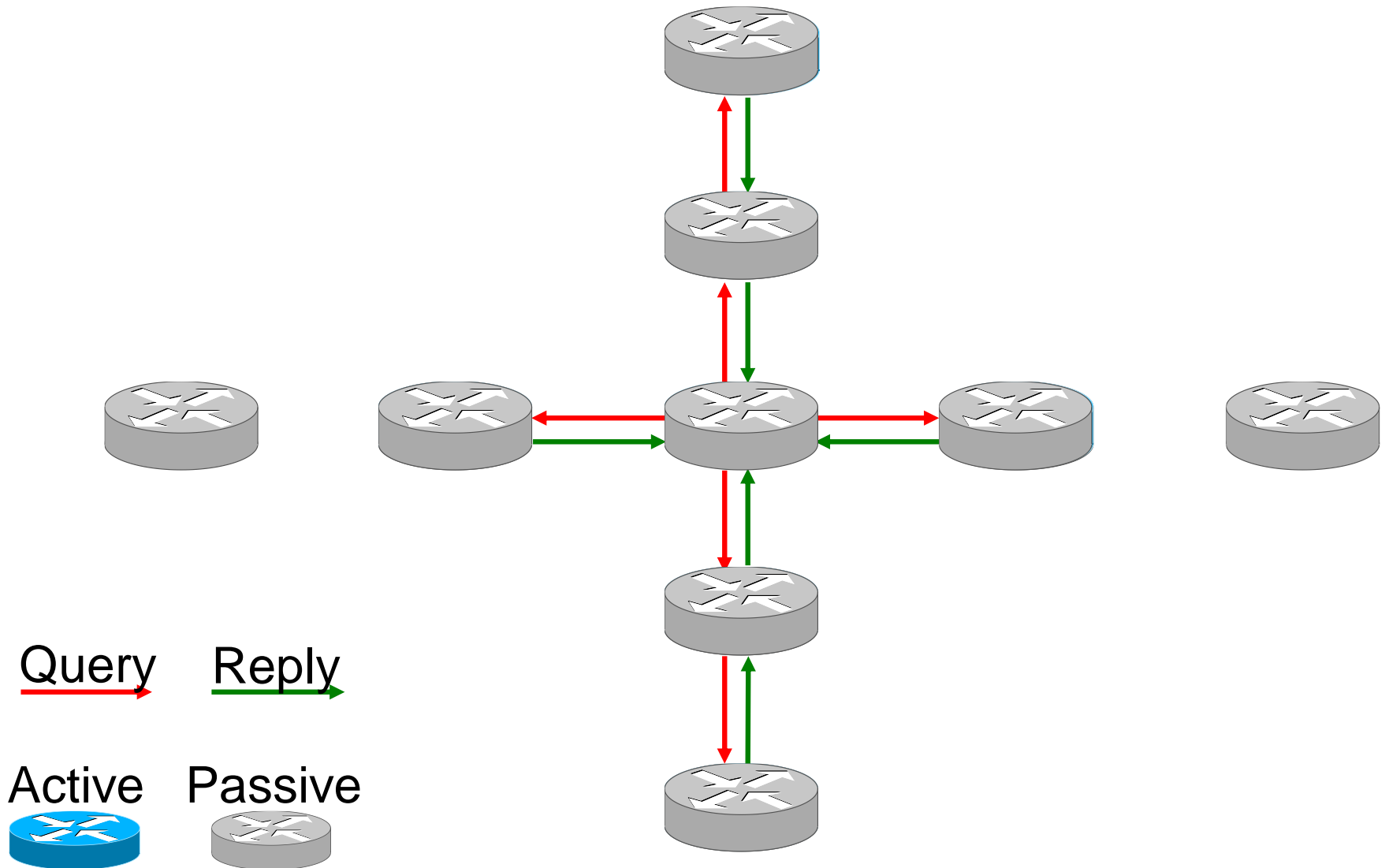
- State of destination network when no successor is known and there is no FS either
 - Router is actively searching/computing successor and best route

EIGRP Terms ③

▪ Diffusal computations

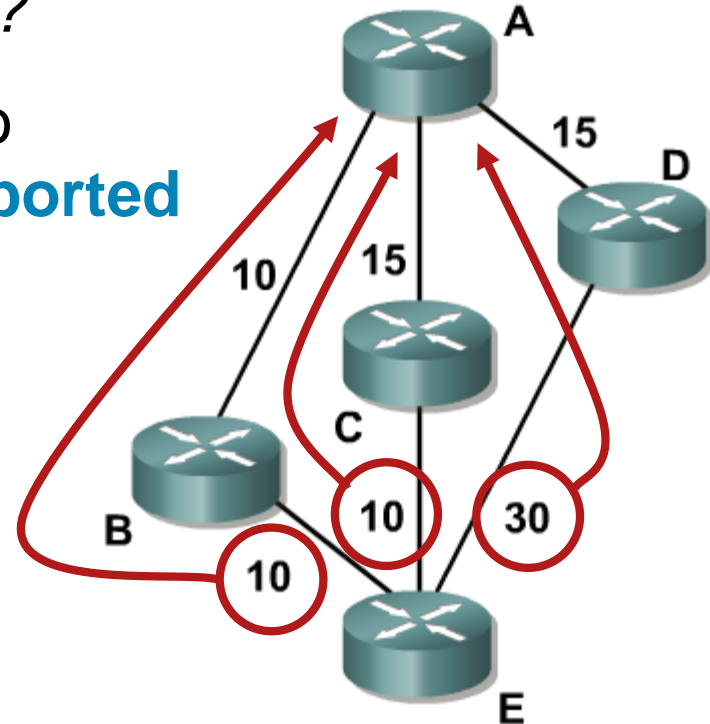
- Way of managing distributed computation of best-route selection
- Goal is to run computation, gather information about its progress, collect all results and correctly inform about its end
 - DUAL begin with sending **query** to neighbors
 - IF neighbor knows answer THEN neighbor **replies** to the query
ELSE neighbor router generates own queries to its neighbor routers
 - Router must send reply in order – router knows answer AND router isn't waiting for reply on its own query

Diffuse Computation



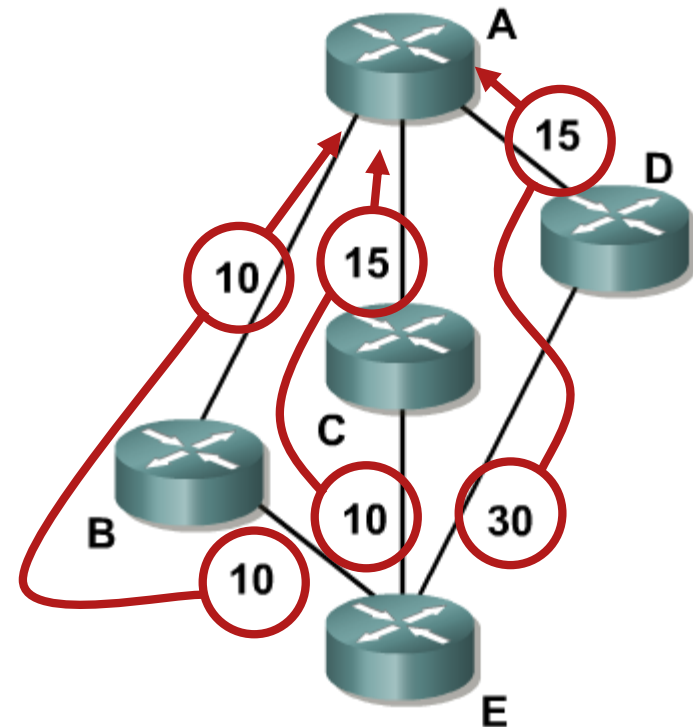
Use-Case ①

- *How does EIGRP know which routes don't cause loops in topology?*
- It uses neighbors distances to destination network called **reported distances (RD)**
- Every neighbor router of *A* is advertising its RD to *E*
 - $RD(B) = 10$
 - $RD(C) = 10$
 - $RD(D) = 30$



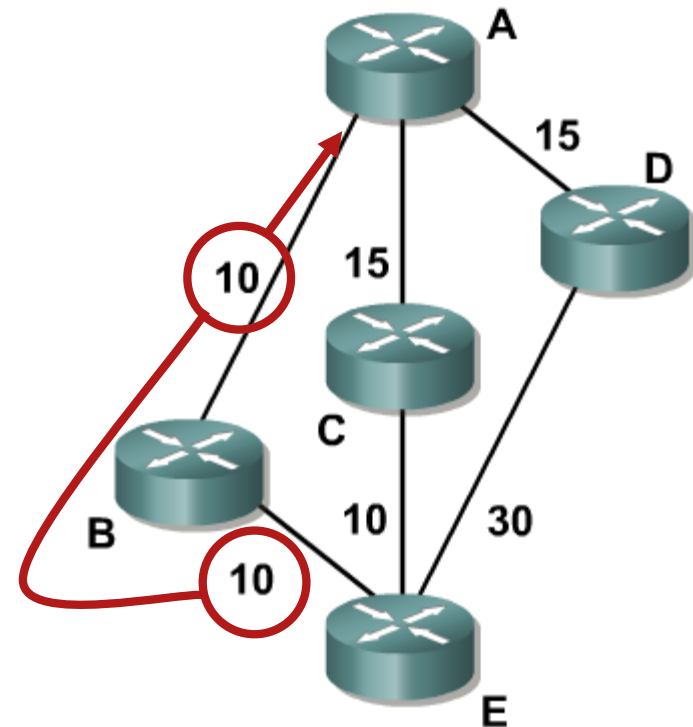
Use-Case ②

- For *A*, the computed distance to *E* is:
 - $\text{via}(B) = 20$
 - $\text{via}(C) = 25$
 - $\text{via}(D) = 45$
- **feasible distance (FD)** is the lowest distance – 20 in this case
- More precisely, FD is the best known distance ever to the destination network *E*, from the *A* point of view



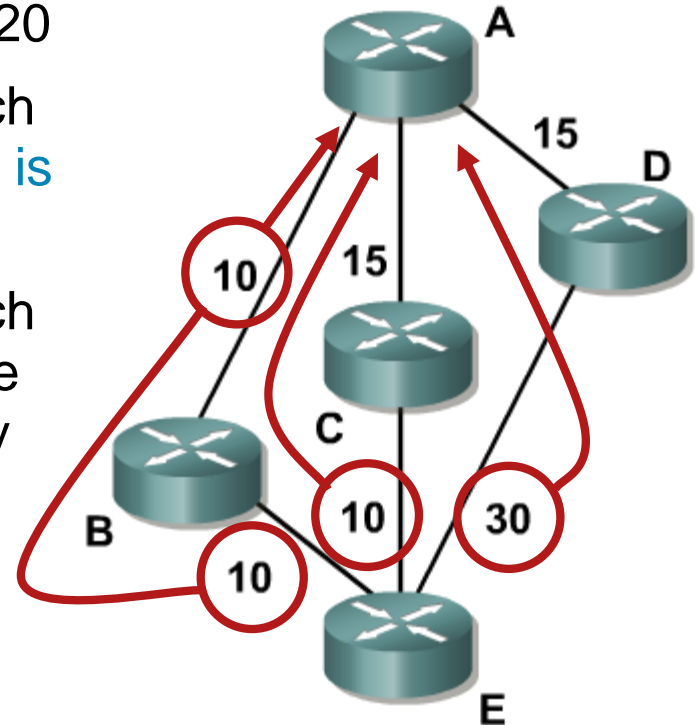
Use-Case ③

- Router A is using FD and RD to check **feasibility condition**
- FD is etalon for this validation – every route with **RD < FD** is without any doubts loop less
- Some loop less routes are (falsely) denied by this rule
- But it never accepts route which certainly cause loop in topology



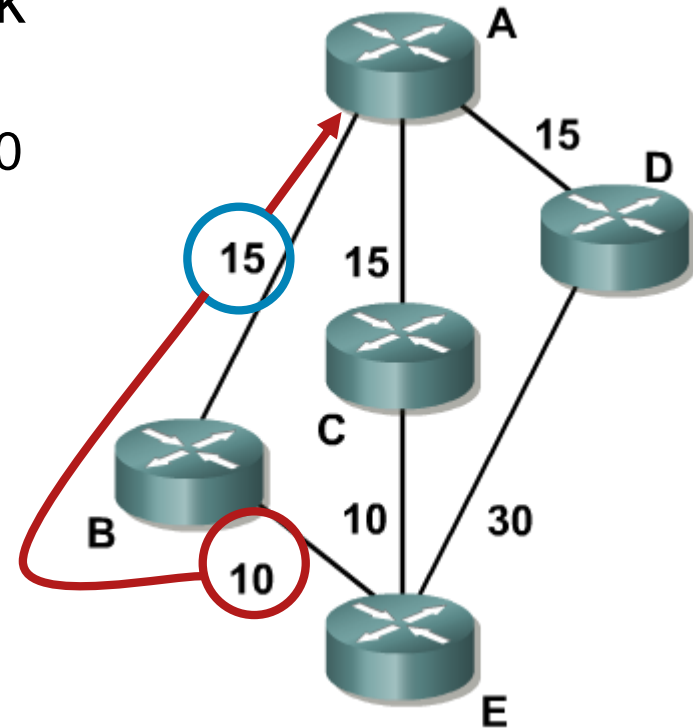
Use-Case ④

- Router A:
 - Route via *B* is best with $FD(B) = 20$
 - Route via *C* has $RD(C) = 10$ which is lower than FD hence this route is loop less
 - Route via *D* has $RD(D) = 30$ which is higher than FD hence this route “potentially” could cause topology loop



Use-Case ⑤

- FD measure **historically lowest** distance to destination network
 - Lets imagine that cost of link between *A* to *B* increases from 10 to 15
 - Lowest distance from *A* to *E* will be 25, but **FD remains on value 20**
 - Value 25 will appear in routing table of *A* and in updates sent from *A* to neighbors
 - FD is used internally by EIGRP but it's not advertised anywhere



Feasible Distance

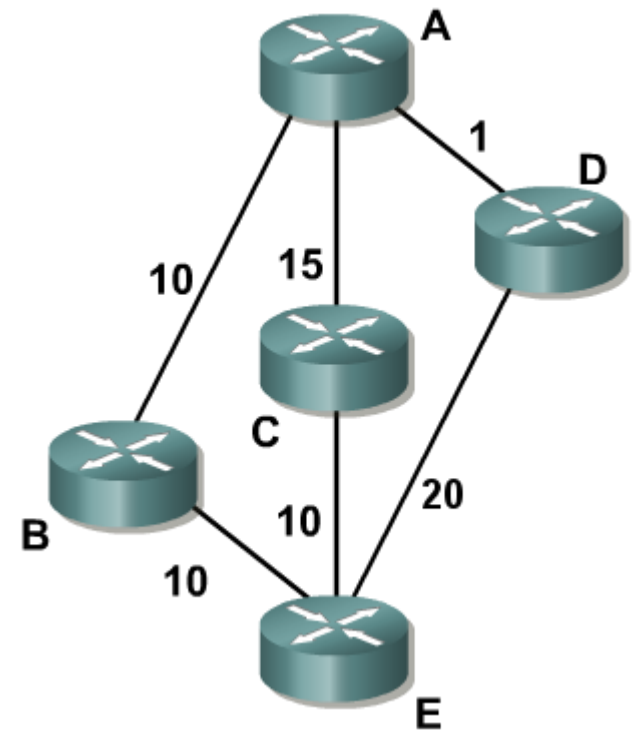
- FD could be changed only during these occasions:
 - Better route found – FD is decremented to a new value
 - Successor is lost, no feasible successor available, route transits to active state – DUAL computation is trying to calculate new FD
- FD could only decrement during passive state
 - Whenever FD increases router must transit to active state and diffuse computation begins
- Other FD definition:
 - Value of **historically shortest path** to destination network – “history” ends with transition to active state
- Meaning of FC:
 - IF *our neighbor is closer to destination network (than we have ever been)* THEN *there could not be topology loop closing through us*

Feasible Successor ①

- EIGRP has RDs and CDs in topology table from all of its neighbors
- Scenario
 1. FD has been changed
 2. Router searches topology table for new neighbor with lowest CD
 3. Router checks founded possible feasible successor **IF** route through new neighbor satisfy FC
 - A. `true` **THEN** neighbor is used as new next-hop router
 - B. `false` **OR** new candidate couldn't be found in topology table **THEN DUAL** is initiated

Feasible Successor ②

- In converged topology from *A* to *E*:
 - FD is 20
 - $CD(B) = 20$ is the shortest
 - $CD(C) = 25$, satisfy FC
 $RD(\text{via } C) = 10 < 20$
 - $CD(D) = 21$, but it doesn't satisfy FC
 $RD(\text{via } D) = 20 < 20$
- Scenario:
 1. *B* goes down
 2. *A* knows that shortest route goes via *D* because $\text{via}(C) > \text{via}(D)$
 3. But *D* doesn't satisfy FC;
IF *A* decide to use route via *C* THEN it won't be shortest
 4. Hence, DUAL is started



EIGRP Functionality ①

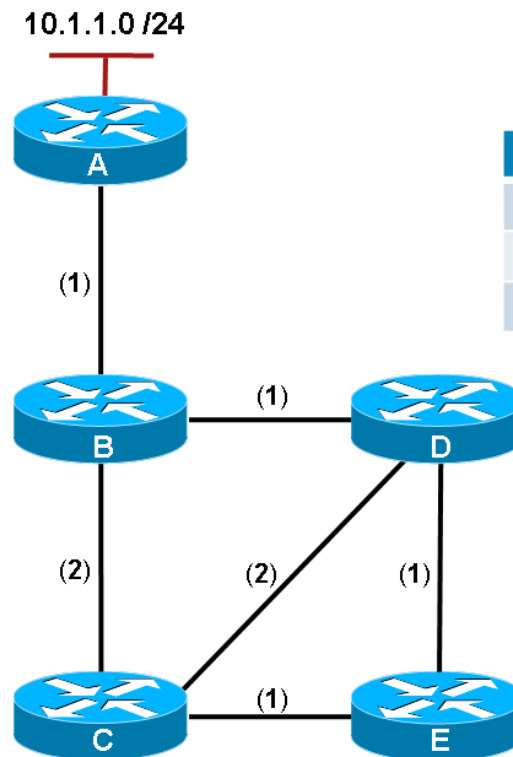
- *DUAL itself is quite trivial 😊*
 - DUAL initiating router sends **query packet** in which it advertises **new distance to destination network via current successor** (it could be infinite)
 - Neighbor receiving this packet updates its topology table according to information:
 - a) IF this doesn't cause route failure on neighbor THEN neighbor just answers with its own current distances
 - b) IF neighbor lost current route to destination network AND doesn't know any possible feasible successors THEN neighbor initiates own DUAL computation and asks its neighbors
- *It's not **extra computation** just spreading queries and waiting for replies!*

EIGRP Functionality ②

- IF router recognized **increase of distance** to destination network AND transits route to active state THEN it will do following:
 1. Router sets its RD and FD to increased value
 2. Router generates query packet to all possible neighbors
 3. WHILE router is waiting to receive all replies THEN it can't change neither of these values nor updates its routing table
 4. WHEN router has received all replies THEN it's allowed to update RD, FD and routing table accordingly to new situation
- Correct diffuse computation (choice of next-hop router and validating of FC) is directed by **DUAL finite-state machine**

DUAL Example

DUAL Example ①



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	2		***** Passive *****
via B	2	1	Successor
via C	5	3	

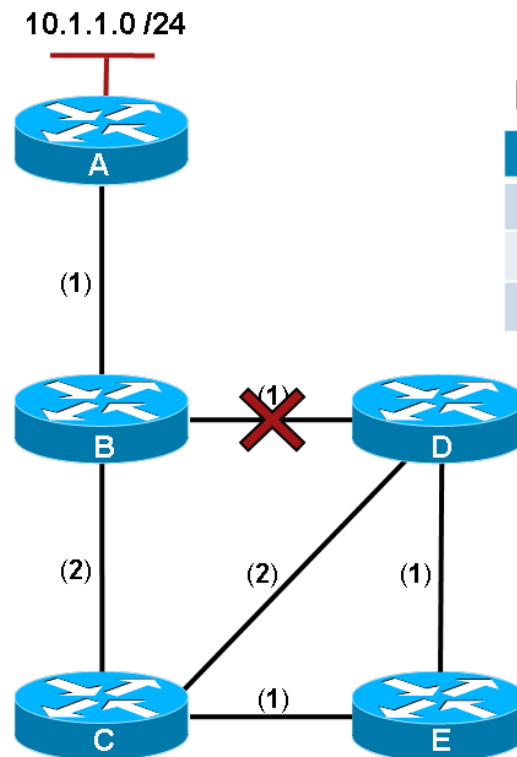
Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D	4	2	Feasible Successor
via E	4	3	

Router E

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via D	3	2	Successor
via C	4	3	

DUAL Example ②



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	2		***** Passive *****
via B	2	1	Successor
via C	5	3	

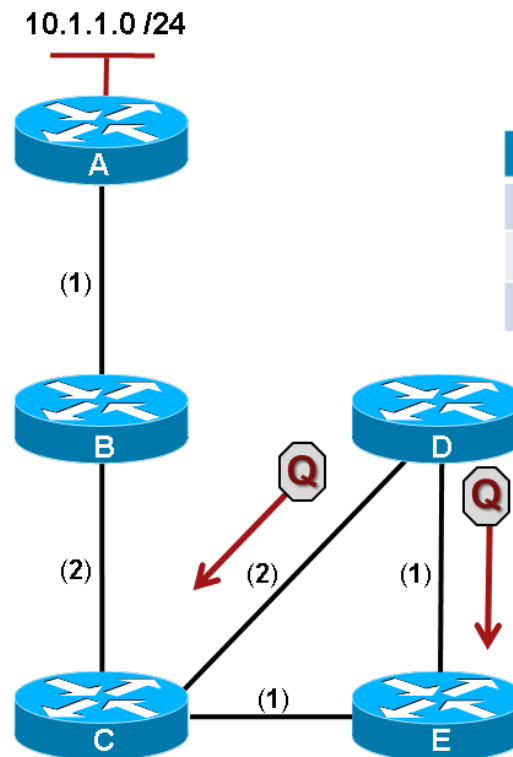
Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D	4	2	Feasible Successor
via E	4	3	

Router E

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via D	3	2	Successor
via C	4	3	

DUAL Example ③



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	-1		***** ACTIVE *****
via E			(Q) Query
via C	5	3	(Q) Query

Q = Query

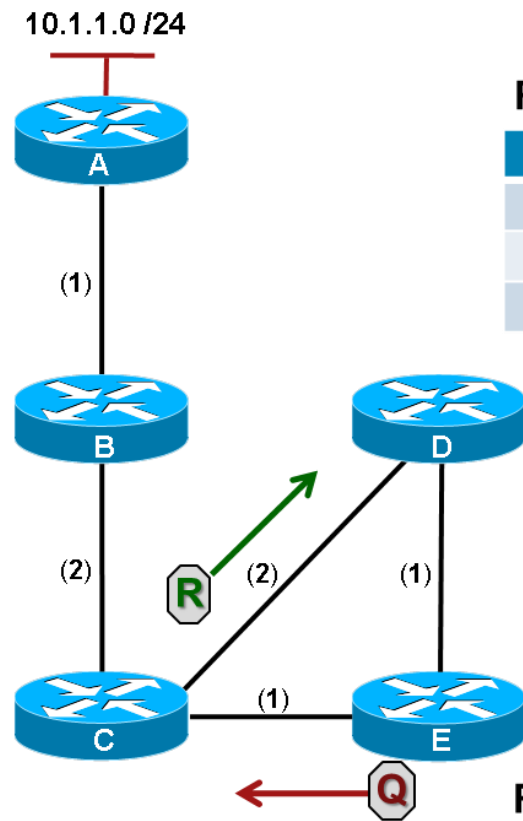
Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D	4	2	Feasible Successor
via E	4	3	

Router E

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via D	3	2	Successor
via C	4	3	

DUAL Example ④



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	-1		***** ACTIVE *****
via E			(Q) Query
via C	5	3	

Q = Query
R = Reply

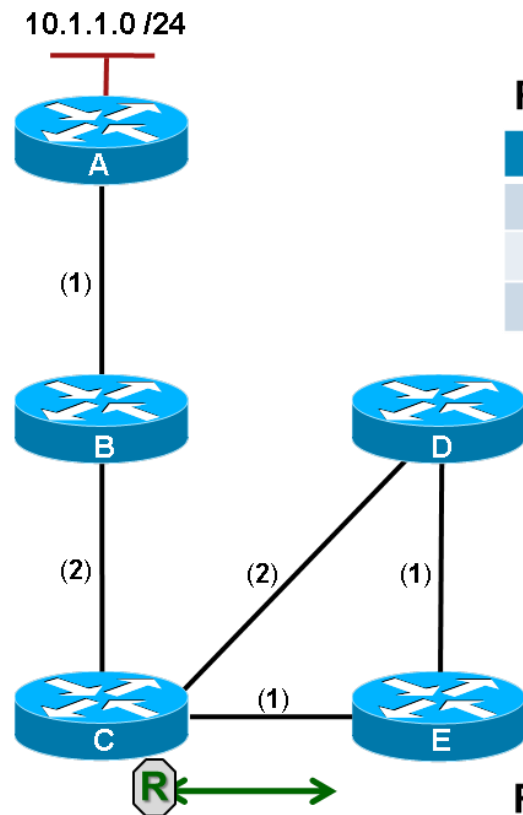
Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D			
via E	4	3	

Router E



EIGRP	FD	AD	Topology
10.1.1.0 /24	-1		***** ACTIVE *****
via D			
via C	4	3	(Q) Query

DUAL Example ⑤



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	-1		***** ACTIVE *****
via E			(Q) Query
via C	5	3	

 = Query
 = Reply

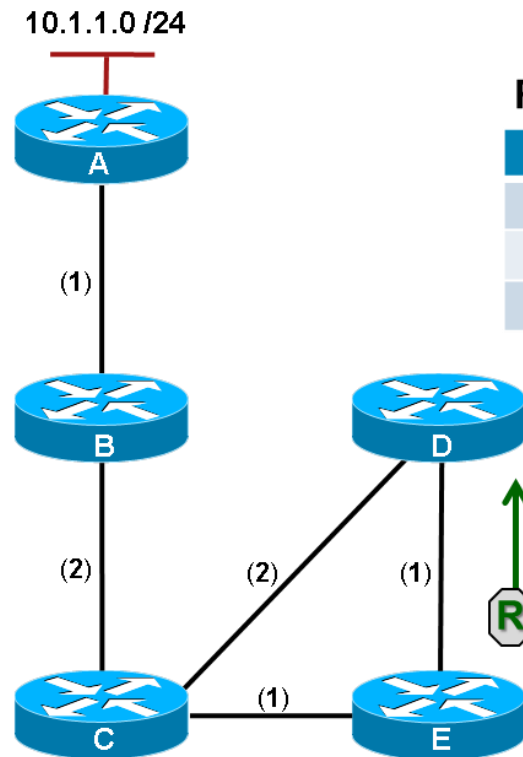
Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D			
via E			

Router E



EIGRP	FD	AD	Topology
10.1.1.0 /24	4		***** Passive *****
via C	4	3	Successor
via D			

DUAL Example ⑥



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	5		***** Passive *****
via C	5	3	Successor
via E	5	4	Successor

 = Query
 = Reply

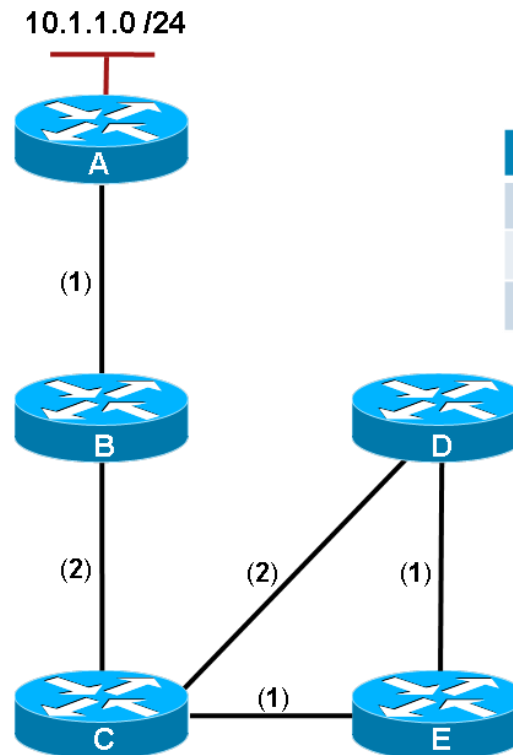
Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D			
via E			

Router E

EIGRP	FD	AD	Topology
10.1.1.0 /24	4		***** Passive *****
via C	4	3	Successor
via D			

DUAL Example ⑦



Router D

EIGRP	FD	AD	Topology
10.1.1.0 /24	5		***** Passive *****
via C	5	3	Successor
via E	5	4	Successor

Router C

EIGRP	FD	AD	Topology
10.1.1.0 /24	3		***** Passive *****
via B	3	1	Successor
via D			
via E			

Router E

EIGRP	FD	AD	Topology
10.1.1.0 /24	4		***** Passive *****
via C	4	3	Successor
via D			

Composite Metric

Metric

- **Composite metric** consists of 4 factors
 - **K1 – Bandwidth** (static parameter, turned on by default)
 - **K3 – Delay** (static parameter, turned on by default)
 - **K4, K5 – Reliability** (dynamically evaluated, turned off by default)
 - **K2 – Load** (dynamically evaluated, turned off by default)
 - **MTU** (some literature mentions it as tie-breaker, but it is in fact useless)
- For calculation are by default:
 - used: Bandwidth, Delay
 - unused: Reliability and Load
- Reliability and Load are measured only once at initial setup of route!
- EIGRP metric is 32bit unsigned integer, IGRP metric is computed same way but it's only 24bit long
 - $\text{EIGRP} = \text{IGRP} \ll 8$, $\text{IGRP} = \text{EIGRP} \gg 8$
- **All neighbors MUST share same K-values!**

Metric: Bandwidth

- EIGRP uses the slowest bandwidth (BW) in its metric calculation

$$BW = \frac{10^7}{\min BW} \cdot 256$$

- The value of the bandwidth may or may not reflect (e.g. serials) the actual physical bandwidth of the interface
- To modify the bandwidth value, use command:

```
Router(config-if) # bandwidth kilobits
```

Metric: Delay

- Delay is a measure of the time it takes for a packet to traverse a route

$$DLY = \frac{\sum delay}{10} \cdot 256$$

- The delay (DLY) metric is a static value based on the type of link to which the interface is connected and is expressed in microseconds

```
Router(config-if) #  
    delay tens-of-microseconds
```

Media	Delay
100M ATM	100 μ S
Fast Ethernet	100 μ S
FDDI	100 μ S
1HSSI	20,000 μ S
16M Token Ring	630 μ S
Ethernet	1,000 μ S
T1 (Serial Default)	20,000 μ S
512K	20,000 μ S
DSO	20,000 μ S
56K	20,000 μ S

Metric Calculation

- The EIGRP composite metric formula consists of values K1 through K5, known as EIGRP metric weights
- Default composite formula:

$$metric = K1 \cdot BW + K3 \cdot DLY$$

- Complete composite formula:

$$metric = \left(K1 \cdot BW + \frac{K2 \cdot BW}{256 - load} + K3 \cdot DLY \right) \cdot \frac{K5}{rel + K4}$$

Metric Calculation: Example ①

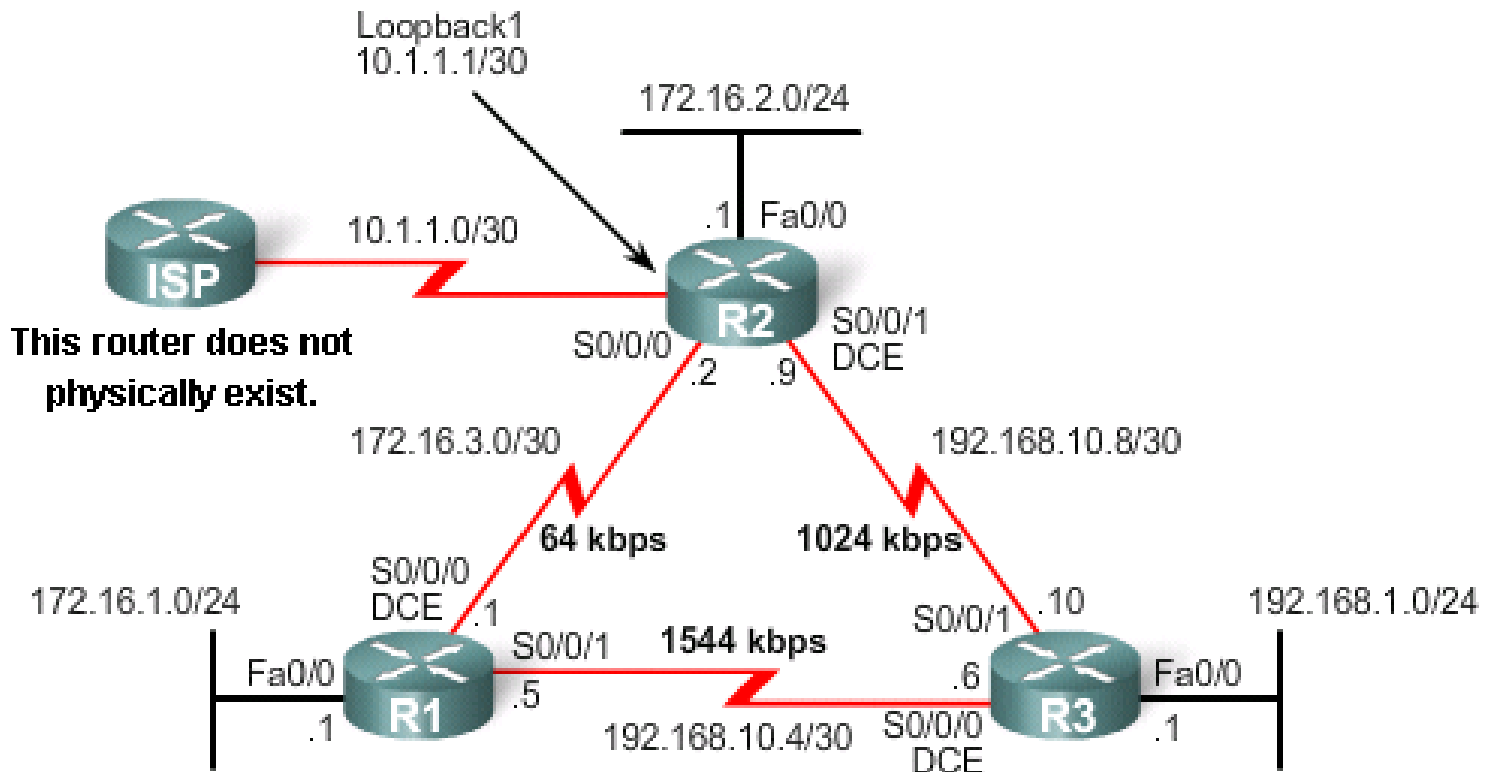
```
R2#show ip route
```

```
<code output omitted>
```

```
Gateway of last resort is not set
```

```
    192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.10.0/24 is a summary, 00:00:15, Null0
D       192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:15, Serial0/0/1
C       192.168.10.8/30 is directly connected, Serial0/0/1
    172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D       172.16.0.0/16 is a summary, 00:00:15, Null0
D       172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C       172.16.2.0/24 is directly connected, FastEthernet0/0
C       172.16.3.0/30 is directly connected, Serial0/0/0
    10.0.0.0/30 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Loopback1
D       192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
```

Metric Calculation: Example ②



- $BW = \frac{10^7}{1024} \cdot 256 = \underline{2499840}$
- $DLY = \frac{20000+100}{10} \cdot 256 = \underline{514560}$
- $metric = 2499840 + 514560 = \underline{3014400}$

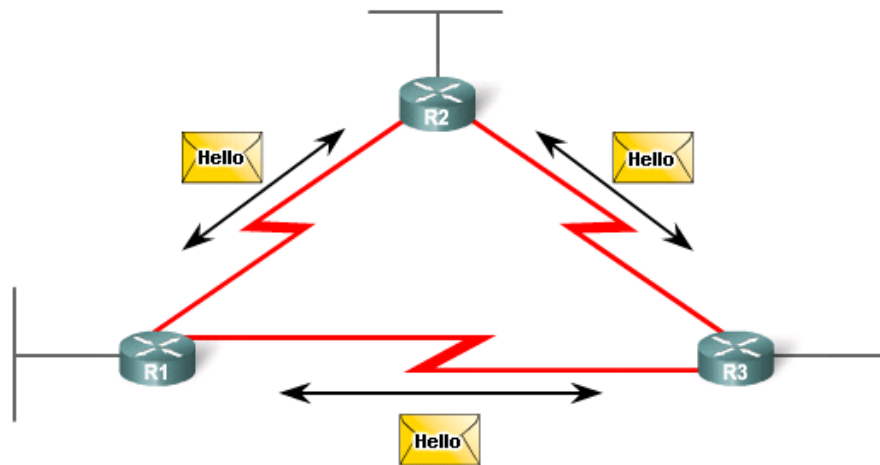
```
R2#show inter ser 0/0/1
Serial0/0/1 is up, line protocol is up
Hardware is PowerQUICC Serial
Internet address is 192.168.10.9/30
MTU 1500 bytes, BW 1024 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
<remaining output omitted>
```

EIGRP Message Exchange

Packet Types ①

■ Hello

- Detection of EIGRP neighbors, exchange of K-values, AS numbers, timers and authentication
- Sent periodically every 5 seconds (on links slower than T1 every 60 seconds) on [224.0.0.10](#)
- Hello packets are not acknowledged



Bandwidth	Example Link	Default Hello Interval	Default Hold Time
1.544 Mbps	Multipoint Frame Relay	60 seconds	180 seconds
Greater than 1.544 Mbps	T1, Ethernet	5 seconds	15 seconds

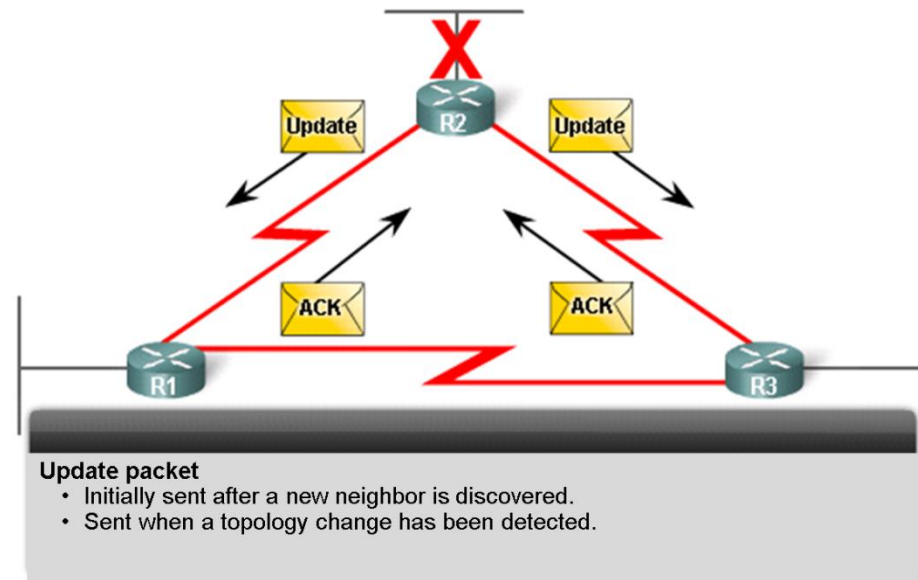
Packet Types ②

▪ Update

- Carries routing information
- Could be sent either as unicast or as multicast
- Update packets are acknowledged by receiving party
- They could cause receivers to start DUAL

▪ ACK

- Used for acknowledging Update, Query and Reply packets
- Sent as unicast (and of course are not acknowledged)
- *In principal it's Hello packet with empty body*



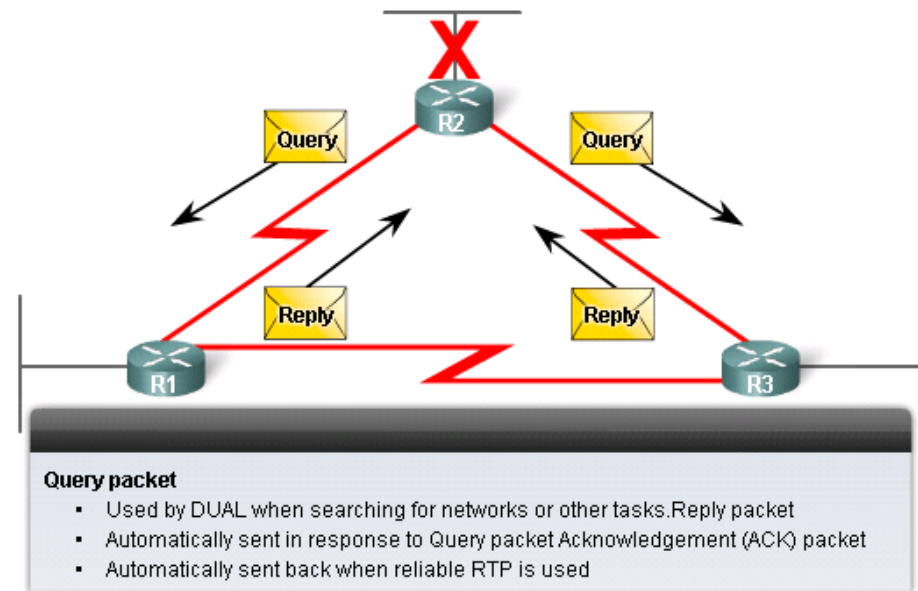
Packet Types ③

▪ Query

- Router is actively looking for best path to destination network in topology
- Usually sent as multicast
- Query packets are acknowledged
- They're starting DUAL or spreading it to neighbors

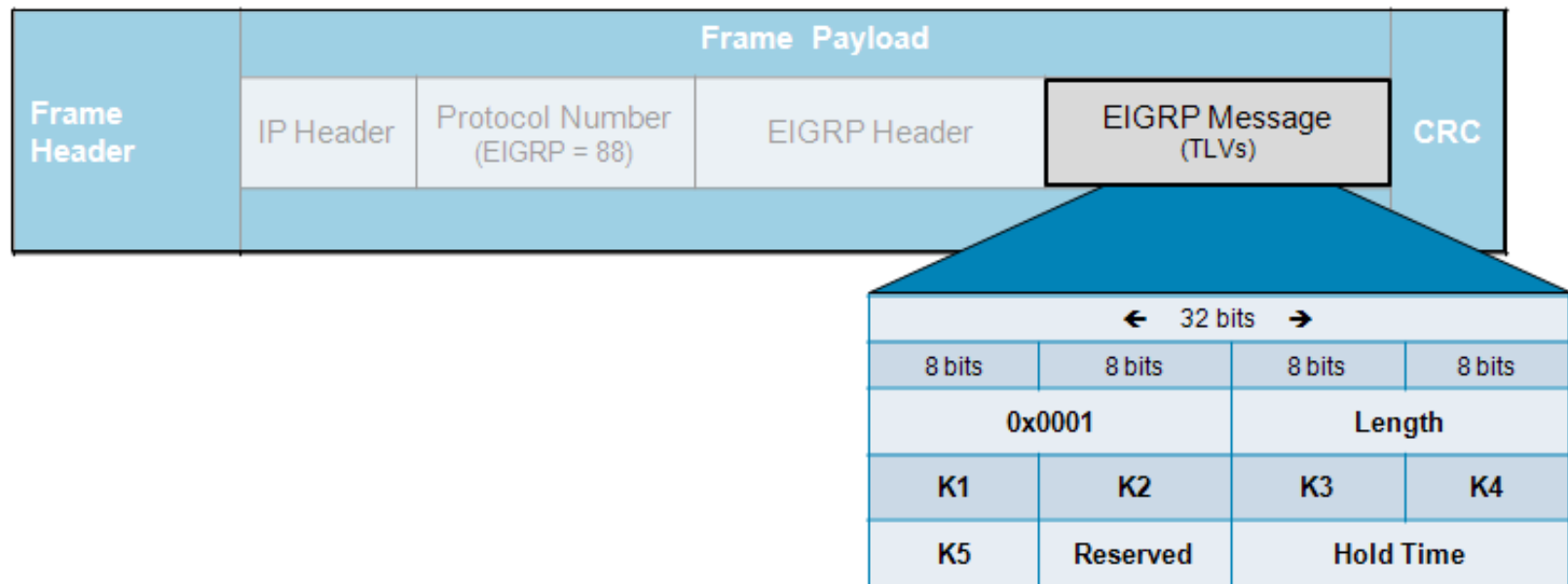
▪ Reply

- Answer to Query packet
- Sent as unicast
- Reply packets are acknowledged

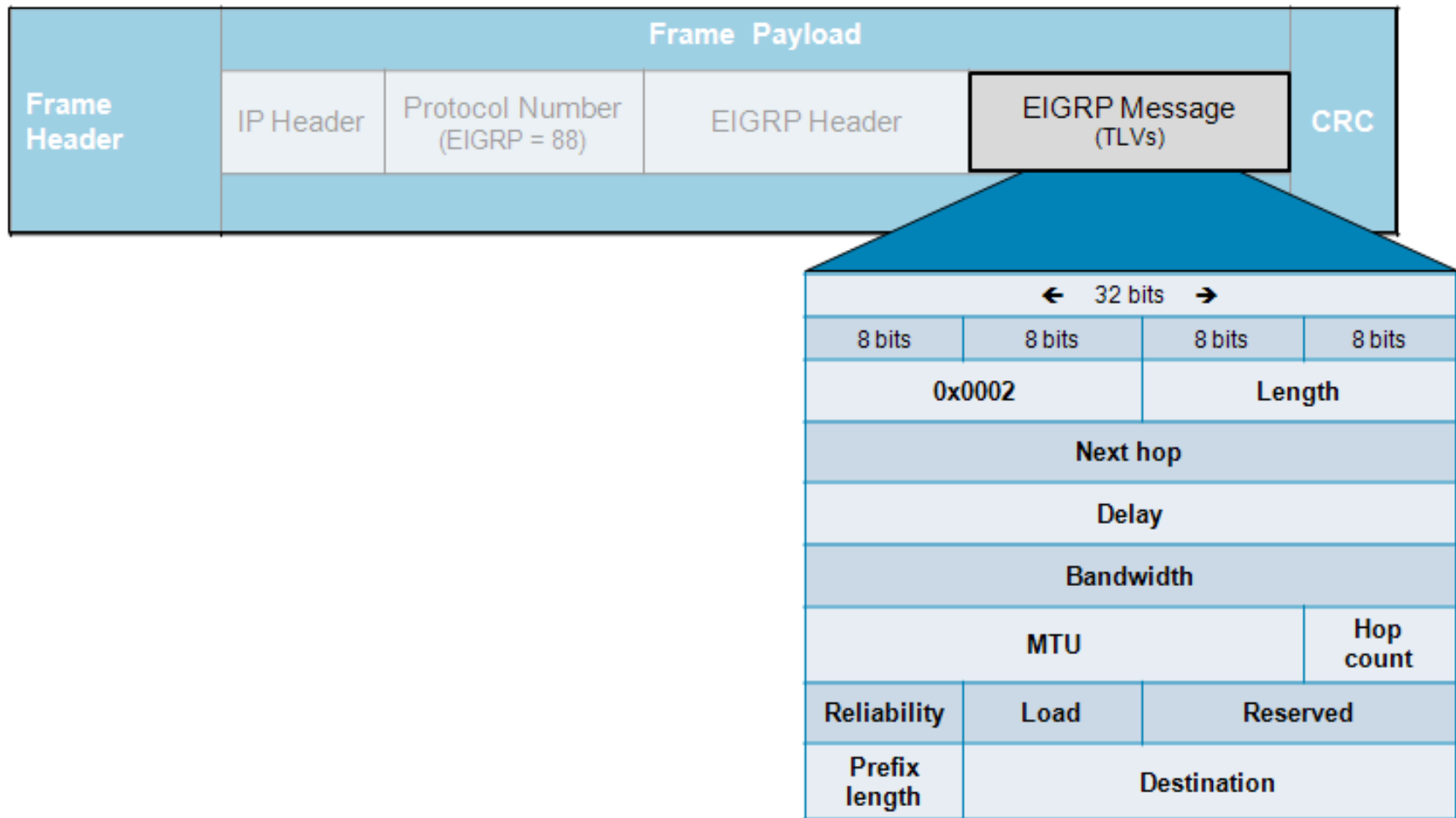


Packet Structure in More Detail ①

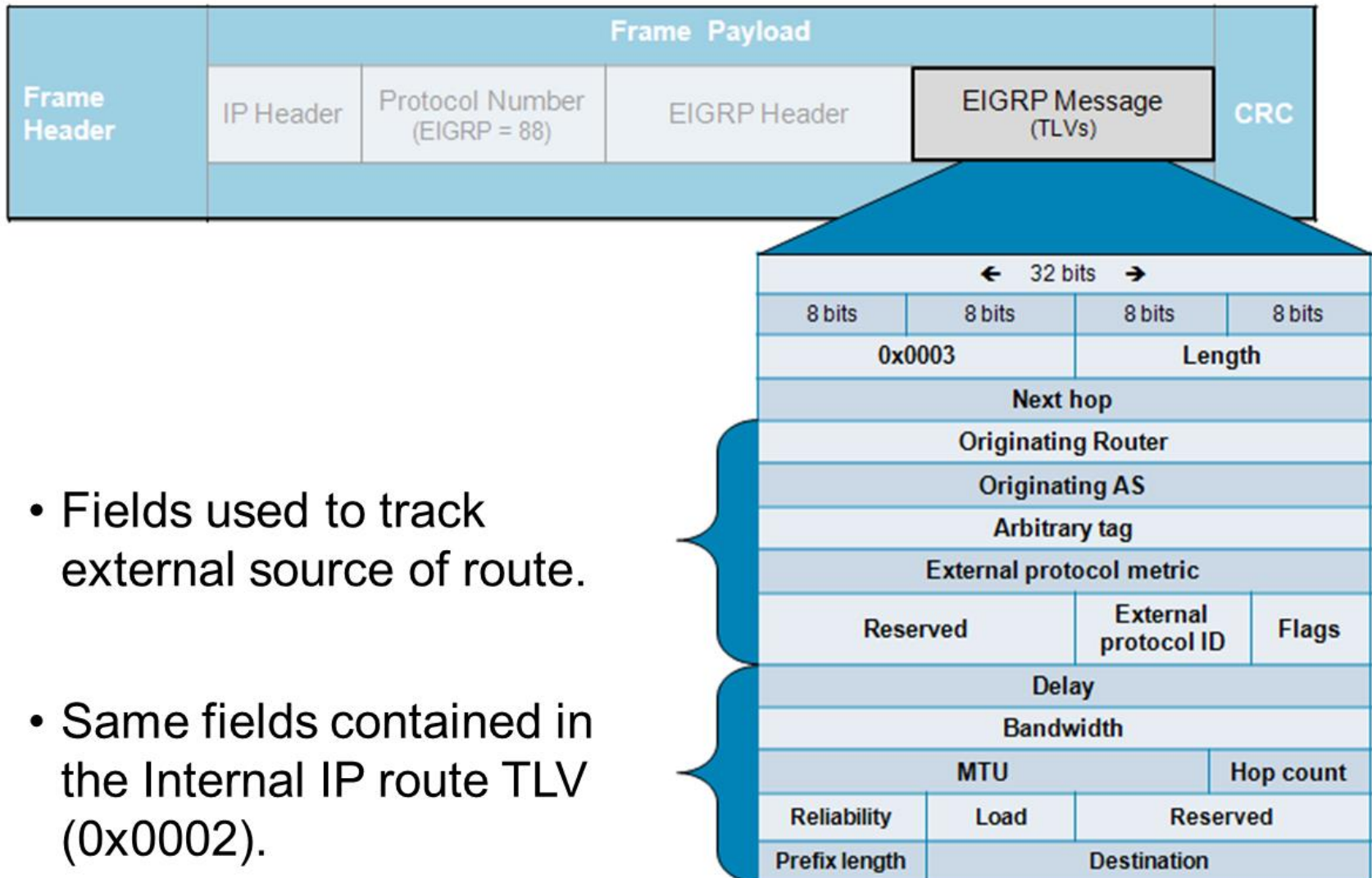
- *All perfectly designed protocols are using TLV!*
- Three TLV types:
 - 0x0001 – EIGRP parameters
 - 0x0002 – Internal routes
 - 0x0003 – External routes



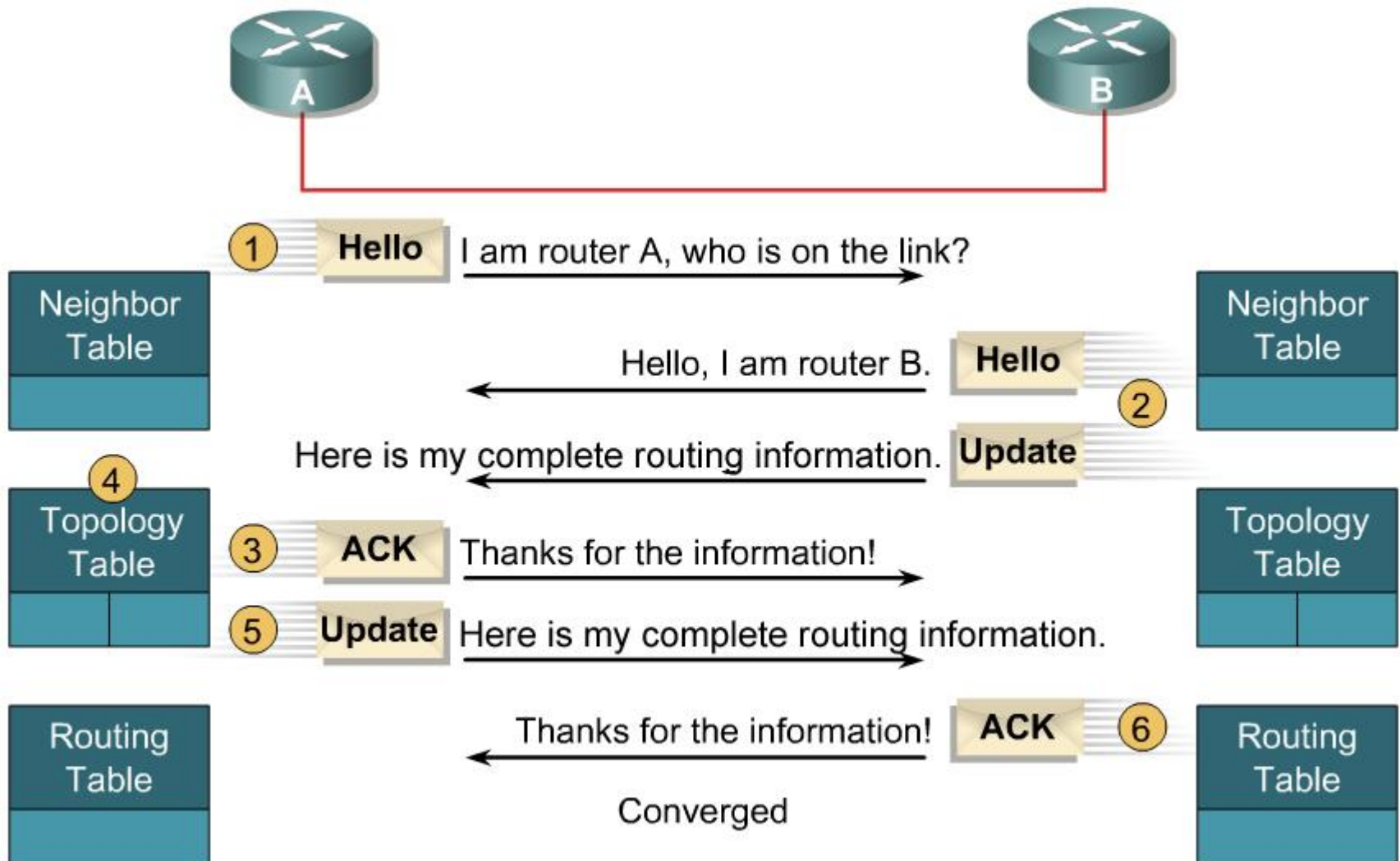
Packet Structure in More Detail ②



Packet Structure in More Detail ③



Initial Packet Exchange



Neighbor Table

```
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

In how many seconds will be neighbor announced as dead. Receiving of Hello packet resets this timer

For how long is neighbor up

Round Trip Time in manner of EIGRP request/response communication

In how many seconds router will generate new packet whenever the last one wasn't acknowledged

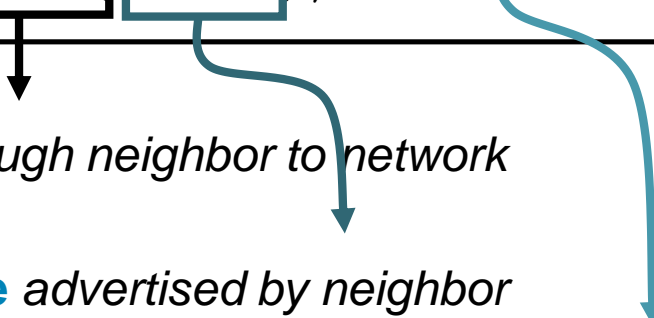
Topology Table

```
RTRA# show ip eigrp topology
```

```
IP-EIGRP Topology Table for AS(1)/ID(192.168.1.1)
```

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

```
P 192.168.3.0/24, 1 successors, FD is 409600  
   via 10.0.0.3 (409600/128256), FastEthernet0/0
```

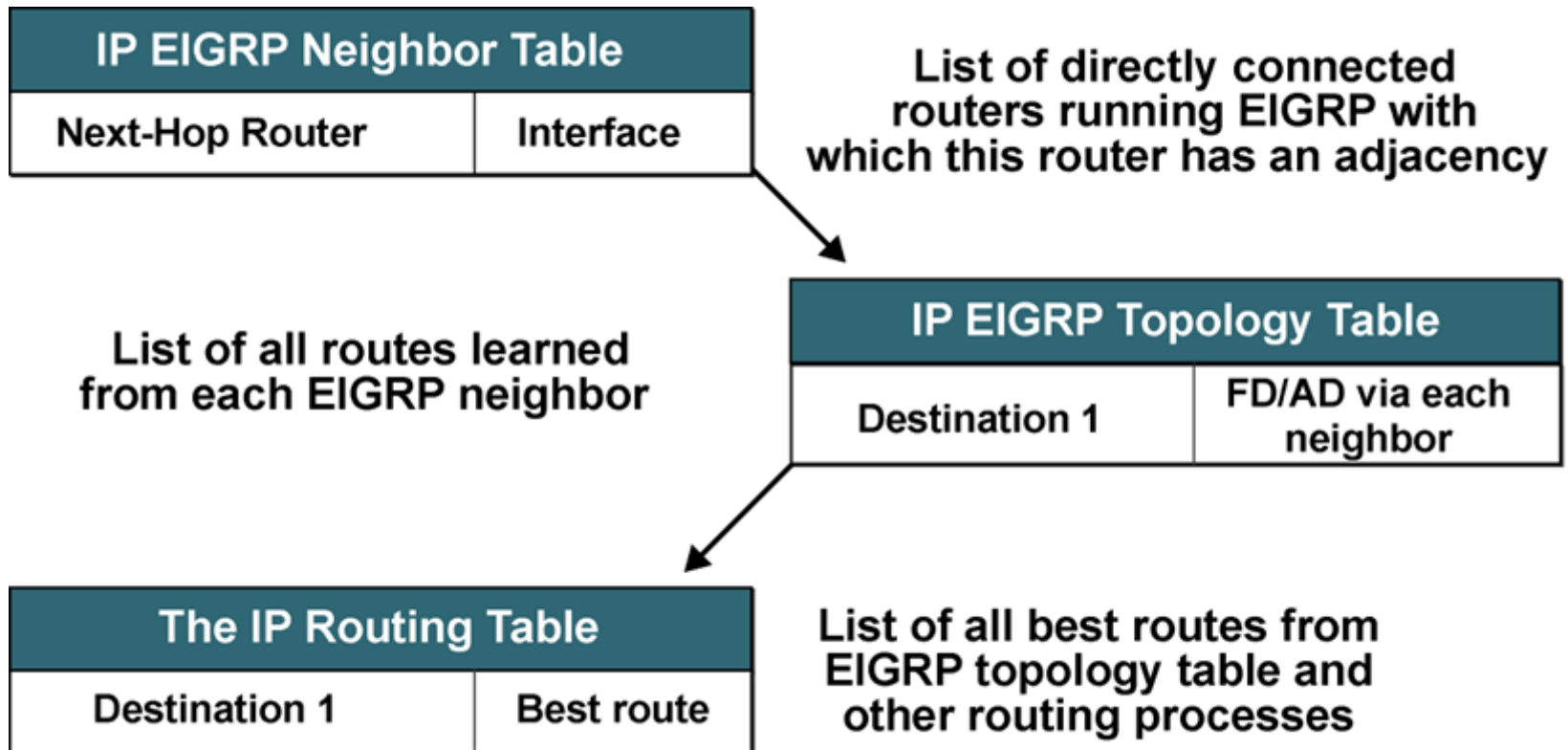


Current/computed distance *through neighbor to network*

Reported Distance *advertised by neighbor*

Feasible Distance *for destination network*

Workflow of Routing Information ①



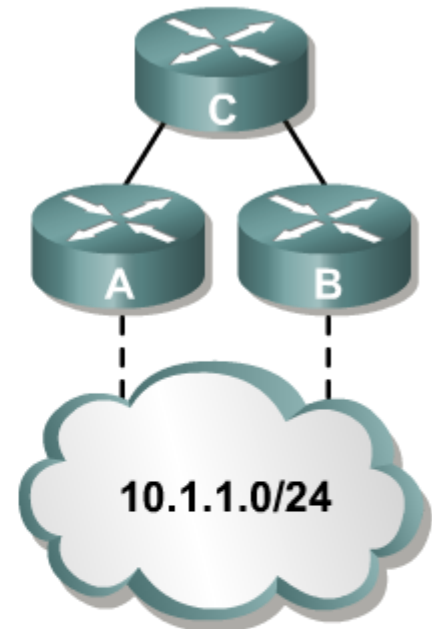
Workflow of Routing Information ②

On router C:

IP EIGRP Neighbor Table	
Next-Hop Router	Interface
Router A	Ethernet 0
Router B	Ethernet 1

IP EIGRP Topology Table			
Network	Feasible Distance (EIGRP Metric)	Advertised Distance	EIGRP Neighbor
10.1.1.0 /24	2000	1000	Router A (E0)
10.1.1.0 /24	2500	1500	Router B (E1)

The IP Routing Table			
Network	Metric (Feasible Distance)	Outbound Interface	Next Hop (EIGRP Neighbor)
10.1.1.0 /24	2000	Ethernet 0	Router A



EIGRP Configuration

Basic Configuration Guide ①

Router (config) #

router eigrp *autonomous-system-number*

- Activates EIGRP and binds it to AS number
 - All routers which suppose to exchange routing information **must** belong to same AS

Router (config-router) #

no auto-summary

network *network-number* [*wildcard-mask*]

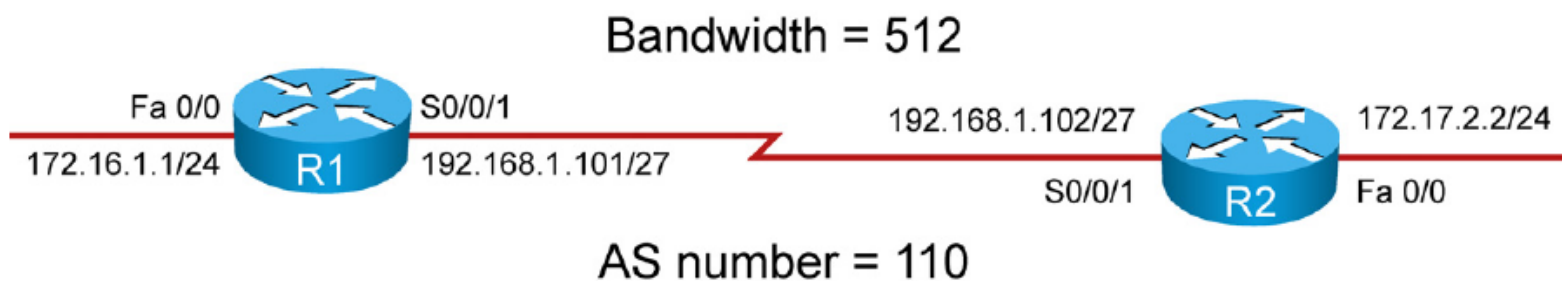
- Target network is associated with EIGRP routing process
- Wildcard parameter isn't mandatory but recommended
 - IF parameter isn't used THEN network behave as classful

Basic Configuration Guide ②

```
Router(config-if) # bandwidth kilobits
```

- On interfaces it's necessary to specify real bandwidth to let EIGRP works properly
- EIGRP is using bandwidth...
 - when computing metric
 - when computing how much BW could EIGRP internally use for signaling traffic
- It's **not recommended** to use **bandwidth** to influence computation of metric
 - Instead of this use the **delay** command

Example: EIGRP Basic Startup



R1#

```
interface FastEthernet0/0
  ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/1
  bandwidth 512
  ip address 192.168.1.101 255.255.255.224
!
router eigrp 110
  network 172.16.1.0 0.0.0.255
  network 192.168.1.0
```

R2#

```
interface FastEthernet0/0
  ip address 172.17.2.2 255.255.255.0
!
interface Serial0/0/1
  bandwidth 512
  ip address 192.168.1.102 255.255.255.224
!
router eigrp 110
  network 172.17.2.0 0.0.0.255
  network 192.168.1.0
```

Change of Metric Computation

- In case of need it's possible to change boolean K-values
- Configuration snippet:

```
Router(config-router) # metric weights 0 K1 K2 K3 K4 K5
```

- *Usually changing the weights is not recommended and have to be consulted with Cisco technical support ☺*

Change of Timers

- EIGRP has two timers
 - **Hello Interval** – interval between sending of two consecutive Hello packets
 - **Hold Time** – maximal time after receiving Hello packet in which is router considered alive
 - Routers Hold Time **is used by neighbors**
- Timers between neighbors could be different!!!
 - *Every router could send Hello packet with different frequency and advertise different Hold Times*
 - There's no (mathematical) connection between Hello Interval and Hold Time, change of the one doesn't influence other
- Command snippet for per-interface change:

```
Router(config-if) #
```

```
ip hello-interval eigrp as-number hello-interval
```

```
ip hold-time eigrp as-number hold-time
```

Default Route ①

- Default route information could be installed in two ways:
 1. Either as static 0.0.0.0/0 route
 2. Or as destination network with flag describing it as default route
- 1. Static 0.0.0.0/0 route must be redistributed into EIGRP
- 2. Using **ip default-network** command
 - Default route ≈ route to “default-network”
 - Typically this “default-network” is interconnecting border router of EIGRP AS with ISP

Default Route ②

- Command guideline for “default-network” has two steps:
 1. Network which will be classified as „default-network“ must be included in topology – e.g. defined by **network** command or redistributed into it
 2. On border router classify it with command **ip default-network**
- Disadvantage of this approach:

- `ip default-network` is classful!
- IF *A.B.C.D* isn't major network (it's classless) THEN configuration would automatically generate command:

```
ip route CL_NET(A.B.C.D) CL_MASK(A.B.C.D) A.B.C.D
```

- Following this generated route MUST be redistributed into EIGRP:

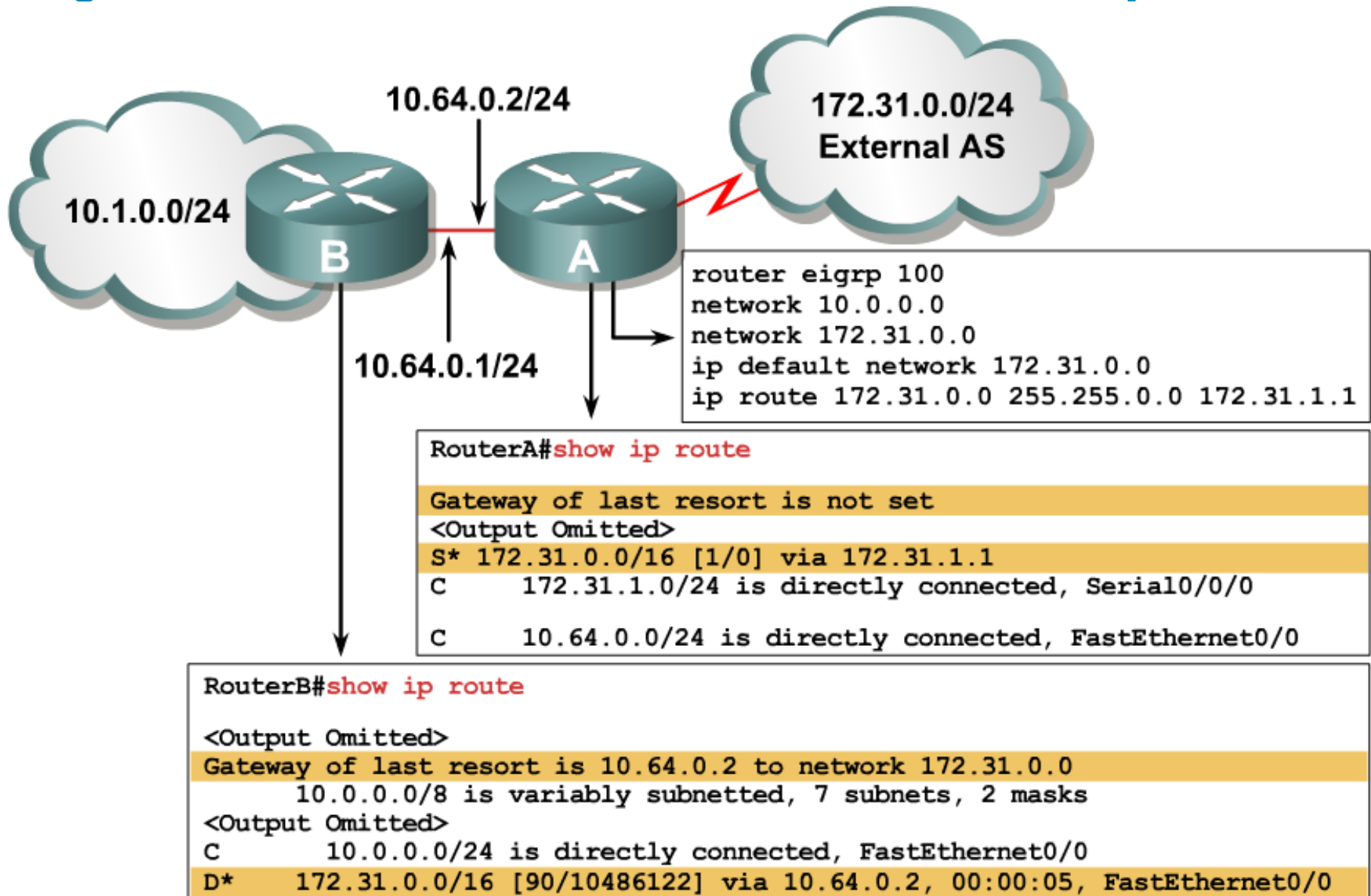
```
Router(conf-router) # redistribute static
```

- And finally classified as “default-network”:

```
Router(conf) # ip default-network CL_NET(A.B.C.D)
```

- Document ID : 16448

ip default-network Command Example



Load Balancing ①

- *Every routing protocol is able to do equal-cost load balancing...*
- *...but only EIGRP is capable of unequal-cost load balancing thanks to FC and feasible successor!*
- Successor is always used
- Following command influences EIGRP load-balancing:

```
Router(config-router) # variance multiplier
```

- In routing-table will be inserted every route going to destination network through feasible successor satisfying current distance in interval $\langle Distance, multiplier \cdot Distance \rangle$
- *In short words command tells how “bad” feasible successor route could leak from topology table!*

Load Balancing ②

- Load balancing could work on **per-packet** or **per-destination** basis
- It's recommended to have CEF enabled to improve routers functionality:

```
Router(config)# ip cef
```

- Configuration snippet in **outgoing interface** section:

```
Router(config-if)# ip load-sharing per-packet  
Router(config-if)# ip load-sharing per-destination
```

EIGRPv6

EIGRPv6 – Basics

- EIGRP for IPv6 is available since IOS 12.4T
- 4-bytes RouterID is required
 - IF RouterID cannot be discovered (no IPv4 address on loopback or interface) THEN it has to be specified manually
 - Ensure that RouterID is set in IPv6 only networks!
- There are two ways how to configure EIGRP for IPv6
 - 1) Using interface section
 - 2) Using Named Mode – new way, similar as BGP config

EIGRPv6 Interface Configuration

- Add interface to a EIGRP ASN

```
Router(config)# interface fa0/0
Router(config-if)# ipv6 eigrp 64512
Router(config-if)# exit
```

- Ensure that router has router-id set in case that there is no IPv4 address

```
Router(config)# ipv6 router eigrp 64512
Router(config-rtr)# router-id 158.193.138.255
```

- Sometimes it is necessary to enable EIGRP routing (depends on the IOS version)

```
Router(config)# ipv6 router eigrp 64512
Router(config-rtr)# no shutdown
```

EIGRP Named Mode

EIGRP Named Mode

- The traditional way to configure EIGRP requires various parameters to be configured under the interface and EIGRP configuration mode
 - In order to configure EIGRP IPV4 and IPv6, it is required to configure separate EIGRP instances
 - Traditional EIGRP does not support Virtual Routing and Forwarding (VRF) in IPv6 EIGRP implementations
- With Named mode EIGRP, everything is configured at a single place under the EIGRP configuration and there are no restrictions as mentioned previously
- Supported in
 - 15.0(1)M
 - 12.2(33)SRE
 - 12.2(33)XNE
 - Cisco IOS XE Release 2.5
- Preferred mode of configuring EIGRP whenever IOS supports it
 - All commands for new features in EIGRP will be made available in the named mode only.

EIGRP Named Mode – IPv4

- Choose an arbitrary name

```
R1 (config)# router eigrp TEST
```

- IPv4 EIGRP address family for AS number 1 is enabled

```
R1 (config-router)#  
address-family ipv4 unicast autonomous-system 1
```

- Set Router-id

```
R1 (config-router-af)# eigrp router-id 1.1.1.1
```

- Set networks

```
R1 (config-router-af)# network 192.168.2.0 0.0.0.3
```


EIGRP Named Mode – IPv6

- IPv6 address family can be configured similarly:

```
R1 (config-router) #  
    address-family ipv6 unicast autonomous-system 1
```

- **Note!** EIGRP is immediately enabled on all interfaces with IPv6 address!
- You can override the behavior by setting your own defaults:

```
R1 (config-router-af) # af-interface default  
R1 (config-router-af-interface) # shutdown  
R1 (config-router-af) # af-interface FastEthernet0/1  
R1 (config-router-af-interface) # no shutdown
```

Named Mode Sections

▪ Address Family (AF) section

- mandatory section, configuration that specifies the particular address family for which an EIGRP instance shall be started

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

▪ Per-AF-interface section

- optional section, settings pertaining to the specified interface and AF

```
(config-router-af) # af-interface IFACE  
(config-router-af-interface) #
```

▪ Per-AF-topology section

- inside a particular AF, related to the support of Multi Topology Routing

```
(config-router-af) # topology base  
(config-router-af-topology) #
```

Traditional vs. Named Mode

Traditional EIGRP configuration

```
Interface Ethernet0/0
ip address 10.10.10.1
ip hello eigrp 1 30
ipv6 enable
ipv6 enable eigrp 1
ipv6 bandwidth-percent eigrp 1 40
```

```
router eigrp 1
network 10.0.0.0 255.0.0.0
```

```
address-family ipv4 vrf savage
autonomous-system 65534
network 192.168.0.0
```

```
ipv6 router eigrp 1
no shutdown
```

*no support for ipv6 vrf

EIGRP Named mode configuration

```
Interface Ethernet0/0
ip address 10.10.10.1
ipv6 enable
!
```

```
router eigrp TEST
address-family ipv4 autonomous-system 1
network 10.0.0.0 255.0.0.0
af-interface Ethernet0/0
hello 30
exit-af-interface
!
```

```
address-family ipv4 vrf savage autonomous-system 65534
network 192.168.0.0
!
```

```
address-family ipv6 autonomous-system 1
af-interface Ethernet0/0
no shutdown
bandwidth-percent 40
exit-af-interface
!
```

```
address-family ipv6 vrf TEST autonomous-system 1
af-interface Ethernet0/0
no shutdown
exit-af-interface
```

Conversion To Named Mode

■ Snippet

```
router eigrp 1
  network 10.10.10.1 0.0.0.0
!
interface Ethernet0/0
  ip address 10.10.10.1
  255.255.255.0
  ip hello-interval eigrp 1 100
```

```
R1(config)# router eigrp 1
R1(config-router)#
  eigrp upgrade-cli TEST
Configuration will be converted
from router eigrp 1 to router
eigrp TEST.
Are you sure you want to
proceed? ? [yes/no]: yes
*Oct 10 14:14:40.684: EIGRP:
Conversion of router eigrp 1 to
router eigrp TEST - Completed.
```

■ Converted

```
router eigrp TEST
!
  address-family ipv4 unicast
  autonomous-system 1
!
  af-interface Ethernet0/0
  hello-interval 100
  exit-af-interface
!
  topology base
  exit-af-topology
  network 10.10.10.1 0.0.0.0
  exit-address-family
```

Authentication in EIGRP

Authentication

- EIGRP supports only MD5 authentication
- Configuration guideline is similar to RIPv2 authentication setup

1. Creation of key chain:

```
Router(config)# key chain NAME  
Router(config-keychain)# key NUMBER  
Router(config-keychain-key)# key-string PASSWORD
```

2. Activation of authentication on interface:

```
Router(config-if)# ip authentication mode eigrp AS md5
```

3. Activation of key chain on interface:

```
Router(config-if)# ip authenticat key-chain eigrp AS NAME
```

Authentication

- If Named Mode is used:

```
router eigrp TEST
!
address-family ipv6 unicast autonomous-system 1
!
af-interface GigabitEthernet0/1
  authentication mode md5
  authentication key-chain TEST
exit-af-interface
```

Key Chain Lifetimes ①

```
Router(config-keychain-key) #  
  accept-lifetime start-time {infinite | end-time |  
  duration seconds}
```

- Optional command: defining interval when router is checking received packets signed with this key

```
Router(config-keychain-key) #  
  send-lifetime start-time {infinite | end-time |  
  duration seconds}
```

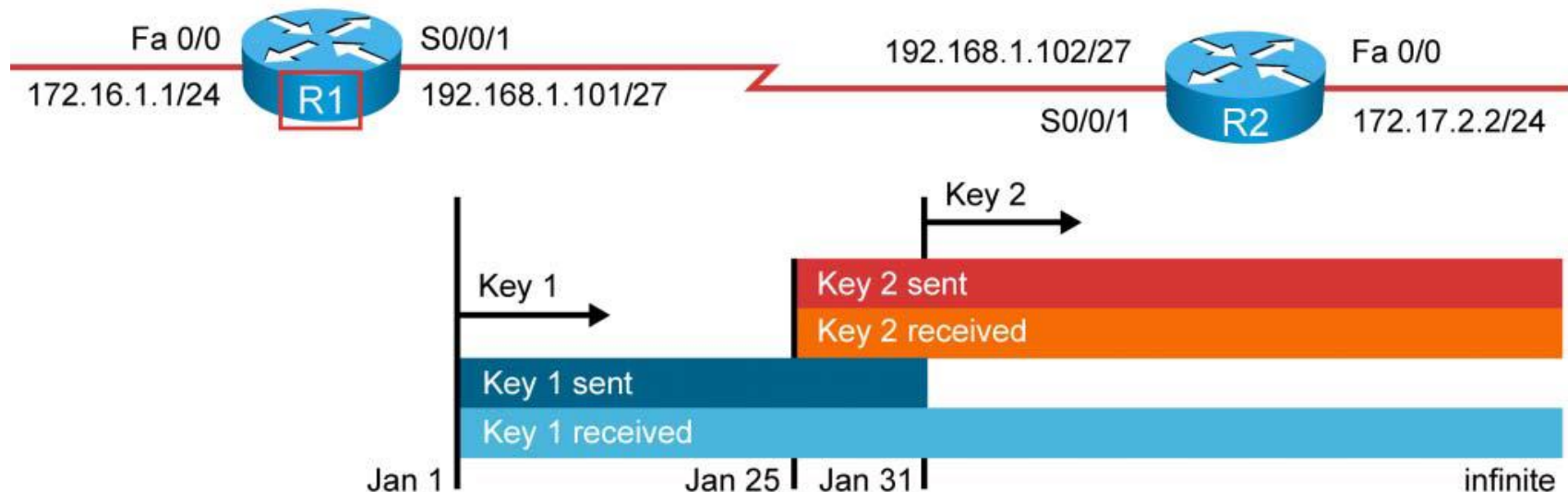
- Optional command: defining interval when router sends packets signed with this key

Key Chain Lifetimes ②

R1(config)#

```
key chain routerR1chain
key 1
  key-string firstkey
  accept-lifetime 04:00:00 Jan 1 2009 infinite
  send-lifetime 04:00:00 Jan 1 2009 04:00:00 Jan 31 2009
key 2
  key-string secondkey
  accept-lifetime 04:00:00 Jan 25 2009 infinite
  send-lifetime 04:00:00 Jan 25 2009 infinite
```

EIGRP AS = 110



Time Server Functionality (NTP)

- NTP client (router which synchronizing time against server) configuration snippet:

```
Router(config)# ntp server IP [prefer]
```

- NTP server (router providing NTP services) snippet:

```
Router(config)# ntp master [1-15]
```

- Timezone configuration:

```
Router(config)# clock timezone CET 1  
Router(config)# clock summer-time CEST recurring last Sun  
Mar 2:00 last Sun Oct 3:00
```

- Verifying command:

```
show ntp association
```

EIGRP Summarization

Summarization ①

- EIGRP supports **automatic** and **manual** summarization
- Same rules as for implementing RIPv2 summarization
 - Takes place when interface in the one major network is sending information about its component to the other major network
 - Beware, it is turned on by default!
 - Immediately when summarization is working **discard route is automatically generated** to Null0 interface
 - Automatic summarization could be done only on components of major networks in which router has directly connected interface – difference between RIPv2 implementation

Summarization ②

- Manual summarization configuration snippet:

```
Router(config-if)# ip summary-address eigrp AS NET MASK  
Router(config-if)# router eigrp AS  
Router(config-router)# no auto-summary
```

- *In case of manual summarization do not forget to turn automatic summarization off!!!*
 - Otherwise both (manual and automatic) routes are sent
- *Turning automatic summarization off is recommended practice as first step in configuring EIGRP*

Summarization ③

- Summarization helps to bound space where queries are advertised
 - Neighbors **do not recognize components** of summary route
 - *What happened when they are asked?*
 - *They answer that they do not know!*
- Hierarchical design is absolutely necessary when implementing EIGRP (especially in case of large topologies)
 - Good summarization **could reduce risk of** so called **Stuck-in-Active State**

EIGRP Stub, SIA-Query, Graceful Shutdown

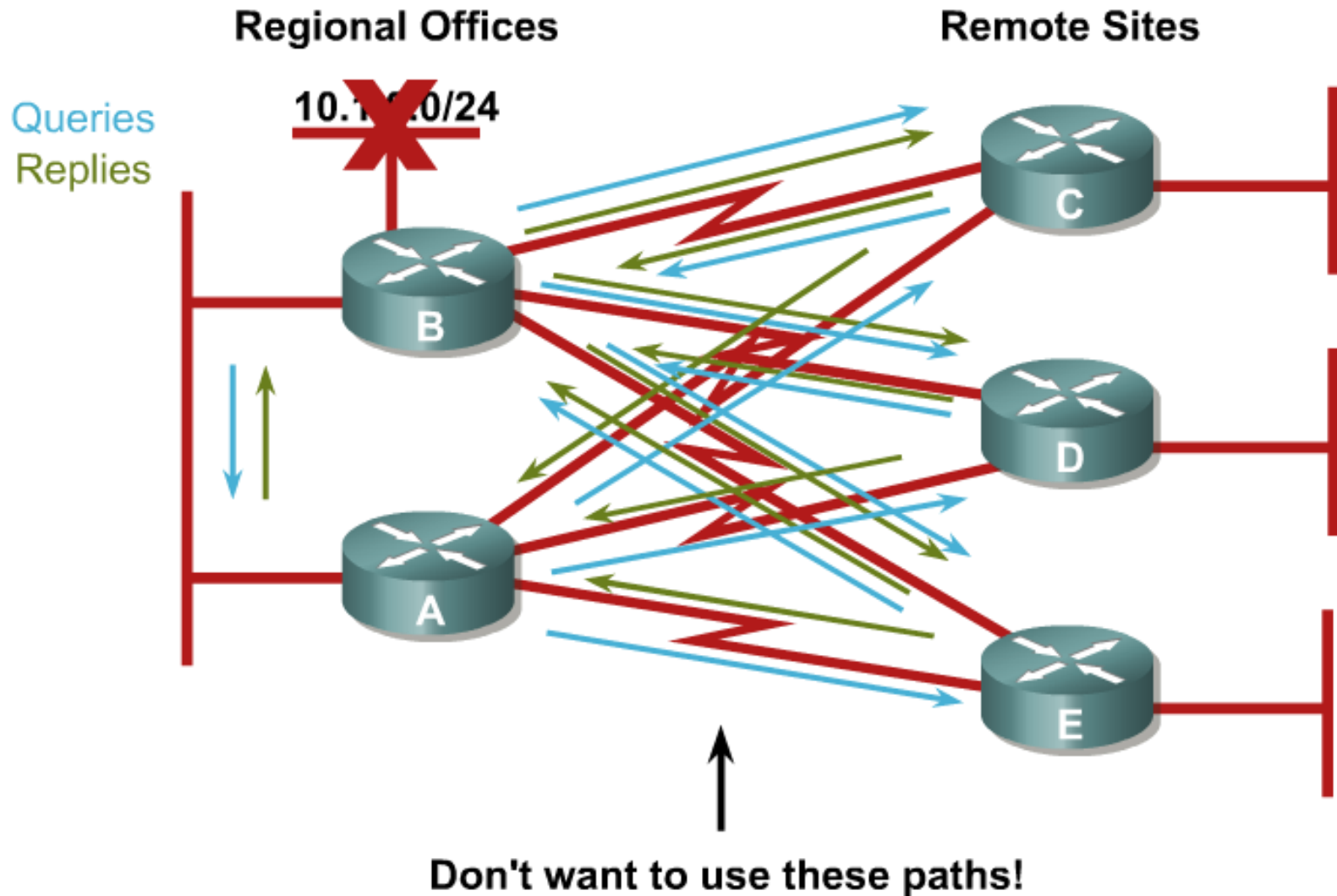
Factors Implementing Convergence

1. Amount of routing information exchanged between neighbors is influenced by good summarization
2. Number of EIGRP routers in topology
3. **Depth of topology** = number of hops between two most distant EIGRP routers which must be able to accept, send or resend routing information
4. Number of all alternate paths to destination network through topology

Queries in EIGRP

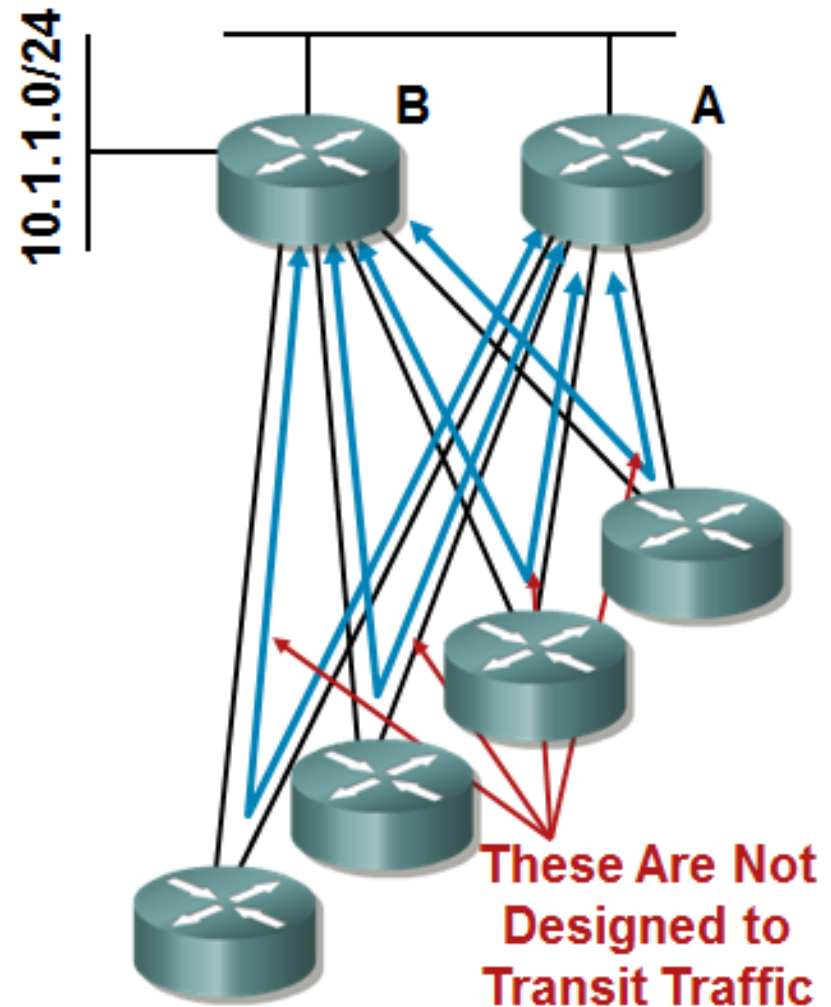
- IF router lost successor and there's no feasible successor in topology table THEN queries are sent to neighbors
- Queries are send to all neighbors through all interfaces
 - Only exception is when Update or Query is received by current successor – in that case router is not querying it back
- Route becomes active by sending query for destination network
- Asked neighbor generates own queries whenever it does not know the answer
- When router replies to query it stops spreading DUAL
 - DUAL can not finish UNTIL router gets all the replies

Queries in Hub-and-Spoke Topology



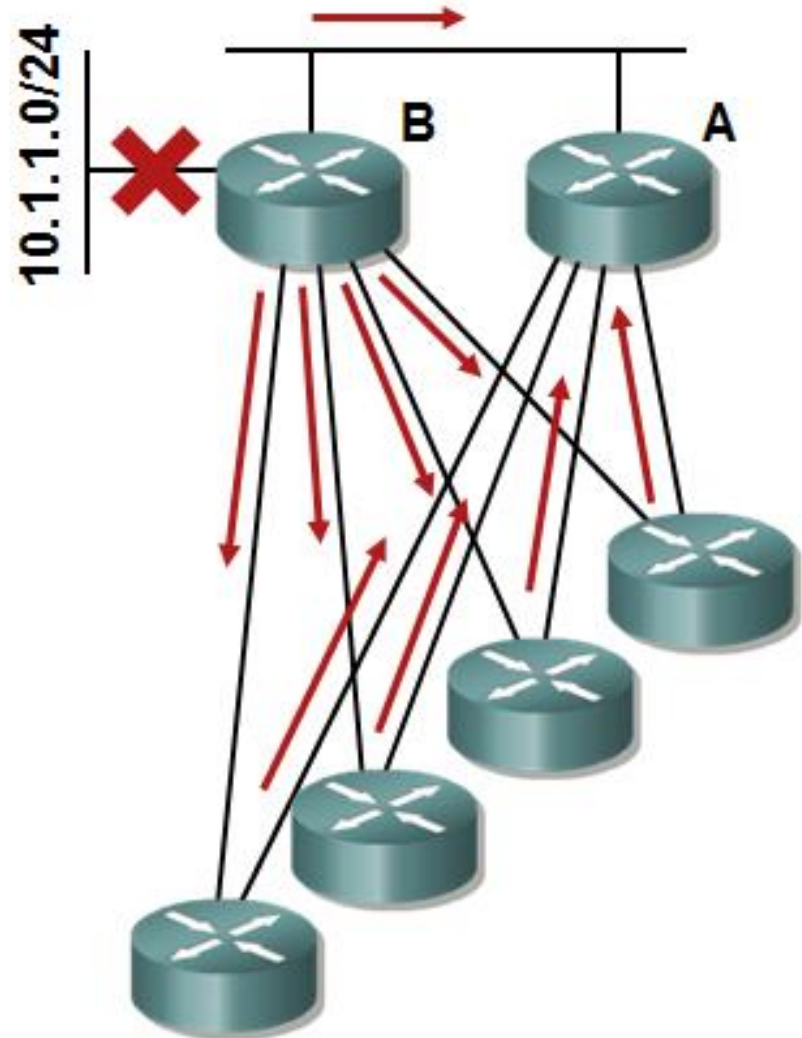
EIGRP Stub ①

- Multiple paths exist in topology to support redundancy of links (not routers)
- *B* should never use spoke routers as transits to networks behind *A*
- *Hence, it is absolutely unnecessary to query spoke routers*



EIGRP Stub ②

1. Whenever *B* lost connection with 10.1.1.0/24 network then it generates and sends five queries to all neighbors
2. Following all spokes query *A* for answer
3. *A* gets five queries on which it must answer
4. All spoke routers will answer initial query from *B*

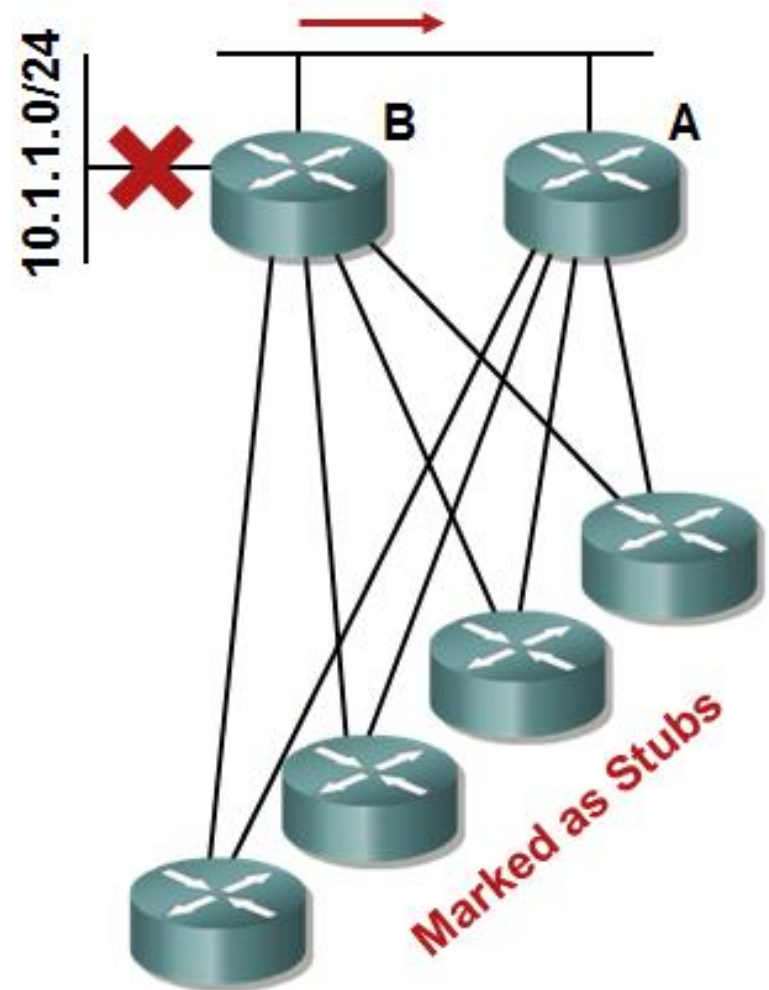


EIGRP Stub ③

- Document: [EIGRP Stub Router Functionality](#)
- **EIGRP stub** is ability of EIGRP to limit topologically unnecessary propagation of queries
- Best used in hub-and-spoke topologies
- Stub router sends EIGRP packets with special TLV record identifying it as stub
- Characteristics of stub configuration:
 - Neighbors of stub routers do not query them
 - By default stub router does not advertised routes learned via EIGRP to its neighbors
 - IF stub receives Query THEN it responds as a matter of principle with “network unreachable”
(Errata exists, it is because of summarization!)

EIGRP Stub ④

- IF spokes are configured as EIGRP stubs THEN routers *A* and *B* never send them Queries
- Whenever router *B* lost connectivity with 10.1.1.0/24 network it sends only one query instead of five
- *A* sees only one path to target network instead of five
 - Spokes do not advertise EIGRP learned routes



EIGRP Stub Configuration ①

```
Router(config-router) #  
eigrp stub [receive-only|connected|static|summary]
```

- **receive-only**: Stub does not inform about any network in its routing table
- **connected**: Stub advertises to EIGRP directly connected networks (together with redistribution)
- **static**: Stub advertises to EIGRP static routes (together with redistribution)
- **summary**: Only summary routes are advertised
- By default **eigrp stub** behaves like:

```
Router(config-router) # eigrp stub connected summary
```

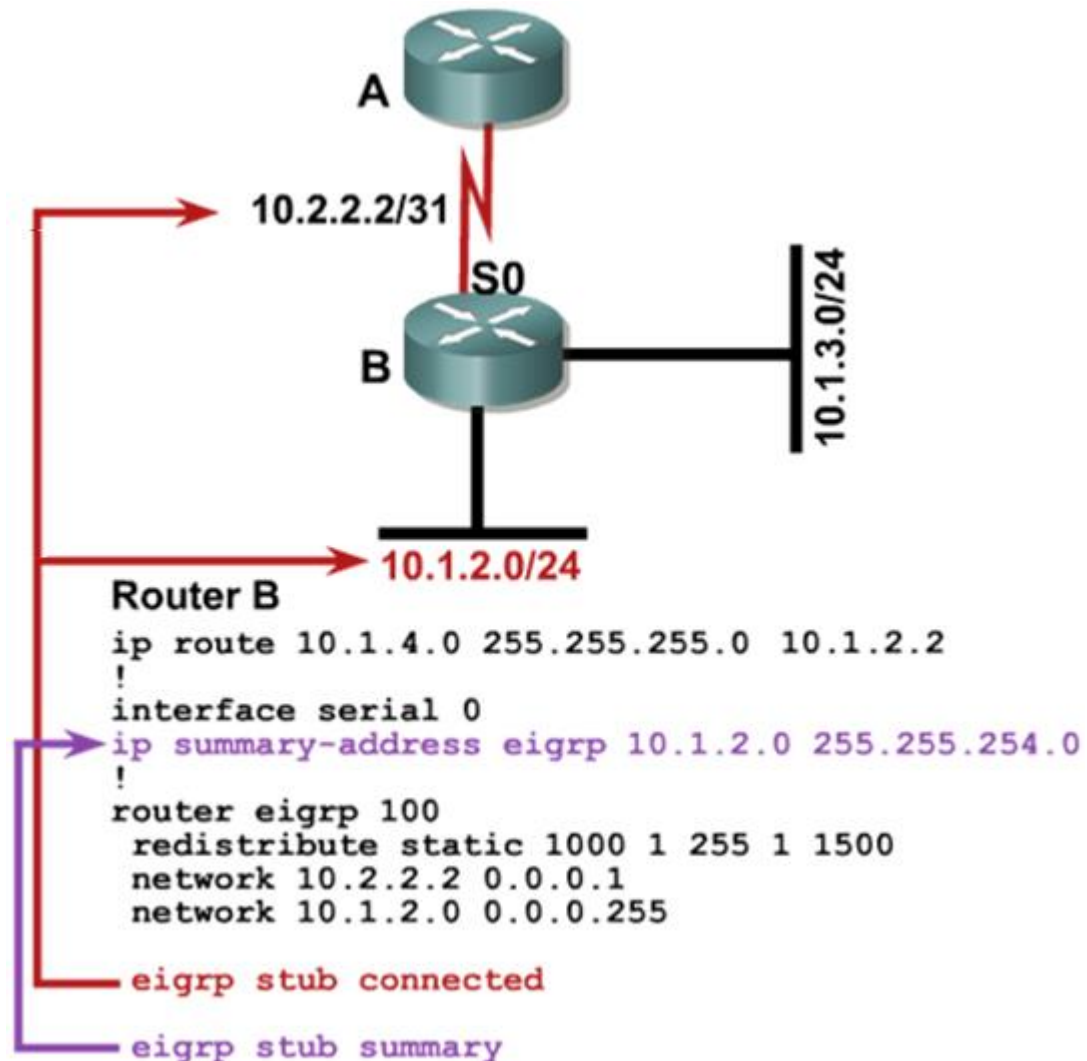
EIGRP Stub Configuration ②

■ stub connected:

- B advertises
10.2.2.1/31,
10.1.2.0/24
- B does not advertise
10.1.2.0/23,
10.1.3.0/24,
10.1.4.0/24

■ stub summary:

- B advertises 10.1.2.0/23
- B does not advertise
10.2.2.1/31,
10.1.2.0/24,
10.1.3.0/24,
10.1.4.0/24



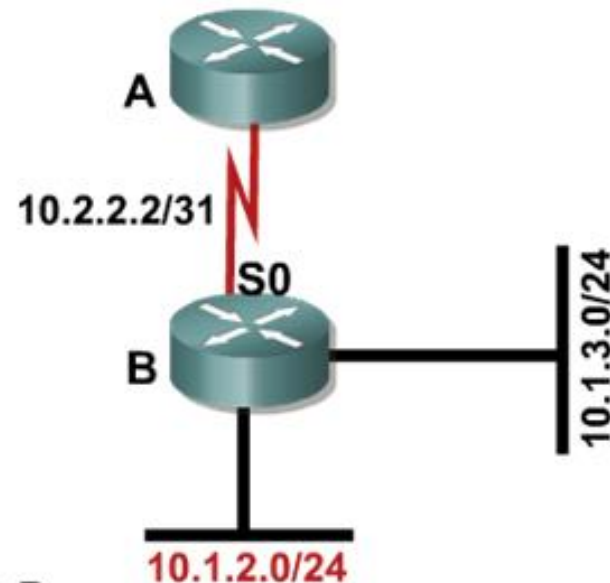
EIGRP Stub Configuration ③

- **stub static:**

- B advertises 10.1.4.0/24
- B does not advertise
10.2.2.1/31,
10.1.2.0/24,
10.1.2.0/23,
10.1.3.0/24

- **stub receive-only:**

- B advertise nothing to A

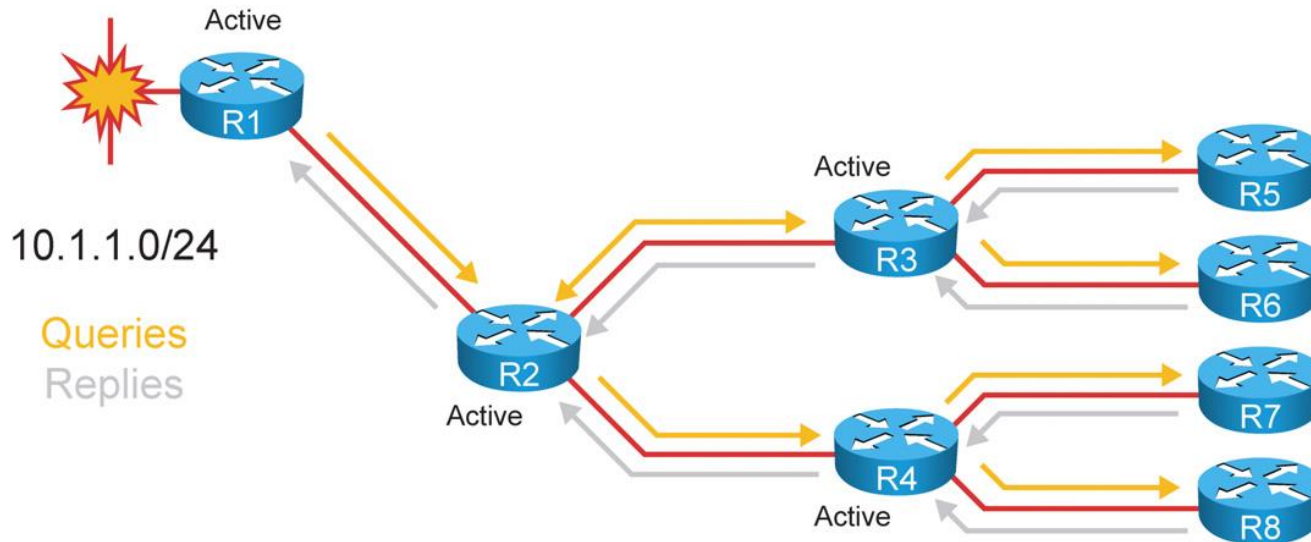


Router B

```
ip route 10.1.4.0 255.255.255.0 10.1.2.2
!
interface serial 0
ip summary-address eigrp 10.1.2.0 255.255.254.0
!
router eigrp 100
→ redistribute static 1000 1 255 1 1500
network 10.2.2.2 0.0.0.1
network 10.1.2.0 0.0.0.255
→ eigrp stub static
eigrp stub summary
```

Stuck-in-Active State

- IF neighbors have target lost route information THEN they answer the query (and stop the query from spreading further) ELSE queries are sent to their neighbors

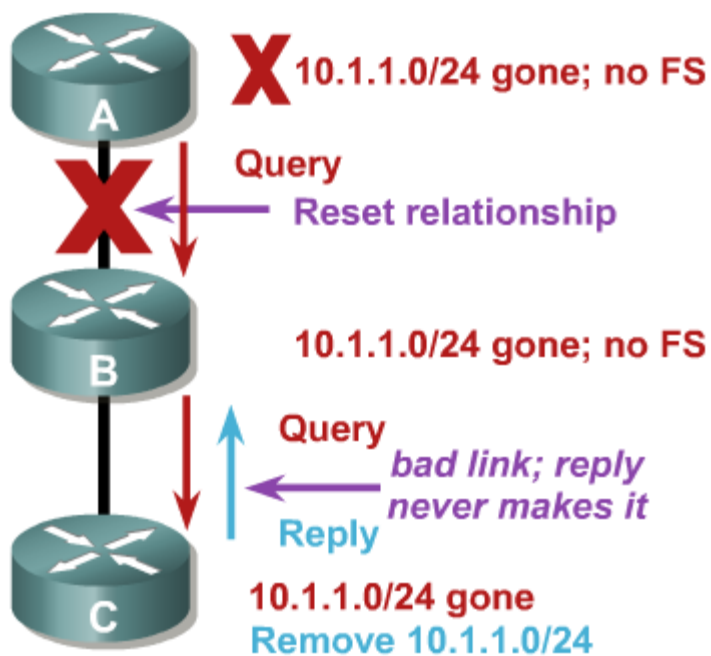


- IF neighbor does not reply to Query packet in **3 minutes limit** THEN the route transits to SIA state AND querying router ends neighbor relationship (is considered down)
- Protection against this is **SIA-Query** described in [Advances in EIGRP](#)

SIA Detection

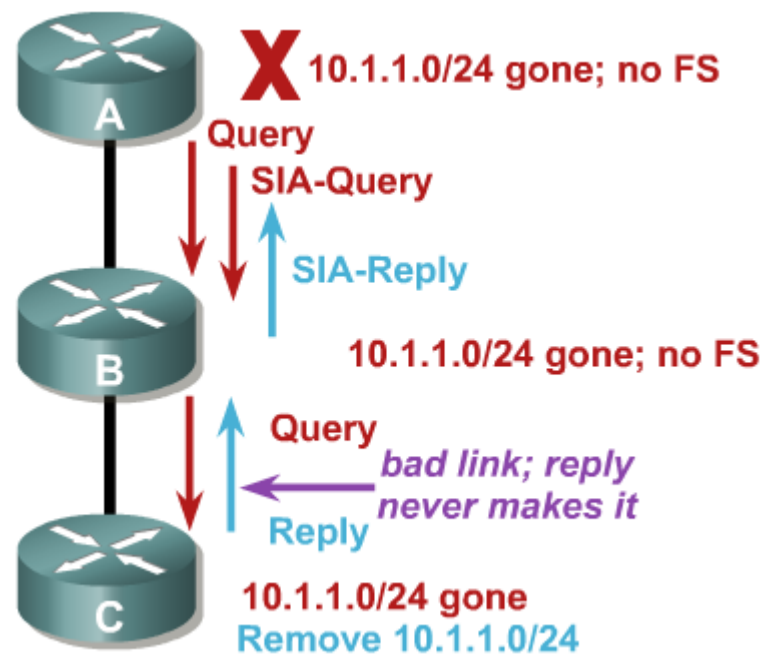
Before

- A resets neighborhood with *B* after 3 minutes timeout even though problem relies on link between *B* to *C*



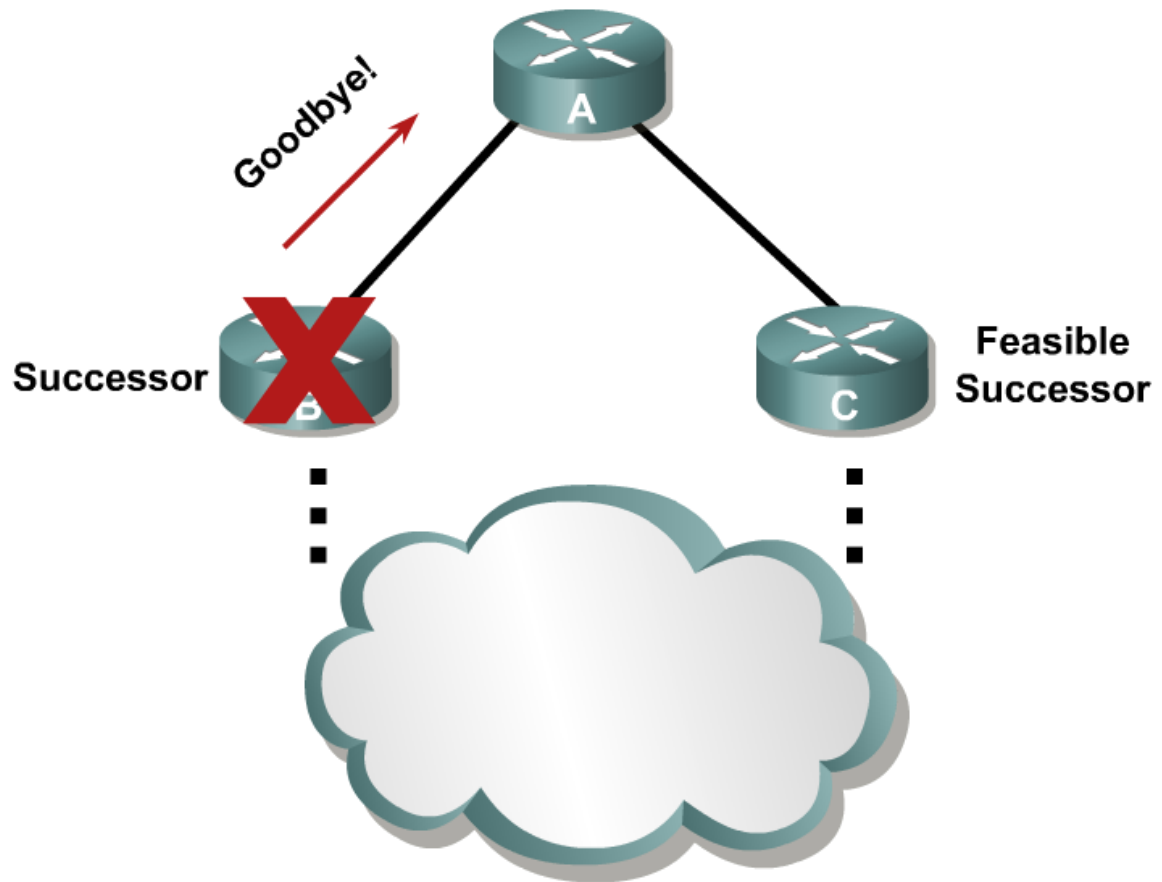
After

- A sends in half of timeout SIA-Query packet to discover whether *B* is still "alive"
- *B* replies with SIA-Reply
- A sends maximally 3× SIA-Queries



EIGRP Graceful Shutdown

- Is ability of router to advertise properly that it is restarting or shutting down
 - In principle it is Hello packet with K-values set to 255
 - Routers without Graceful Shutdown capability reset neighborship also because of K-values mismatch



EIGRP in NBMA Networks

EIGRP in NBMA Networks ①

- Same principle as in case of RIP applies for configuration of EIGRP for NBMA networks because EIGRP is also using multicast communication
 - All directly connected neighbors MUST be defined
 - Together with neighbor IP address also interface MUST be specified where on this interface EIGRP multicast capability is disabled

```
Router(config-router) # neighbor IP IFACE
```

- Split-horizon is by default enabled on all interfaces
- In case of need it could be disabled with command:

```
Router(config-if) # no ip split-horizon eigrp AS
```

EIGRP in NBMA Networks ②

- [Document ID: 13672 \(Configuration Notes for the Implementation of EIGRP over Frame Relay and Low Speed Links\)](#)
- With Frame-Relay it is mandatory to setup correct **bandwidth** on interface
 - Risk of congestion or SIA states...*hence, once again never use bandwidth for manipulating with metric!*
 - By default EIGRP could reserve up to 50% of bandwidth for EIGRP signal communication
 - These percentage could be changed with command **ip bandwidth-percent** on interface
- Three basic rules:
 1. EIGRP traffic passing any interface CAN NOT consume whole bandwidth
 2. Sum of all EIGRP traffic via all virtual circuits on one physical interface CAN'T exceed its overall capacity
 3. Allowed bandwidth consumption of EIGRP traffic MUST be same on both ends of virtual circuits

EIGRP in NBMA Networks ③

- Point-to-Point bandwidth configuration:
 - Set **bandwidth** on CIR speed for target DLCI circuit
- Point-to-Multipoint bandwidth configuration:
 - Set **bandwidth** on value which is result of equation

the lowest CIR · number of DLCIs

- IF result exceeds capacity of physical interface THEN EIGRP consumption percentage have to be changed
- Available bandwidth is equally divided by all VCs

Useful Commands

```
show ip protocols
```

```
show ip eigrp neighbors [detail]
```

```
show ip eigrp topology [all-links] [NETWORK]
```

```
show ip eigrp interfaces [detail]
```

```
show ip eigrp traffic
```

```
show ip route eigrp
```

```
show key chain
```

```
debug eigrp ...
```

```
debug ip eigrp ...
```



Slides adapted by [Vladimír Veselý](#) and [Matěj Grégr](#) partially from official course materials but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

Last update: 2017-09-25

Several notes about FC, DUAL, etc. by Peter Palúch

Diffusing computations

Difúzne výpočty

- V roku 1979, Edsger W. Dijkstra a Carel S. Scholten navrhli pojem „diffusing computation” – difúzny výpočet
 - Dokument [EWD687a](http://www.cs.utexas.edu/users/EWD/), <http://www.cs.utexas.edu/users/EWD/>
 - Difúzny výpočet je distribuovaný výpočet v sieti, pri ktorom jeden uzol prideli časti úlohy na spracovanie svojim susedom
 - Susedia môžu následne podčasti prideleného jobu delegovať ďalej na svojich vlastných susedov
 - Výpočet sa týmto „rozptýli“ – difunduje medzi uzlami
 - Kľúčový problém, ktorý Dijkstra a Scholten riešili, bola detekcia ukončenia tohto výpočtu – návrh vhodnej signálnej schémy

Difúzne výpočty

- Dijkstra a Scholten zaviedli pre riadenie difúzneho výpočtu dva druhy správ: žiadosti (**queries**) a odpovede (**replies**)
 - Query sa používa na delegovanie časti výpočtu na suseda
 - Reply sa používa na odovzdanie výsledku výpočtu susedovi
- Využitím len týchto dvoch správ chceme dosiahnuť, že
 - Keď je výpočet úplne ukončený, pôvodný zadávateľ tohto výpočtu bude o ukončení procesu informovaný
 - Ak je pôvodný zadávateľ výpočtu informovaný o ukončení procesu, distribuovaný výpočet je aj skutočne ukončený

Difúzne výpočty

- Dijkstra a Scholten dokázali, že na splnenie uvedených kritérií v difúznom výpočte stačí dodržať nasledujúce pravidlá:
 - Uzol môže poslať najviac toľko odpovedí, koľko žiadostí dostal. Odpovede sa odosielajú tým susedom, ktorí poslali otázku (každá otázka musí byť zodpovedaná)
 - Ak uzol nespúšťa difúzny výpočet, nesmie poslať žiadosť skôr, ako sám dostane žiadosť od svojho suseda
 - Uzol smie poslať svoju poslednú odpoveď až vtedy, keď sám dostal odpovede na všetky svoje odoslané žiadosti
 - Uzol musí svoju poslednú odpoveď odoslať práve tomu susedovi, od ktorého dostal prvú žiadosť

Difúzne výpočty v EIGRP

- EIGRP využíva signálnu schému difúzných výpočtov ako prostriedok na zainteresovanie dotknutej časti siete do hľadania novej najkratšej cesty k danému cieľu
 - Router, ktorý hľadá novú cestu do cieľa, odošle žiadosť svojim susedom. Žiadosť obsahuje aktuálnu vzdialenosť routera od daného cieľa (podľa typu udalosti môže byť zvýšená alebo nekonečná)
 - Ak susedia nie sú ovplyvnení novou vzdialenosťou uvedenou v žiadosti od svojho suseda, oznámia mu v odpovedi iba vlastnú vzdialenosť a odpoveď neprepošlú
 - V opačnom prípade susedia sami pošlú otázku s vlastnou (zvýšenou) vzdialenosťou, čím šíria difúzny výpočet hlbšie do siete, v marginálnom prípade až k samotnému cieľovému smerovaču
 - Keď router dostane odpovede na všetky svoje otázky, môže si z nich vybrať najlepšiu alternatívu, pretože odosielatelia odpovedí už majú svoju vlastnú cestu do cieľa ujasnenú

Feasibility Condition

Feasibility Condition

- Feasibility Condition (FC) je jeden z výsledkov výskumu, ktorý realizoval Jose Joaquin Garcia-Luna-Aceves v SRI International
- IEEE Transactions on Networking, Vol. 1, No. 1, Február 1993, p. 130-141:
 - Loop-Free Routing Using Diffusing Computations
- Účelom FC je poskytnúť postačujúcu podmienku pre výber takého next-hop routera, ktorý s istotou nespôsobí smerovaciu slučku
 - Táto podmienka sa používa vždy, keď si router vyberá next hop
 - Jednou z jej vlastností je, aby si vyžadovala minimálne množstvo dodatočnej informácie, ktorú si router musí pamätať

Source Node Condition

- Garcia navrhol a dokázal platnosť niekoľkých rôzne formulovaných FC
 - Distance Increase Condition (DIC)
 - Current Successor Condition (CSC)
 - Source Node Condition (SNC)
- EIGRP využíva podmienku Source Node Condition:
 - Ak v čase t musí uzol i zmeniť svoj next-hop do cieľa j , môže si vybrať hociktorého suseda, ktorý poskytuje najnižšiu výslednú metriku a ktorého vzdialenosť od j v čase t je ostro menšia ako minimálna hodnota vzdialenosti i od j do času t , inak musí zachovať svoj existujúci next-hop

K dôkazu Source Node Condition

- Použité značenie:

- $D_j^i(t)$ Vzdialenosť uzla i od cieľa j v čase t
- $D_j^{*i}(t) = FD_j^i(t)$ Minimálna hodnota $D_j^i(t)$ do času t
- $D_{jk}^i(t)$ Vzdialenosť uzla k od cieľa j , ako ju pozná uzol i v čase t
- $s_j^i(t)$ Next-hop (successor) uzla i do cieľa j v čase t
- t_k Čas poslednej aktualizácie smerovacej tabuľky na uzle k
- T_k Čas odoslania posledného relevantného update z uzla k
- n_k k -ty uzol v smerovacej slučke
- τ Moment (čas) vzniku smerovacej slučky

- Dôkaz sporom: predpokladáme, že všetky uzly používajú SNC a že v čase τ vznikne smerovacia slučka tým, že uzol i si vyberie uzol n_1 ako svoj next-hop do cieľa j

Dôkaz Source Node Condition

■ Zrejme platí:

$$\forall t \in \langle t_k, \tau \rangle: s_j^{n_k}(t) = n_{k+1}, D_j^{n_k}(t) = D_j^{n_k}(\tau)$$

$$\forall t: D_j^{n_k}(t) \geq D_j^{*n_k}(t)$$

$$t_1 < t_2 \Rightarrow D_j^{*n_k}(t_2) \leq D_j^{*n_k}(t_1)$$

$$T_{k+1} \leq t_k \leq T_k \leq \tau$$

■ Potom:

$$FD_j^i(\tau) = D_j^{*i}(\tau) > D_{jn_1}^i(\tau) = D_j^{n_1}(T_1)$$

$$D_j^{n_1}(T_1) = D_j^{n_1}(t_1) \geq D_j^{*n_1}(t_1) = FD_j^{n_1}(t_1) > D_{jn_2}^{n_1}(t_1) = D_j^{n_2}(T_2) \Rightarrow FD_j^i(\tau) > D_j^{n_2}(T_2)$$

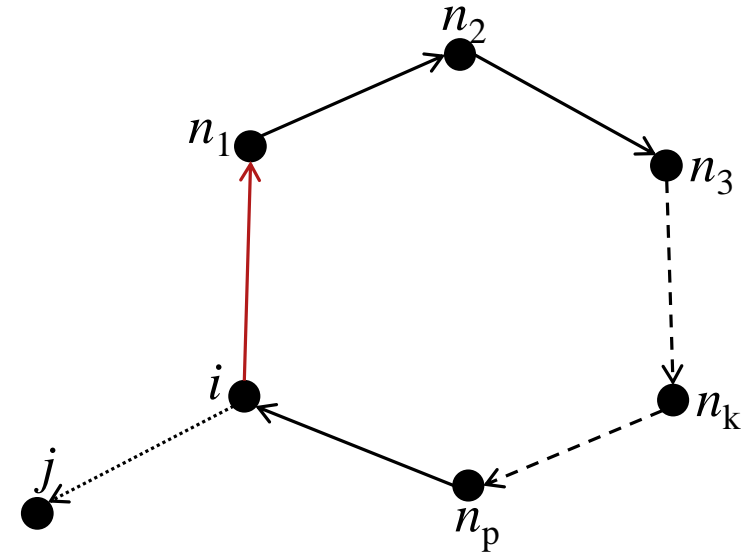
⋮

$$D_j^{n_k}(T_k) = D_j^{n_k}(t_k) \geq D_j^{*n_k}(t_k) = FD_j^{n_k}(t_k) > D_{jn_{k+1}}^{n_k}(t_k) = D_j^{n_{k+1}}(T_{k+1}) \Rightarrow FD_j^i(\tau) > D_j^{n_{k+1}}(T_{k+1})$$

⋮

$$D_j^{n_p}(T_p) = D_j^{n_p}(t_p) \geq D_j^{*n_p}(t_p) = FD_j^{n_p}(t_p) > D_{ji}^{n_p}(t_p) = D_j^i(T_i) \Rightarrow FD_j^i(\tau) > D_j^i(T_i)$$

$$D_j^i(T_i) \geq D_j^{*i}(T_i) \geq D_j^{*i}(\tau) \Rightarrow FD_j^i(\tau) > FD_j^i(\tau)$$



Diffusing Update Algorithm

Diffusing Update Algorithm

- Spojením SNC a difúzných výpočtov je možné vybudovať algoritmus, ktorý hľadá najkratšie cesty a garantuje bezslučkovosť
- Je tu ale problém: predpokladá sa, že počas behu difúzneho výpočtu v sieti nedochádza k ďalším zmenám
 - V praxi je to problém – difúzny výpočet istý čas trvá a v sieti môže počas tohto času dôjsť k ďalším topologickým zmenám
- V tom istom článku „Loop-Free Routing Using Diffusing Computations“, kde je predstavená SNC, Garcia uvádza aj stavový automat DUAL, ktorý popisuje úplné správanie sa difúzneho výpočtu pre všetky možné udalosti, ktoré počas výpočtu môžu nastať

Diffusing Update Algorithm

