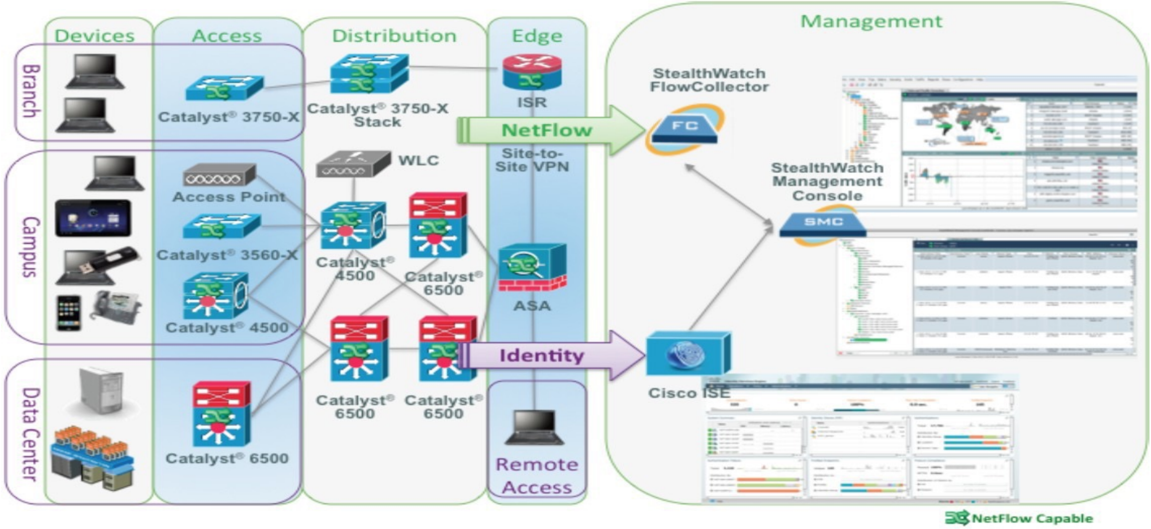
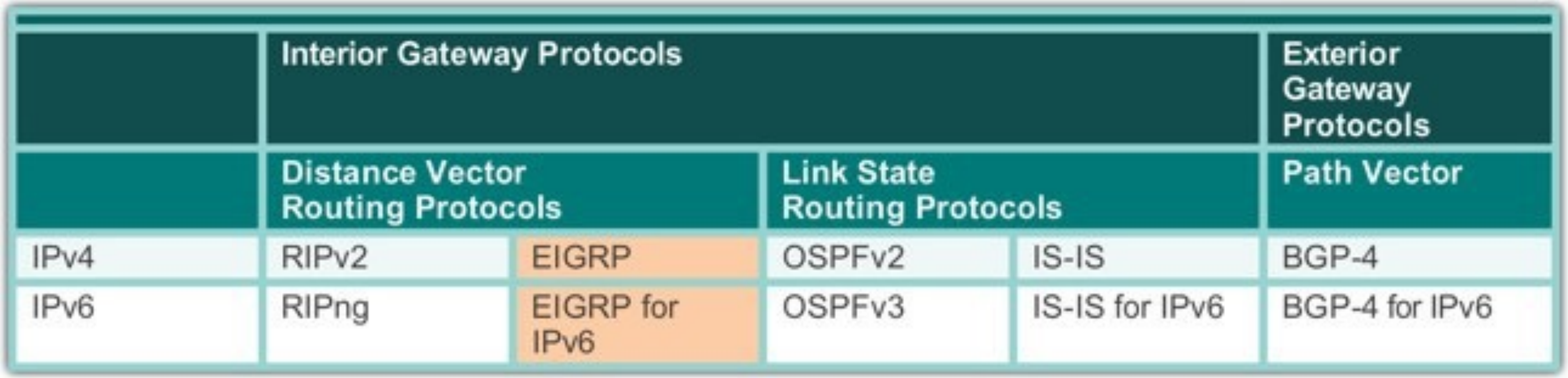
CIS-3250
EIGRP



Dynamic Routing Protocols

Overview of EIGRP



	Interior Gateway Protocols				Exterior Gateway Protocols
	Distance Vector Routing Protocols		Link State Routing Protocols		Path Vector
IPv4	RIPv2	EIGRP	OSPFv2	IS-IS	BGP-4
IPv6	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6	BGP-4 for IPv6

- **Enhanced IGRP** is a *Cisco-proprietary* routing protocol released in 1992.
  - EIGRP was created as a classless version of IGRP.
- EIGRP acts like a link-state routing protocol but is still a distance vector routing protocol.
- In 2013, Cisco released a basic functionality of EIGRP as an open standard to the IETF as an informational RFC.
  - Cisco will continue to maintain control of EIGRP.

Main Components of Routing Protocols

Data Structures

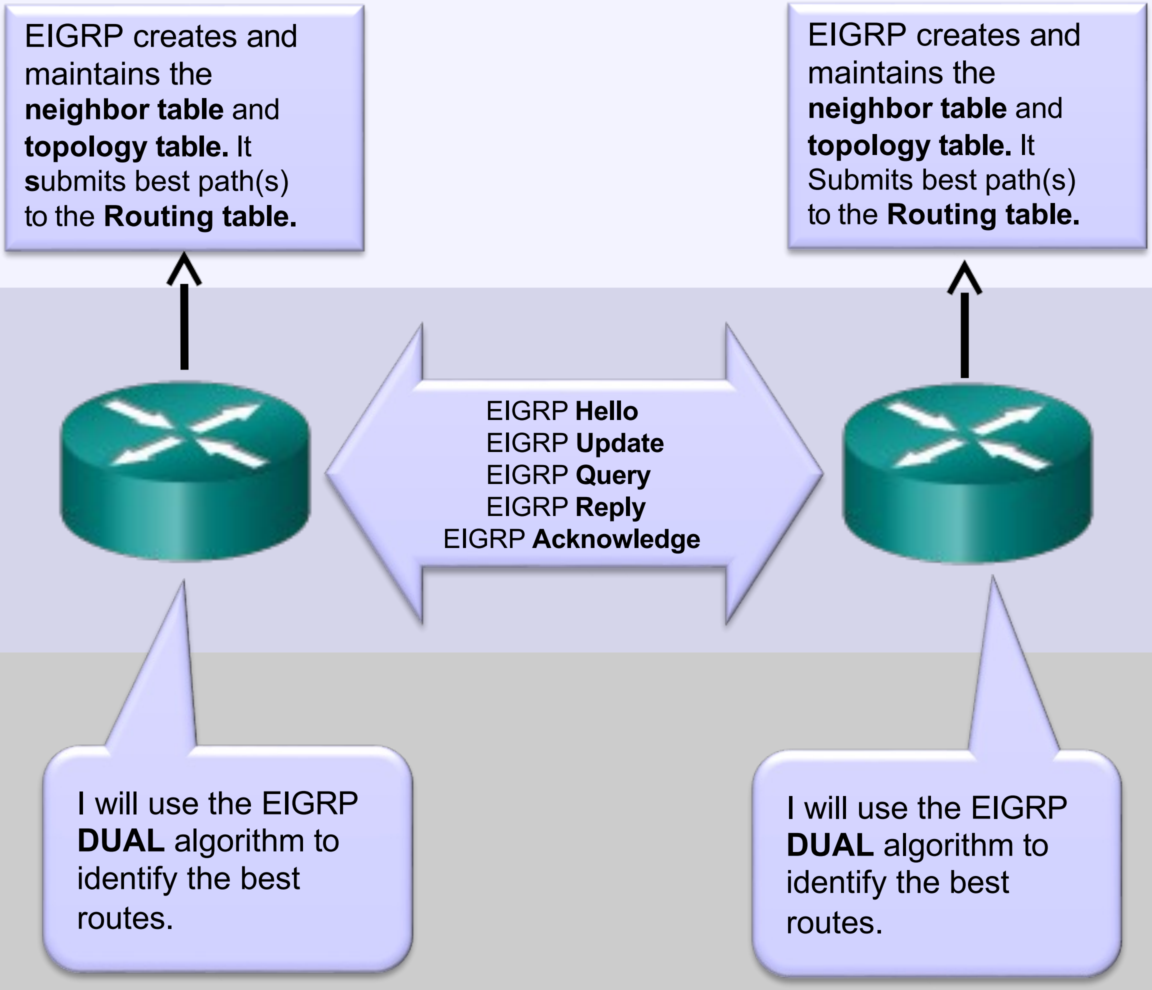
Routing protocols create tables (databases) in RAM to support their operations.

Routing Protocol Messages

Routing protocols use messages to learn and maintain accurate information about the network. Specifically, messages are used to discover neighboring routers, exchange routing information, and other tasks.

Algorithm

Routing protocols use algorithms to determine the best path to various destinations.



EIGRP creates and maintains the **neighbor table** and **topology table.** It **s**ubmits best path(s) to the **Routing table.**

EIGRP creates and maintains the **neighbor table** and **topology table.** It Submits best path(s) to the **Routing table.**

EIGRP **Hello**
EIGRP **Update**
EIGRP **Query**
EIGRP **Reply**
EIGRP **Acknowledge**

I will use the EIGRP **DUAL** algorithm to identify the best routes.

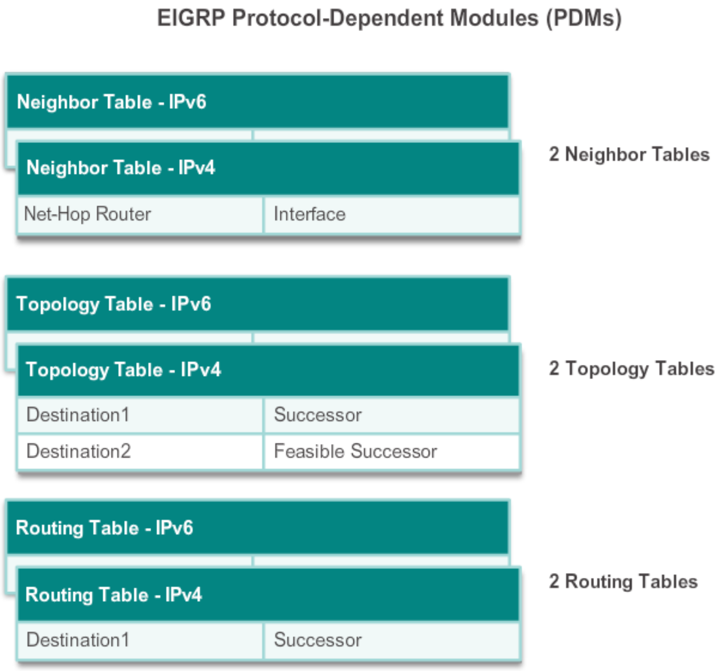
I will use the EIGRP **DUAL** algorithm to identify the best routes.

Features of EIGRP

- Uses Diffusing Update Algorithm (DUAL)
- Does not age out routable entries
- Does not use periodic updates
- Maintains a topology table, separate from the routing table
  - Includes the best paths
  - Loop-free backup paths, if any
- Faster convergence
- Equal and unequal cost load balancing
- Partial and bounded updates:
  - **Partial**: Update only includes information about the route changes
  - **Bounded**: Updates are sent only to those affected routers
- Used Protocol Dependent Modules (PDMs) to support different Layer 3 protocols

EIGRP Key Components

EIGRP Key Components



EIGRP Protocol-Dependent Modules (PDMs)

Neighbor Table - IPv6

Neighbor Table - IPv4	
Net-Hop Router	Interface

2 Neighbor Tables

Topology Table - IPv6

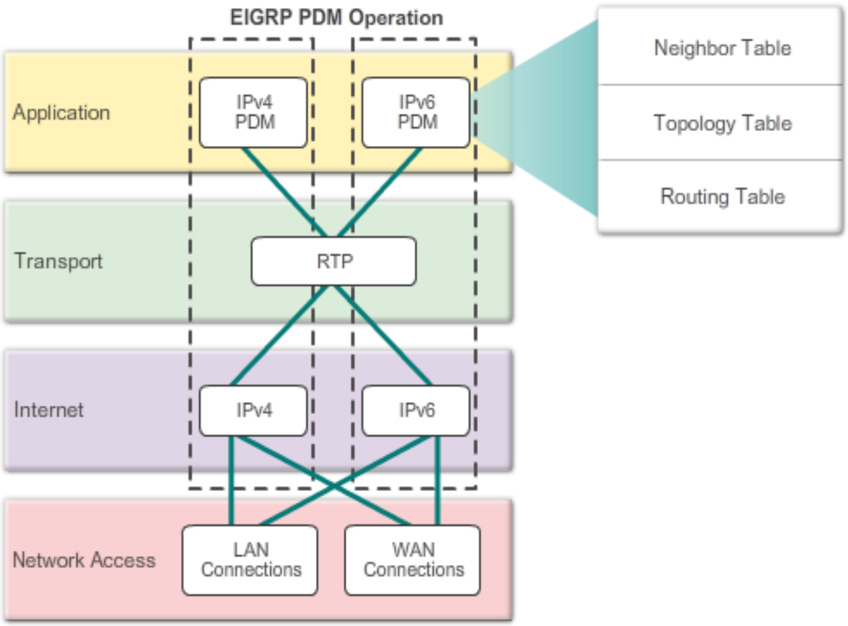
Topology Table - IPv4	
Destination1	Successor
Destination2	Feasible Successor

2 Topology Tables

Routing Table - IPv6

Routing Table - IPv4	
Destination1	Successor

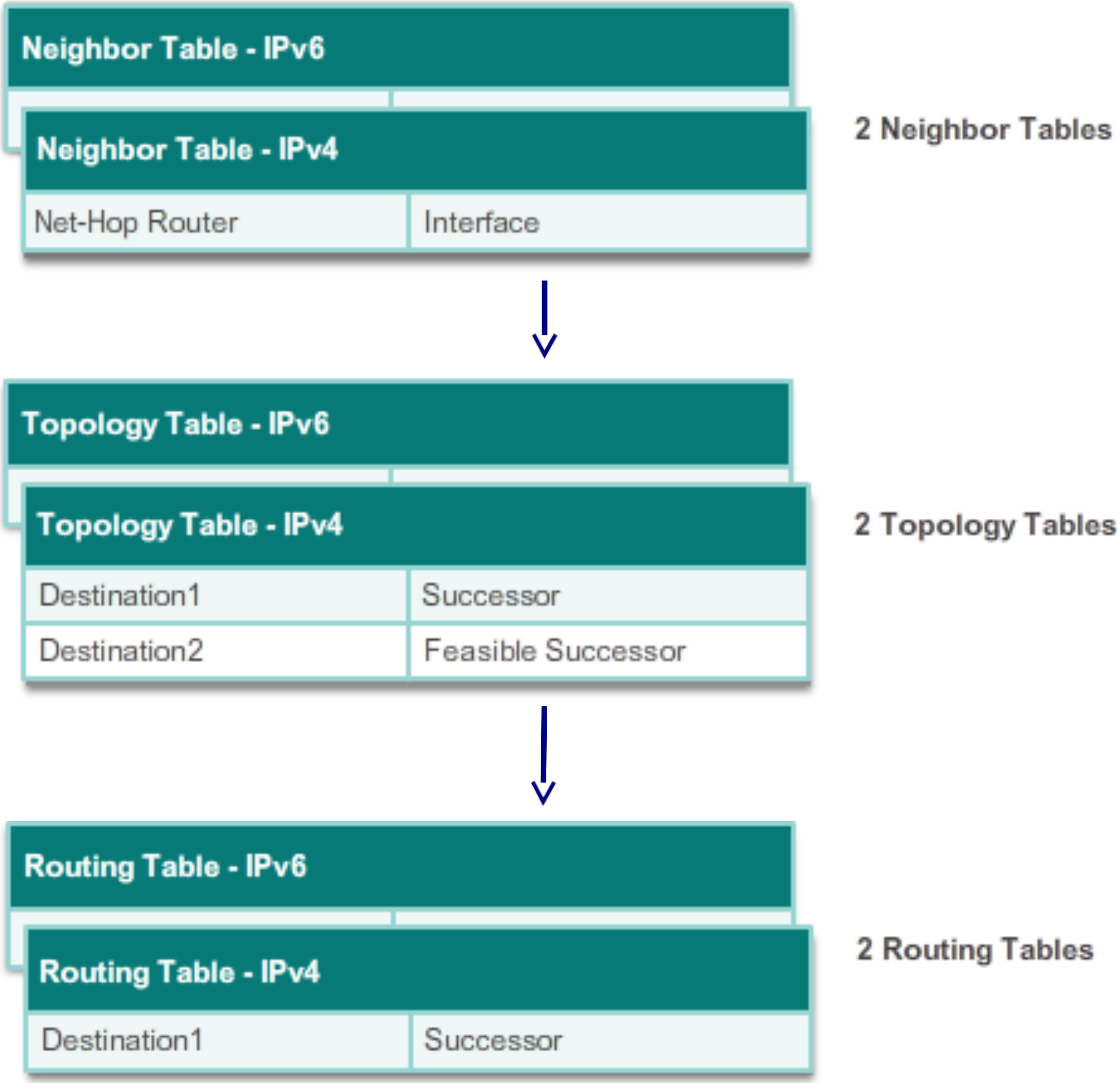
2 Routing Tables



- Protocol-dependent modules (PDMs)
- Reliable Transport Protocol (RTP)
- Neighbor discovery / Recovery (later)
- DUAL finite-state machine (later)

Protocol-Dependent Modules

- EIGRP maintains individual tables for each routed protocol.
- EIGRP uses protocol-dependent modules (PDMs) to support IPv4, IPv6, and legacy protocols IPX and AppleTalk.
- Each PDM is responsible for all functions related to its specific routed protocol.



Neighbor Table - IPv6

Neighbor Table - IPv4

Net-Hop Router	Interface

2 Neighbor Tables

Topology Table - IPv6

Topology Table - IPv4

Destination1	Successor
Destination2	Feasible Successor

2 Topology Tables

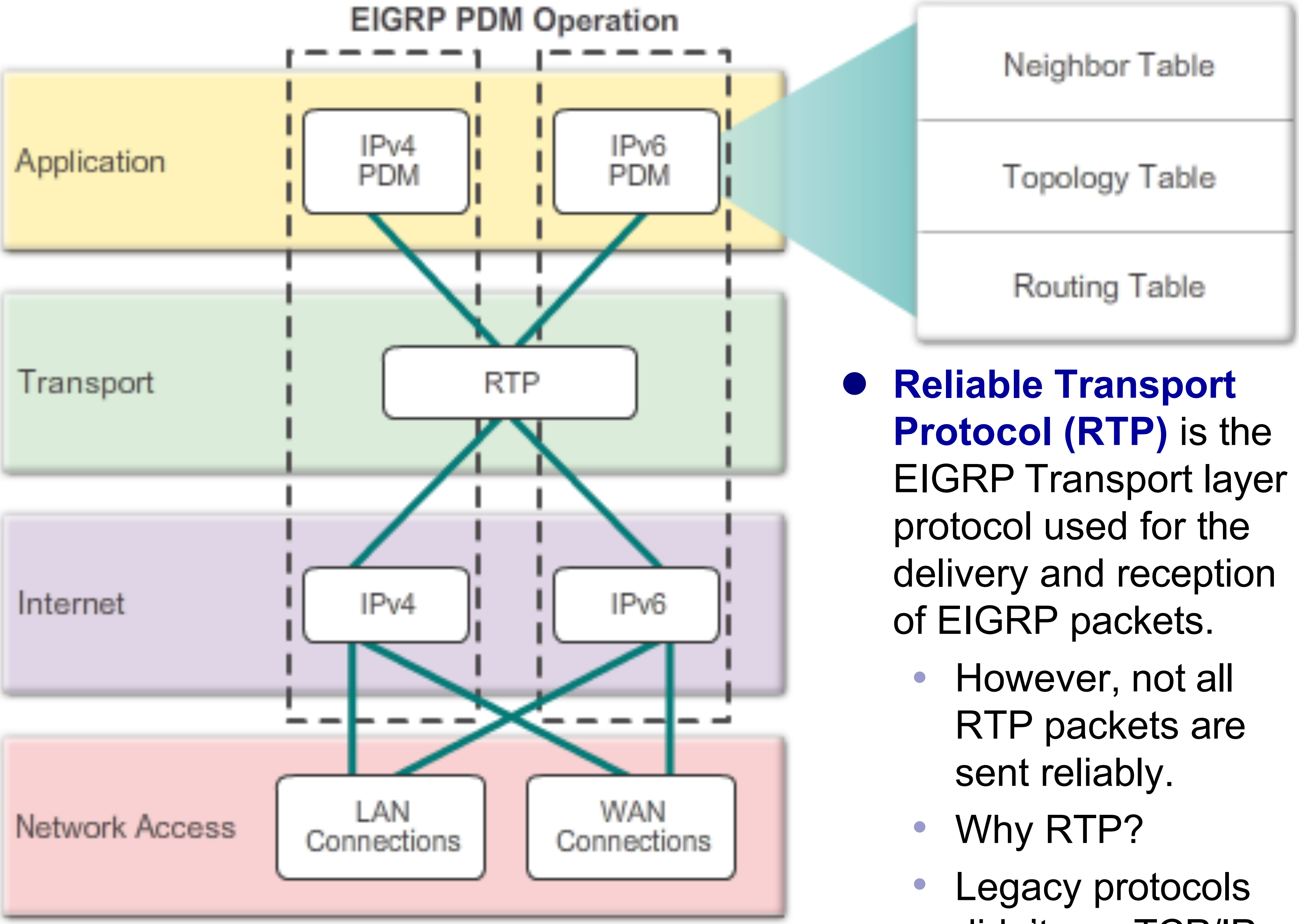
Routing Table - IPv6

Routing Table - IPv4

Destination1	Successor

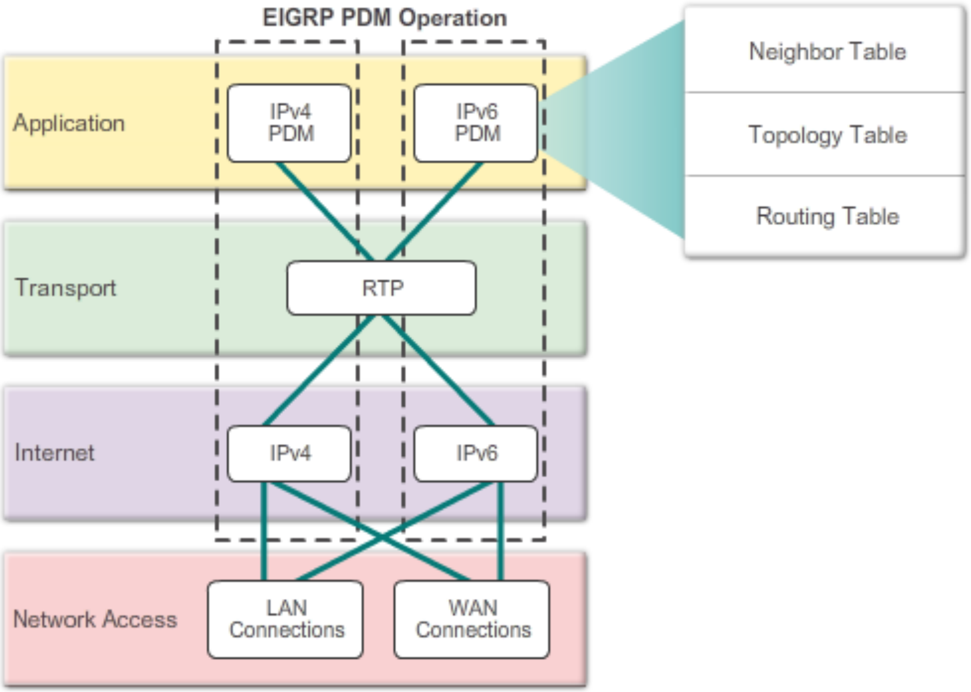
2 Routing Tables

EIGRP RTP



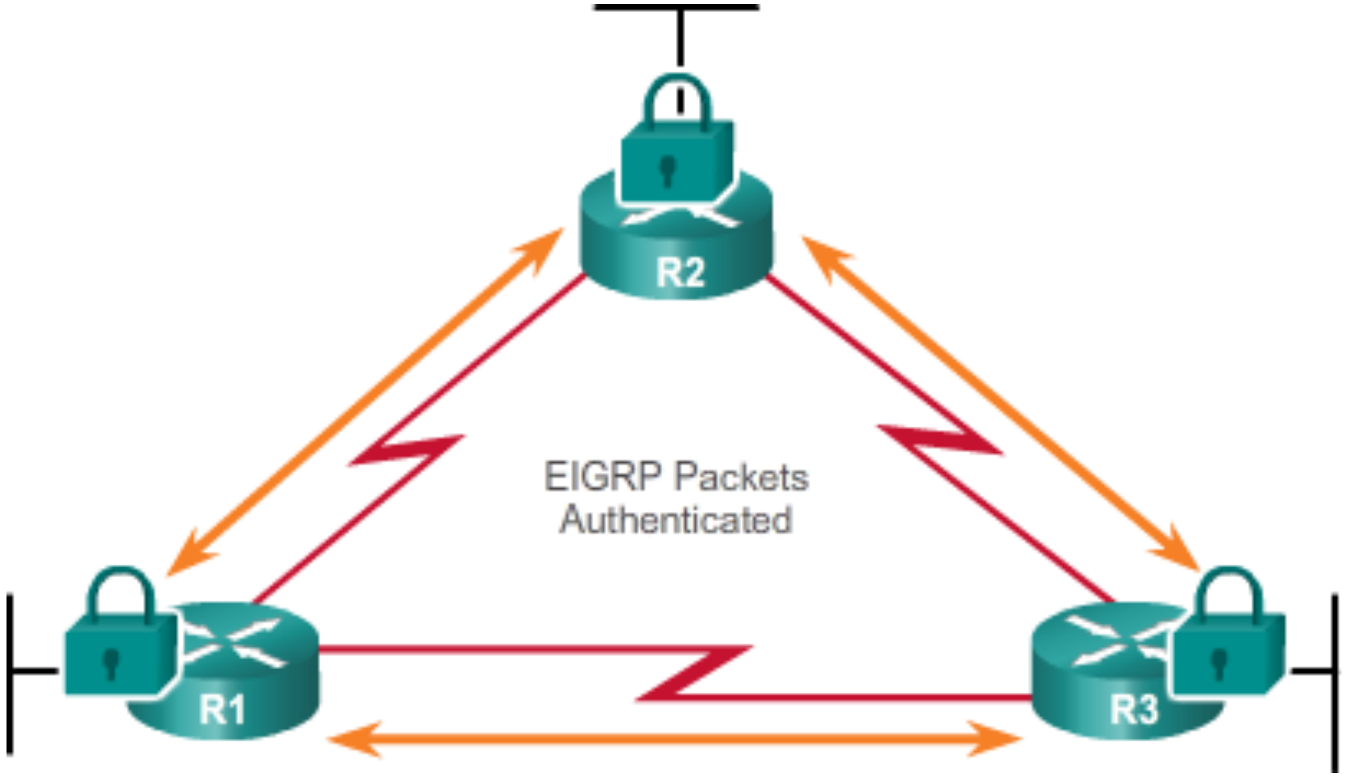
- **Reliable Transport Protocol (RTP)** is the EIGRP Transport layer protocol used for the delivery and reception of EIGRP packets.
  - However, not all RTP packets are sent reliably.
  - Why RTP?
  - Legacy protocols didn’t use TCP/IP.

Reliable Transport Protocol



- **Reliable packets** require explicit acknowledgement from destination
    - Update, Query, Reply
- **Unreliable packets** do not require acknowledgement from destination
    - Hello, ACK
- RTP can send EIGRP packets as unicast or multicast.
    - **IPv4 EIGRP** multicast address **224.0.0.10**.
    - **IPv6 EIGRP** multicast address **FF02::A**.

EIGRP Authentication



- EIGRP can authenticate the routing update source.
  - Ensures router only accepts routing updates from legitimate peers.
- Note:
  - Authentication does not encrypt the EIGRP routing updates.

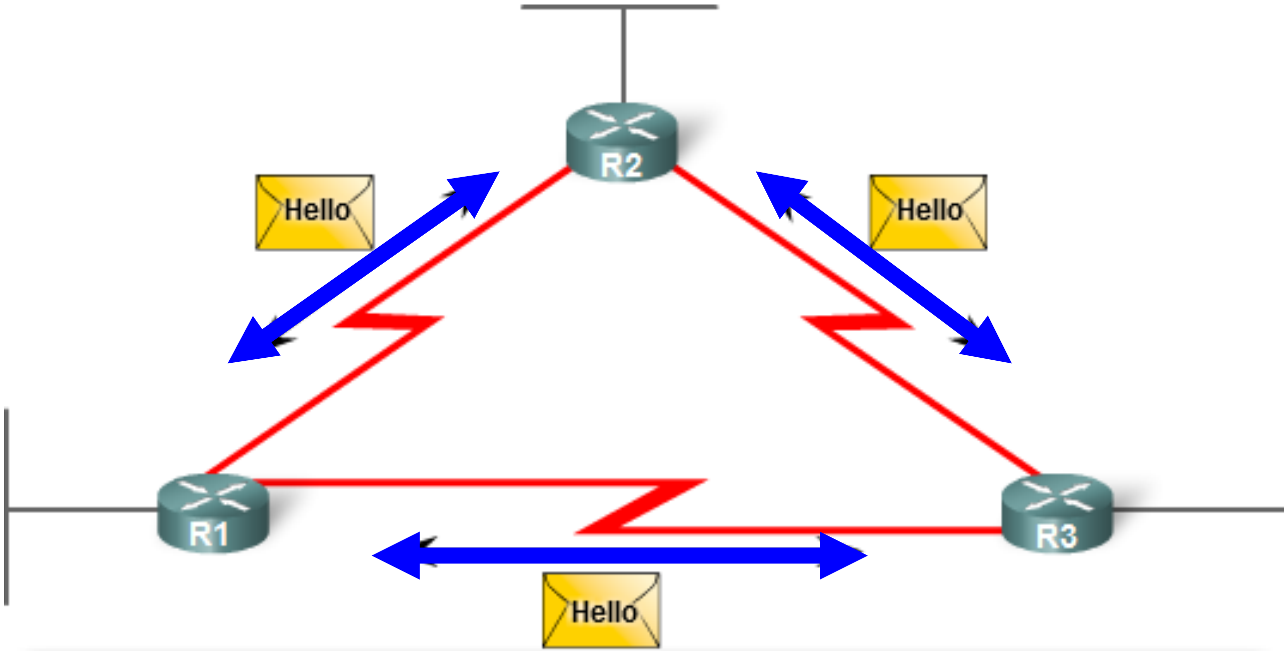
EIGRP Packets

EIGRP Packets

- IP EIGRP relies on 5 types of packets to maintain its various tables and establish complex relationships with neighbor routers.

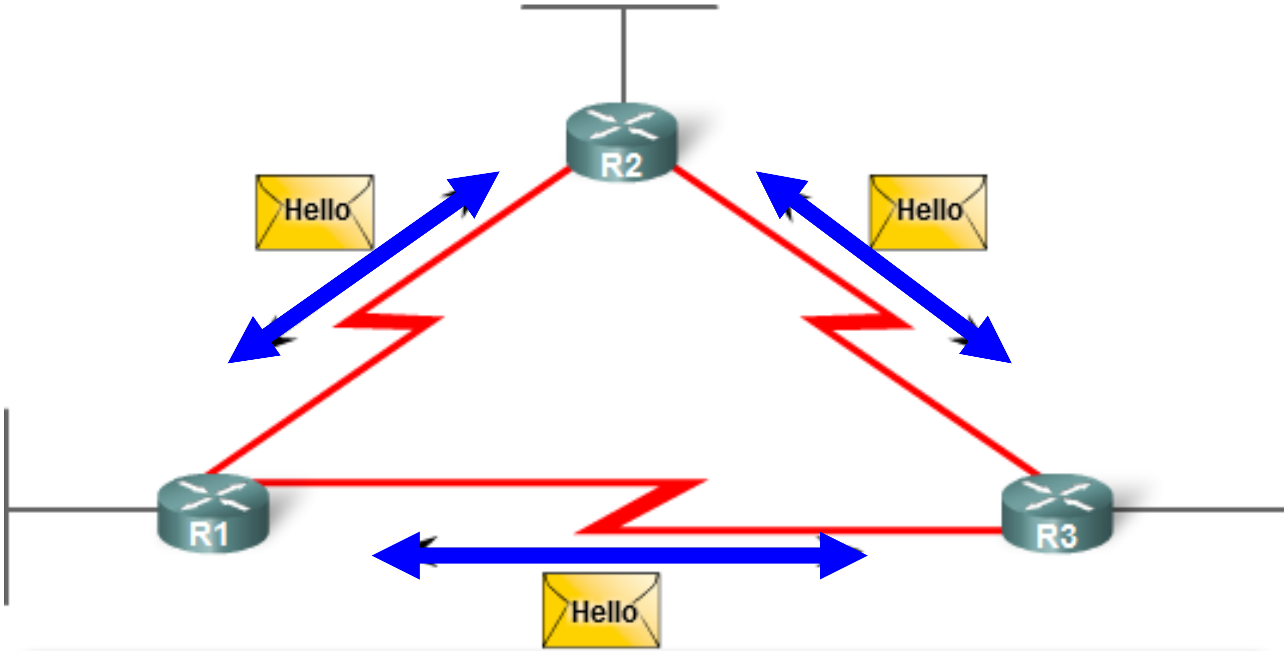
Packet Type	Description
Hello	Used to discover other EIGRP routers in the network.
Acknowledgement	Used to acknowledge the receipt of any EIGRP packet.
Update	Convey routing information to known destinations.
Query	Used to get specific information from a neighbor router.
Reply	Used to respond to a query.

EIGRP Hello Packets



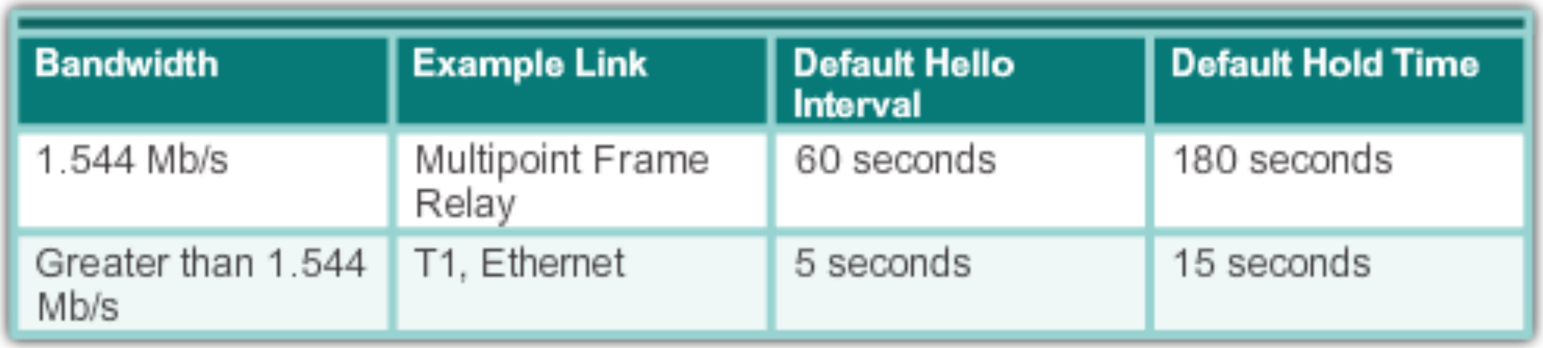
- Hello packets are used to discover & form adjacencies with neighbors.
- Multicasted to:
  - IPv4: **224.0.0.10**
  - IPv6: **FF02::A**
- Hello packets are always sent unreliably.
  - Therefore, Hello packets do not require acknowledgment.

Hello Protocol



- Before any EIGRP packets can be exchanged between routers, EIGRP must first discover its neighbors.
- **EIGRP routers** discover neighbors and establish adjacencies with neighbor routers using the **hello** packet.

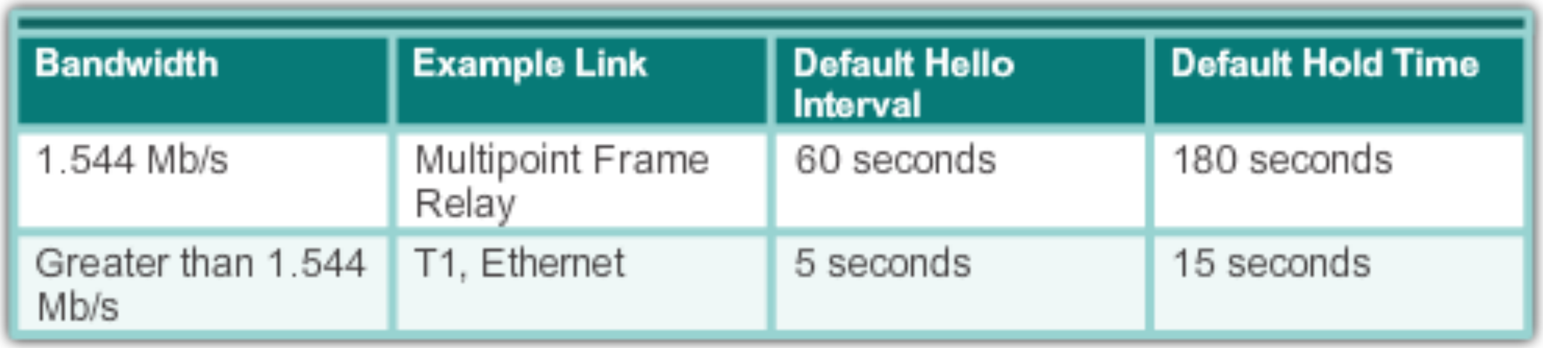
Hello Packets



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

- Hello, packets are sent at regular intervals.
  - The router assumes that the neighbor and its routes remain viable if it receives Hello packets from a neighbor.
- The interval depends on the interface’s bandwidth.
  - **Low Bandwidth = 60 seconds**
    - Default intervals on multipoint nonbroadcast multiaccess networks (NBMA) such as X.25, Frame Relay, and ATM interfaces with access links of T1 (1.544 Mbps) or slower.
  - **High bandwidth = 5 seconds**
    - Default interval on circuits with bandwidth greater than T1 such as Ethernet LANs.

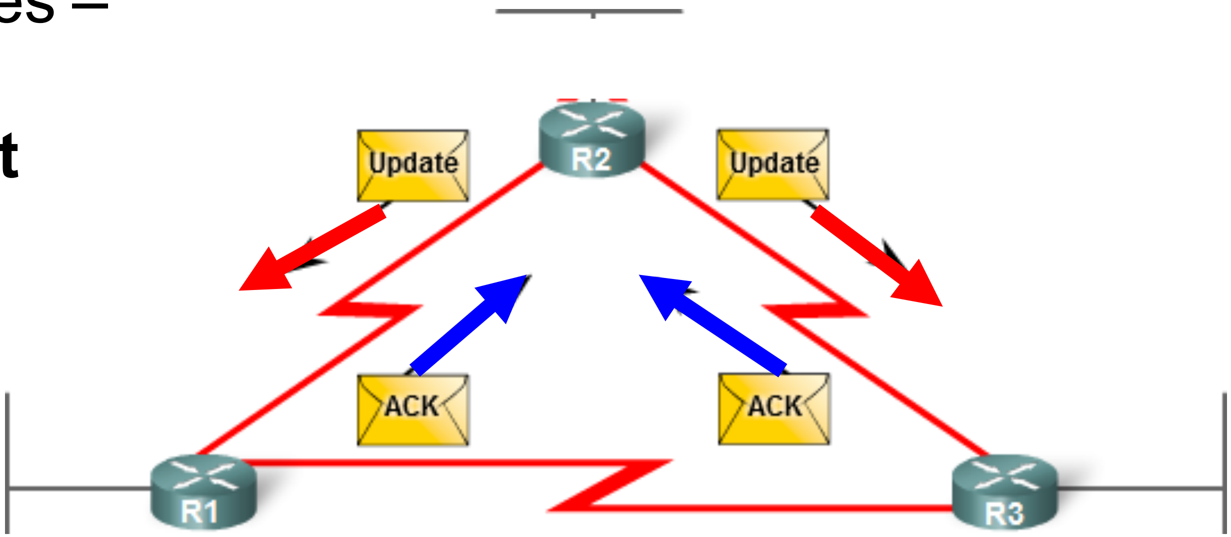
Hello Holdtime



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

- **Hold time** - maximum time the router should wait to receive the next hello before declaring that neighbor as unreachable.
- Default hold time - **3 times the hello interval**
- If the hold time **expires**:
    - EIGRP declares the route as down
    - DUAL searches for a new path in the topology table ***or*** by sending out queries.
- More later.

EIGRP Packet Types – **Update** and **Acknowledgement** Packets



EIGRP uses triggered updates

- **Update Packets**

  Contain only the routing information needed (a change occurs).

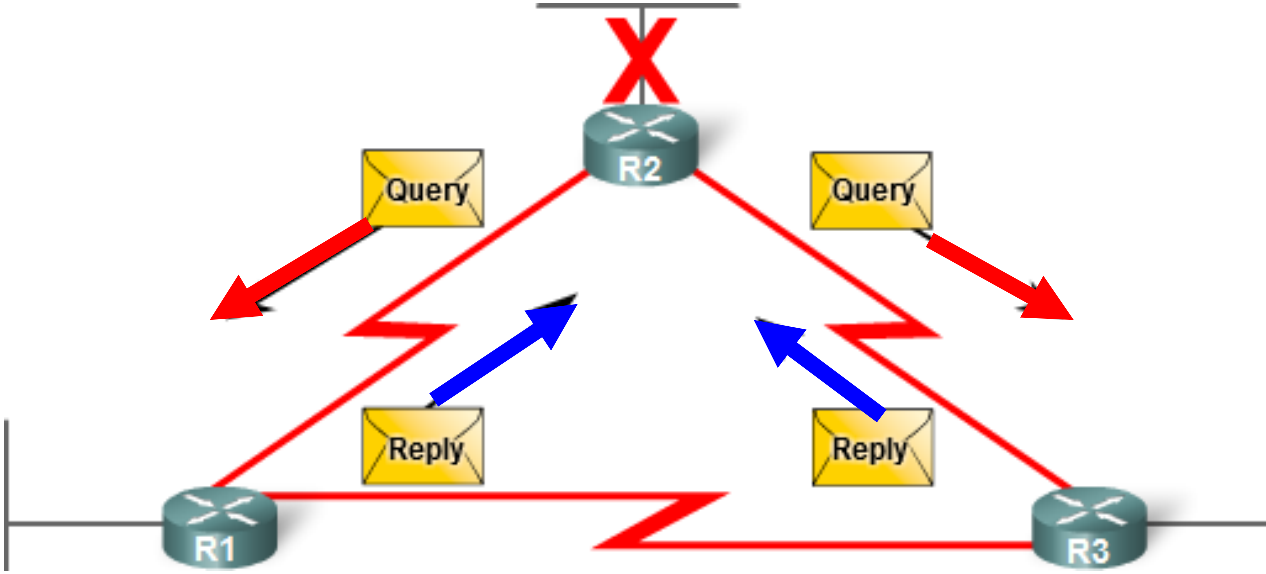
  It is sent only to those routers that require it.

  Uses reliable delivery.

- **Acknowledgment (ACK) Packets**
  - Sent when reliable delivery is used (update, query, and reply packets).
  - Unreliable unicast.

EIGRP Packet Types – **Query** and **Reply** Packets

Why Query? Another router could be attached to the same LAN.



- Used by **DUAL** when searching for networks and other tasks.
- **Queries** and **replies** use reliable delivery.
- **DUAL** is discussed in a later section.
- Queries can use multicast or unicast, whereas Replies are always sent as unicast.

DUAL: An Introduction

J. J. Garcia-Luna-Aceves

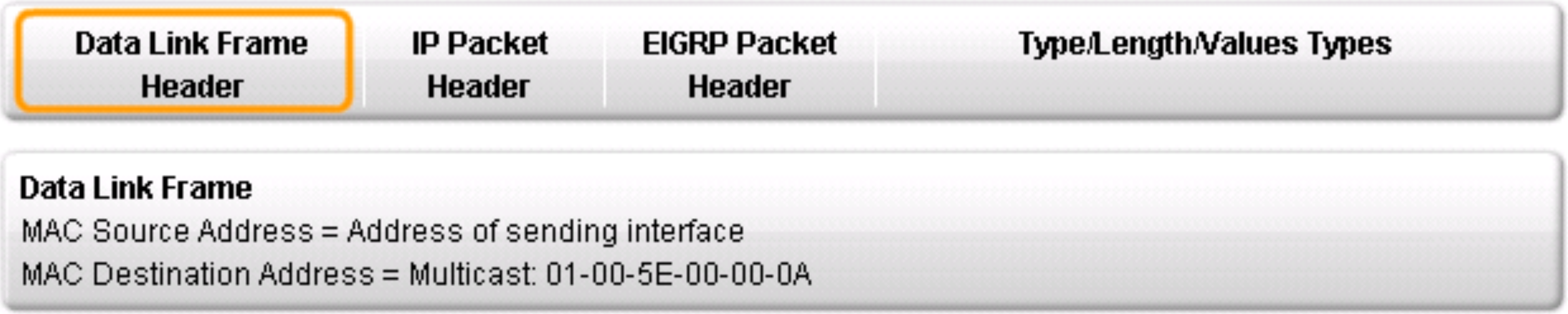
- The **Diffusing Update Algorithm (DUAL)** is the convergence algorithm EIGRP uses.
- First proposed by **E. W. Dijkstra** and **C. S. Scholten**.
- **J. J. Garcia-Luna-Aceves** did the most prominent work with DUAL in the mid-1980s.
- **Distance vector routing protocols** like RIP prevent routing loops with hold-down timers and split horizon.
- Although **EIGRP uses both of these techniques**, it uses them somewhat differently; the primary way that EIGRP prevents routing loops is with the DUAL algorithm.

DUAL: An Overview

Finish me!

EIGRP Message

EIGRP Data Link Header Format



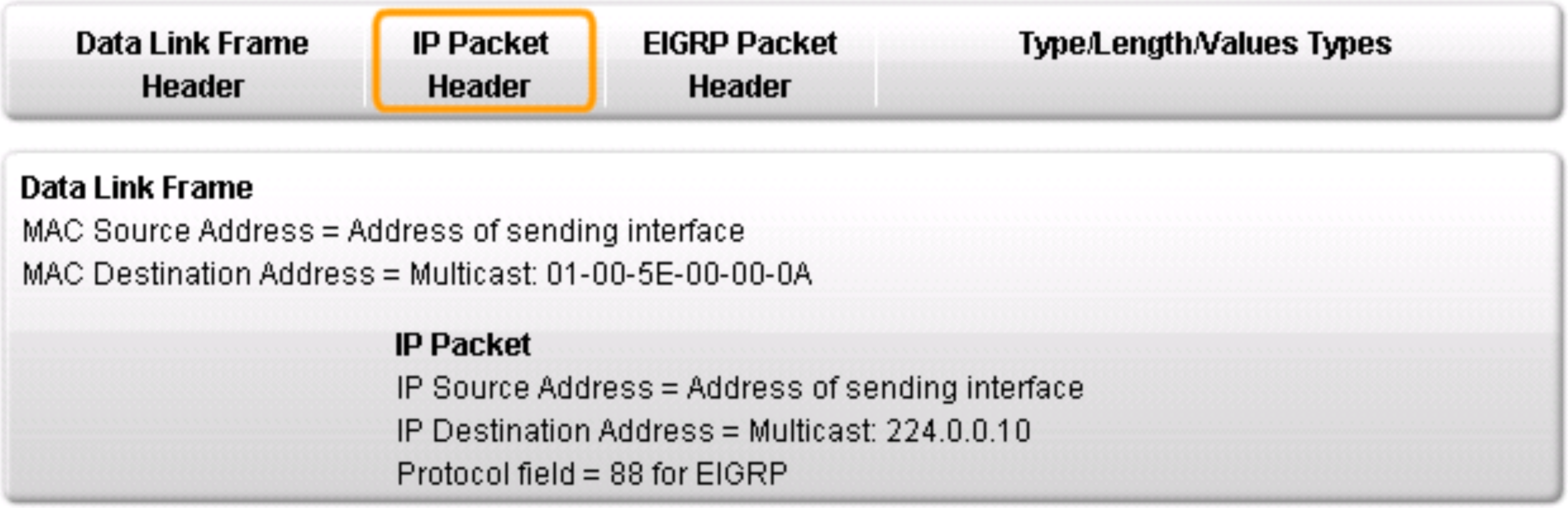
Data Link Frame

MAC Source Address = Address of sending interface

MAC Destination Address = Multicast: 01-00-5E-00-00-0A

- On an Ethernet segment, the EIGRP packet is encapsulated in a frame with a multicast address of: **01-00-5E-00-00-0A**.
- The IP packet header contains:
  - The multicast destination address **224.0.0.10**
  - The protocol field is set to **88** to indicate EIGRP.
- The data portion of the EIGRP message includes a packet header and Type/Length/Value or TLV.
  - The EIGRP packet header identifies the type of EIGRP message.
  - The TLV field contains EIGRP parameters, IP internal and external routes.

EIGRP IP Packet Header Format



Data Link Frame

MAC Source Address = Address of sending interface

MAC Destination Address = Multicast: 01-00-5E-00-00-0A

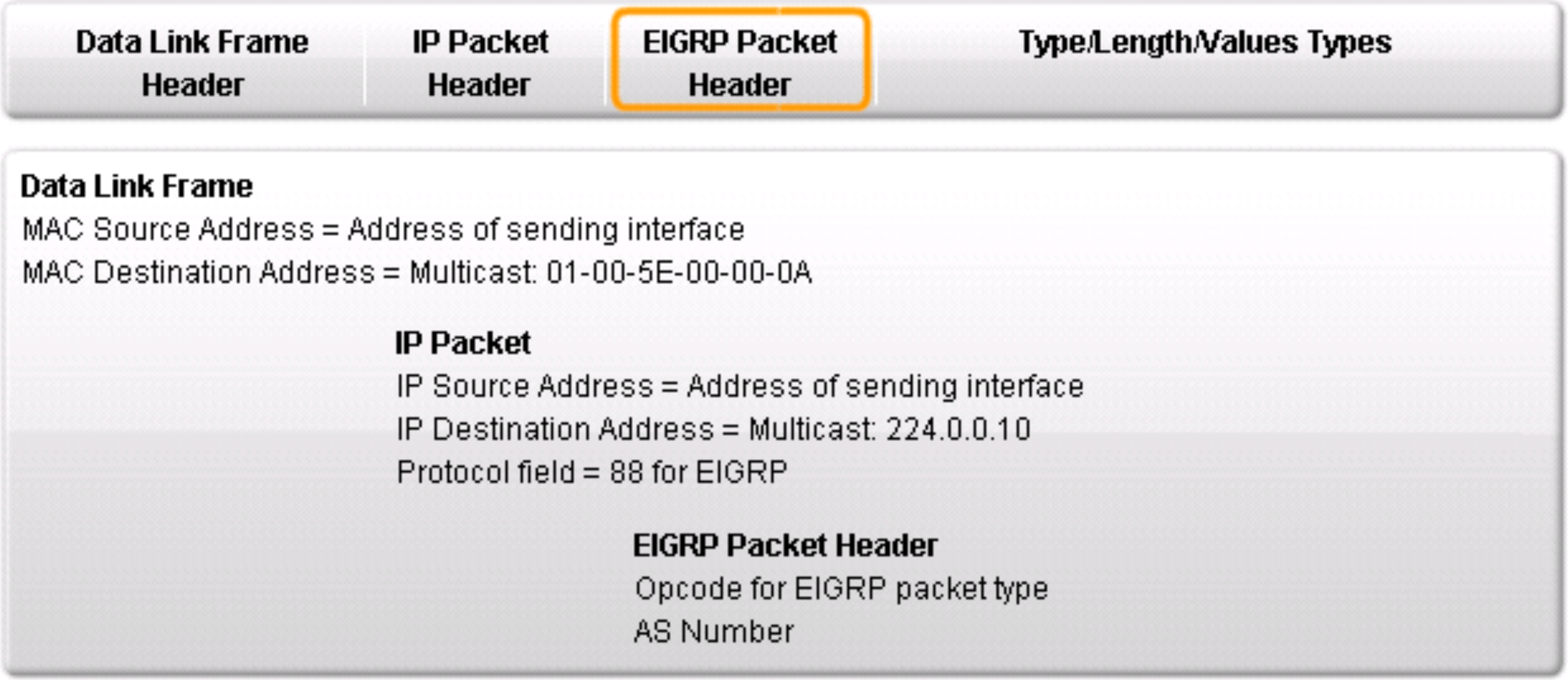
IP Packet

IP Source Address = Address of sending interface

IP Destination Address = Multicast: 224.0.0.10

Protocol field = 88 for EIGRP

EIGRP Packet Header Format



Data Link Frame

MAC Source Address = Address of sending interface

MAC Destination Address = Multicast: 01-00-5E-00-00-0A

IP Packet

IP Source Address = Address of sending interface

IP Destination Address = Multicast: 224.0.0.10

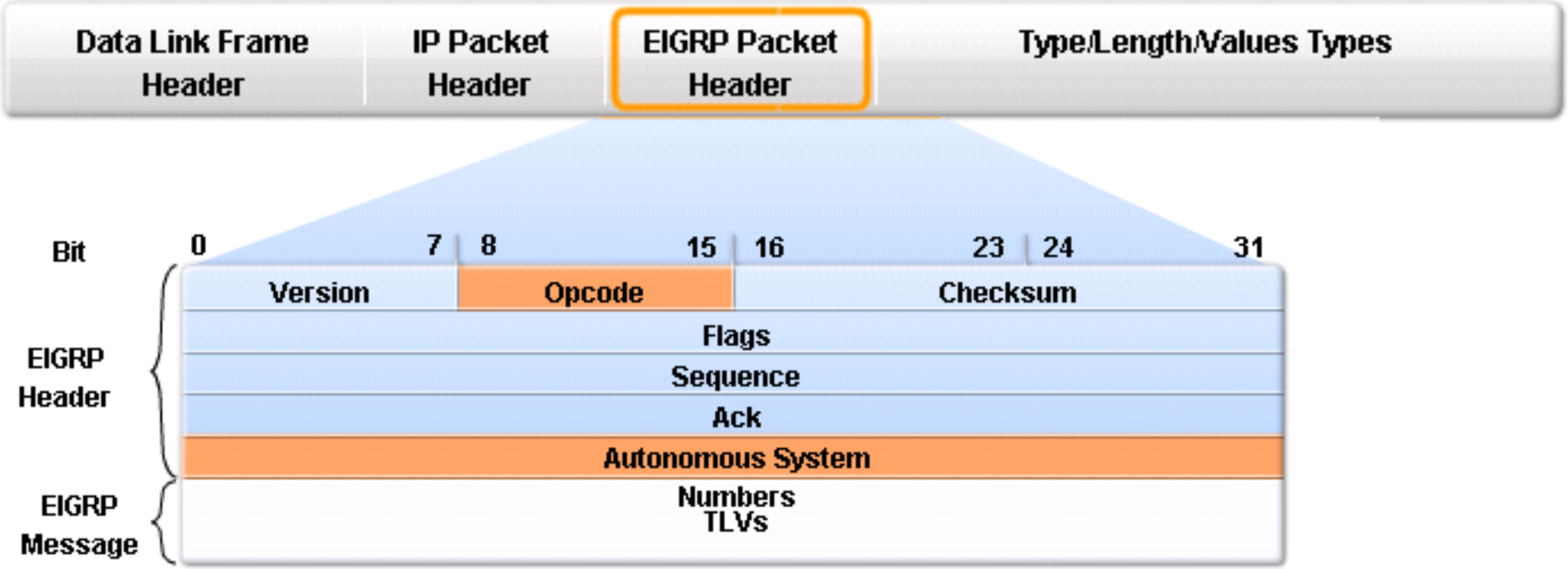
Protocol field = 88 for EIGRP

EIGRP Packet Header

Opcode for EIGRP packet type

AS Number

EIGRP Packet Header Format



Data Link Frame Header

IP Packet Header

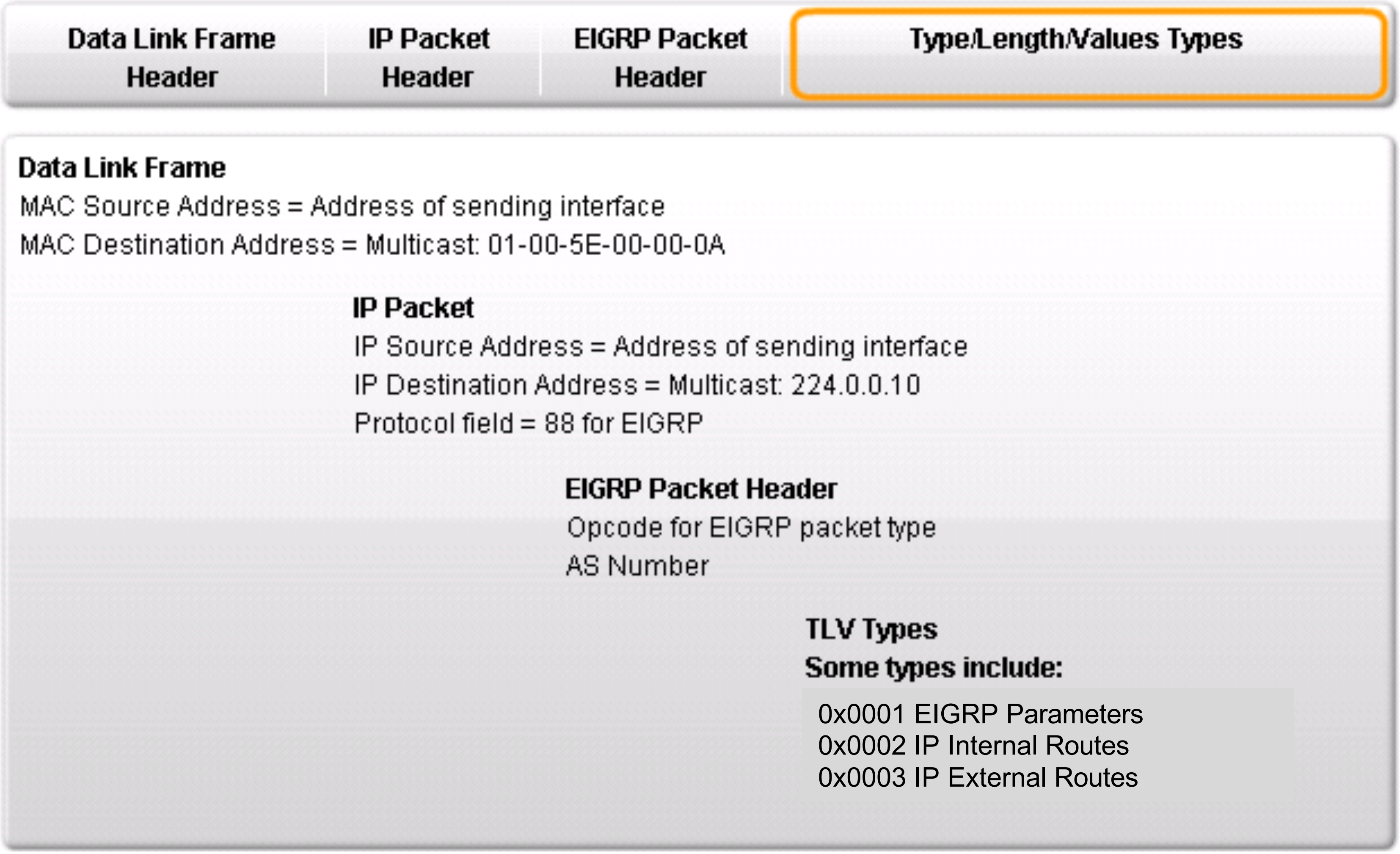
EIGRP Packet Header

Type/Length/Values Types

Bit	0 … 7	8 … 15	16 … 23	24 … 31
EIGRP Header	Version	Opcode	Checksum	
	Flags			
	Sequence			
	Ack			
	Autonomous System			
EIGRP Message	Numbers TLVs			

- **Opcode:** EIGRP Packet Type: Update (1),Query (3),Reply(4),Hello (5)
- **Autonomous System Number:** ID for this EIGRP routing process

EIGRP TLV Message Format



Data Link Frame Header	IP Packet Header	EIGRP Packet Header	Type/Length/Values Types

Data Link Frame

MAC Source Address = Address of sending interface

MAC Destination Address = Multicast: 01-00-5E-00-00-0A

IP Packet

IP Source Address = Address of sending interface

IP Destination Address = Multicast: 224.0.0.10

Protocol field = 88 for EIGRP

EIGRP Packet Header

Opcode for EIGRP packet type

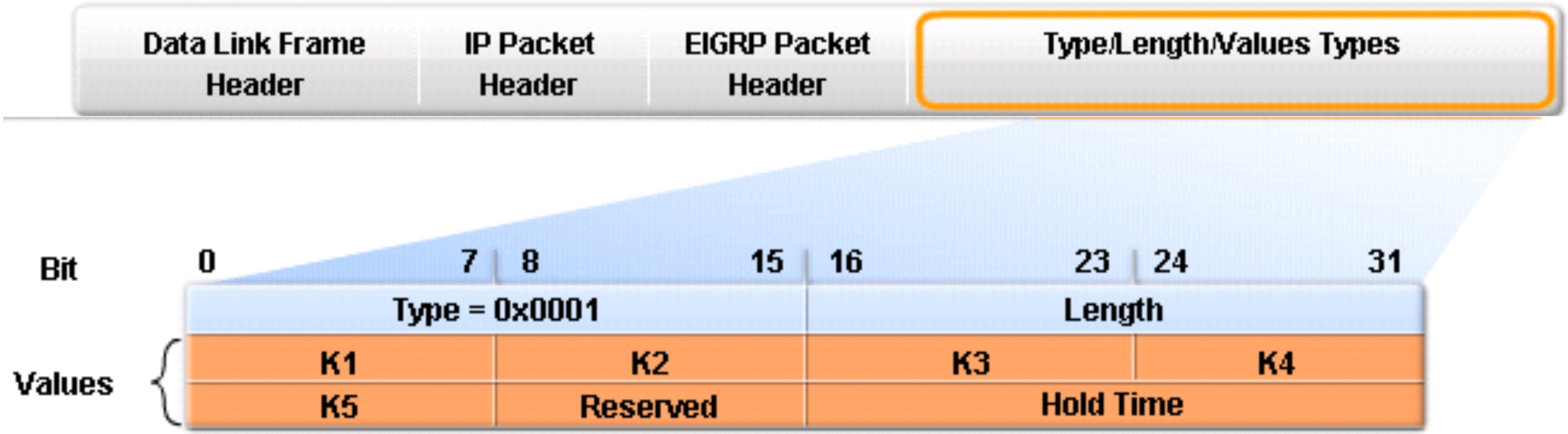
AS Number

TLV Types

Some types include:

0x0001 EIGRP Parameters
0x0002 IP Internal Routes
0x0003 IP External Routes

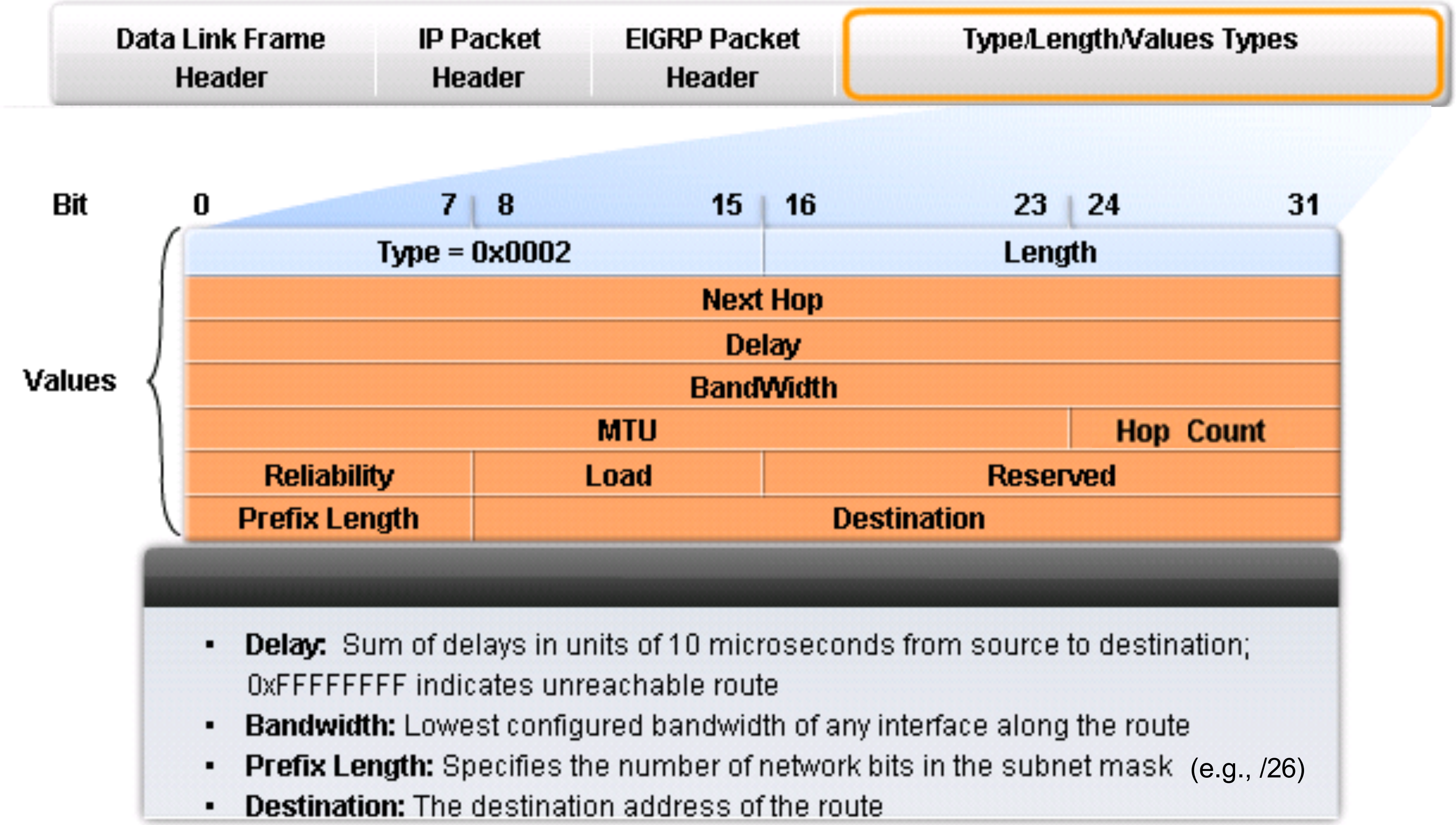
Type 0x0001 – EIGRP Parameters



Bit	0 7	8 15	16 23	24 31
	Type = 0x0001		Length	
Values	K1	K2	K3	K4
	K5	Reserved	Hold Time	

- **K1** and **K3:** Weights for bandwidth and delay; set to 1
- **Hold Time:** Maximum time router should wait for the next hello

Type 0x0002 – IP Internal Routes Format



Data Link Frame Header	IP Packet Header	EIGRP Packet Header	Type/Length/Values Types

Values

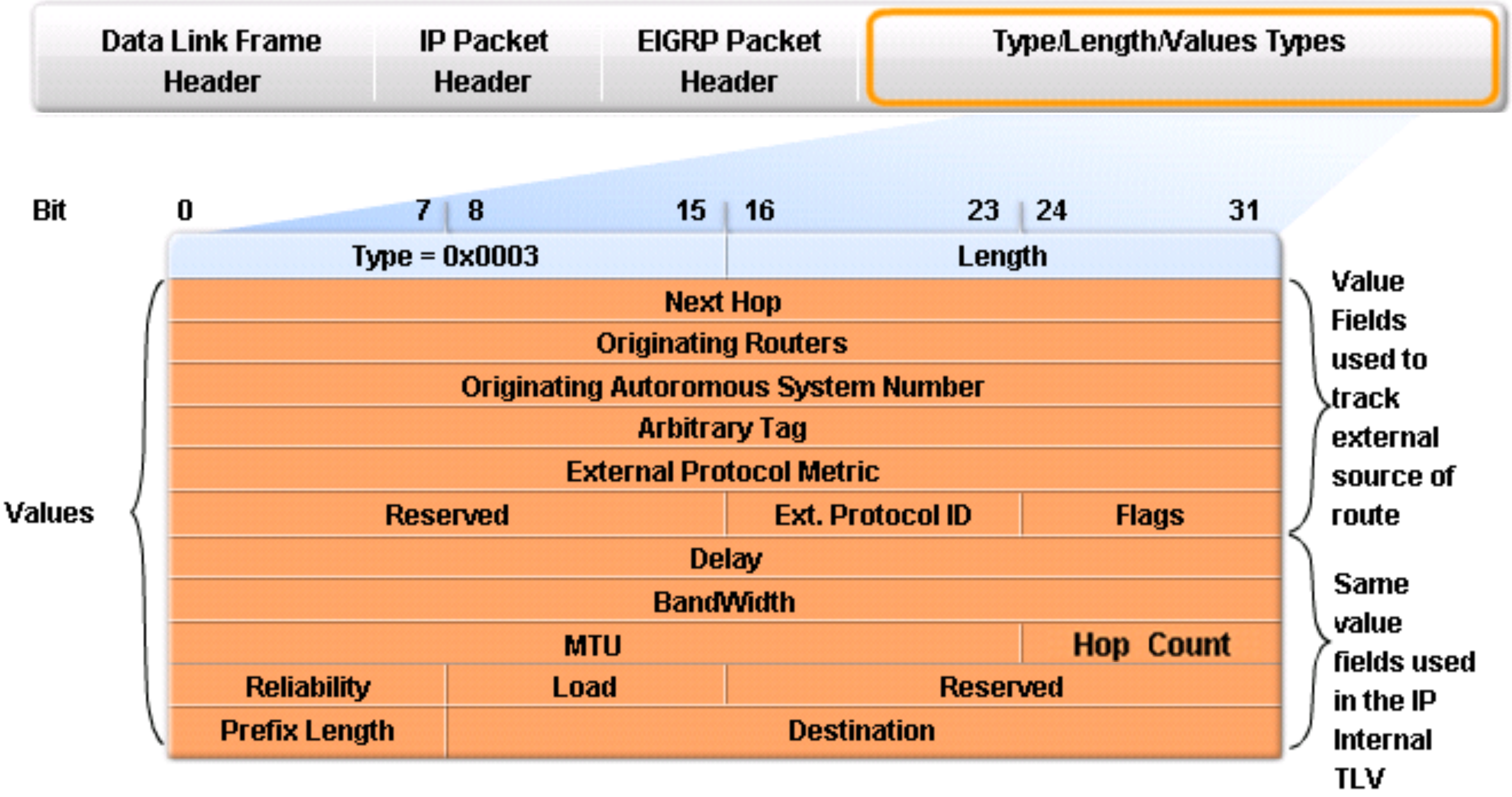
<table>
<tr><th>Bit</th><th>0 … 7</th><th>8 … 15</th><th>16 … 23</th><th>24 … 31</th></tr>
<tr><td></td><td colspan="2">Type = 0x0002</td><td colspan="2">Length</td></tr>
<tr><td></td><td colspan="4">Next Hop</td></tr>
<tr><td></td><td colspan="4">Delay</td></tr>
<tr><td></td><td colspan="4">BandWidth</td></tr>
<tr><td></td><td colspan="3">MTU</td><td>Hop Count</td></tr>
<tr><td></td><td>Reliability</td><td>Load</td><td colspan="2">Reserved</td></tr>
<tr><td></td><td>Prefix Length</td><td colspan="3">Destination</td></tr>
</table>

- **Delay:** Sum of delays in units of 10 microseconds from source to destination; 0xFFFFFFFF indicates unreachable route
- **Bandwidth:** Lowest configured bandwidth of any interface along the route
- **Prefix Length:** Specifies the number of network bits in the subnet mask (e.g., /26)
- **Destination:** The destination address of the route

Note:

- **MTU** (Maximum Transmission Unit is not a metric used by EIGRP.

Type 0x0003 –IP External Routes



Data Link Frame Header | IP Packet Header | EIGRP Packet Header | Type/Length/Values Types

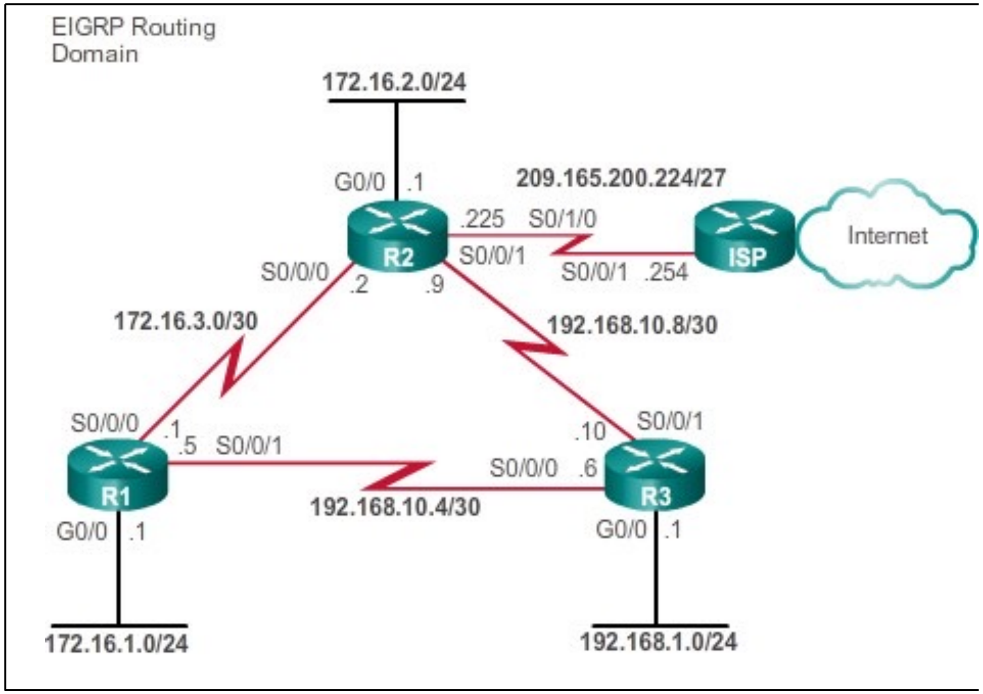
Bit	0 – 7	8 – 15	16 – 23	24 – 31
	Type = 0x0003		Length	
Values	Next Hop			
	Originating Routers			
	Originating Autoromous System Number			
	Arbitrary Tag			
	External Protocol Metric			
	Reserved		Ext. Protocol ID	Flags
	Delay			
	BandWidth			
	MTU			Hop Count
	Reliability	Load	Reserved	
	Prefix Length	Destination		

Value Fields used to track external source of route

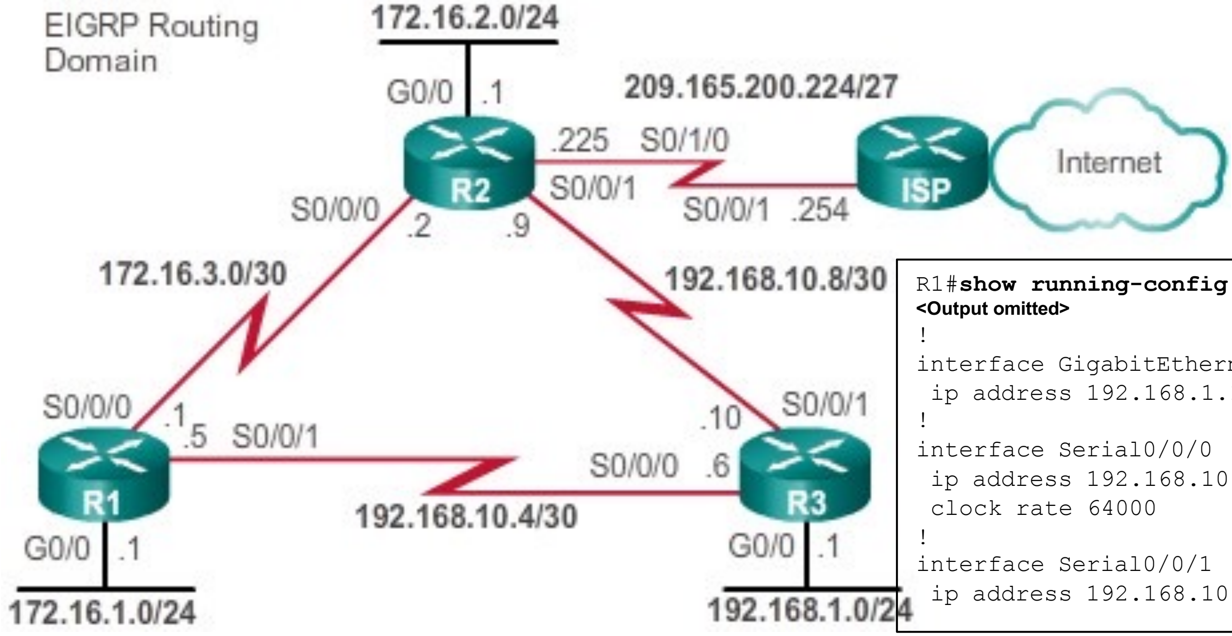
Same value fields used in the IP Internal TLV

IP external routes are routes which are imported into EIGRP through redistribution of a default route or other routing protocols.

Configuring EIGRP



EIGRP Network Topology

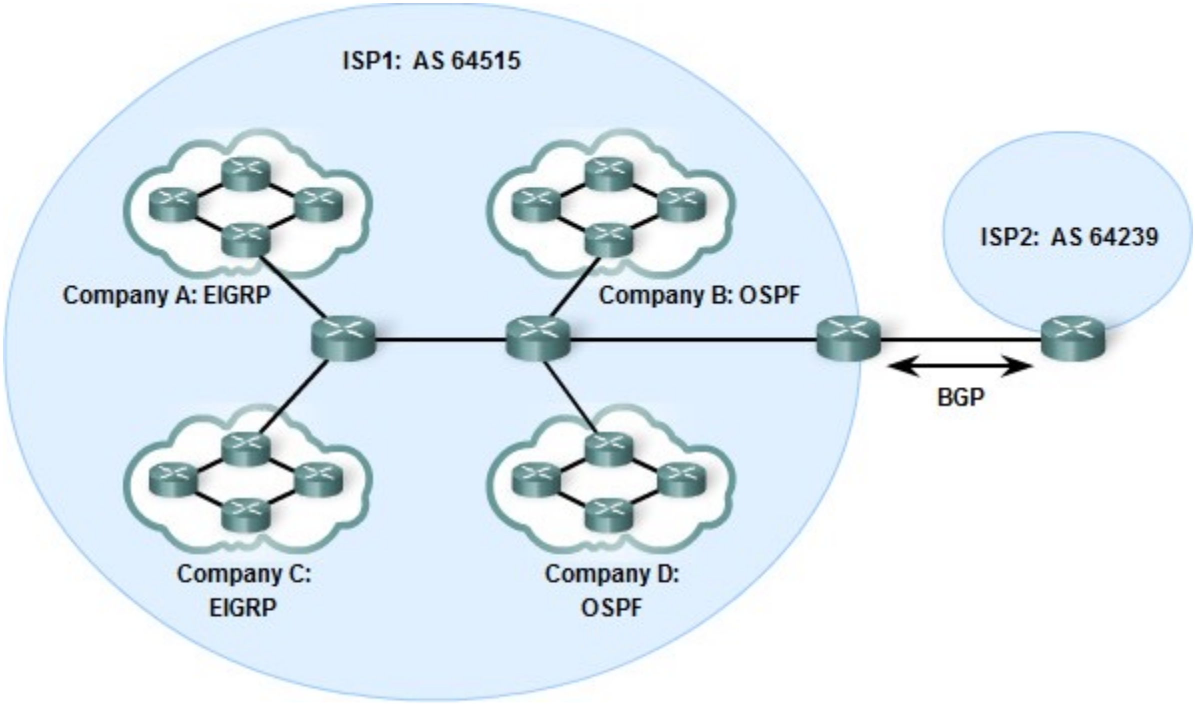


```
R1#show running-config
<Output omitted>
!
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.10.6 255.255.255.252
 clock rate 64000
!
interface Serial0/0/1
 ip address 192.168.10.10 255.255.255.252
```

```
R1#show running-config
<Output omitted>
!
interface GigabitEthernet0/0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.3.1 255.255.255.252
 clock rate 64000
!
interface Serial0/0/1
 ip address 192.168.10.5 255.255.255.252
```

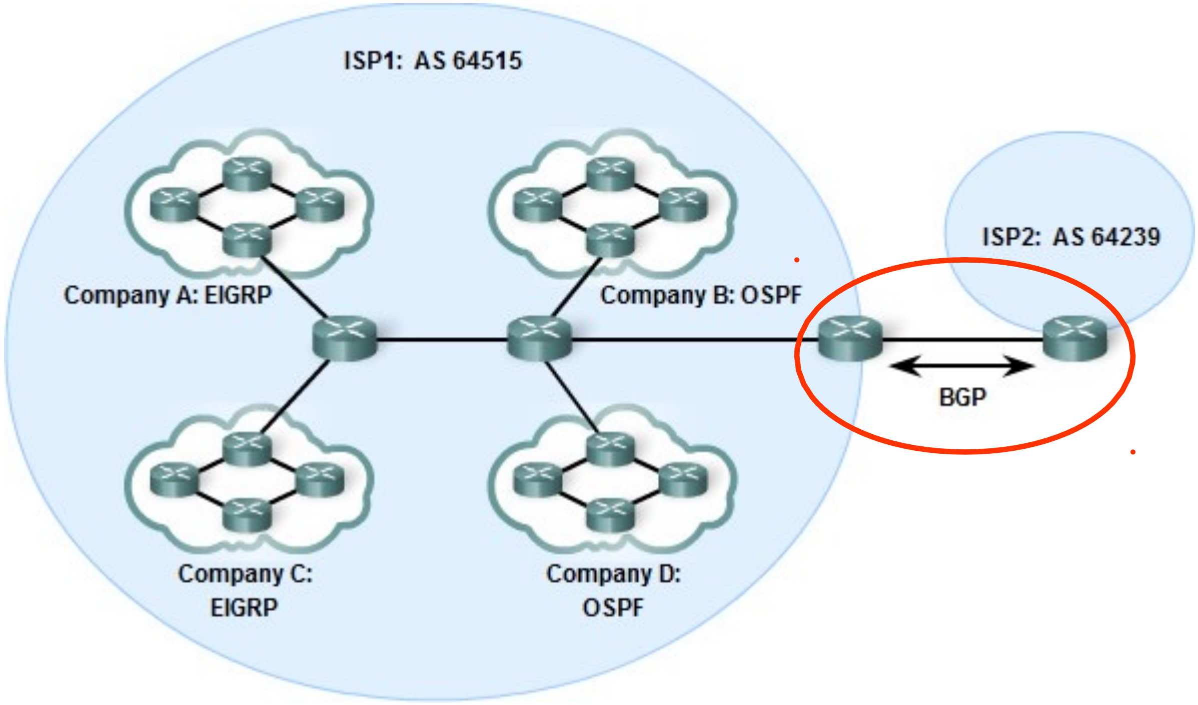
```
R2#show running-config
<Output omitted>
!
interface GigabitEthernet0/0
 ip address 172.16.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.3.2 255.255.255.252
!
interface Serial0/0/1
 ip address 192.168.10.9 255.255.255.252
 clock rate 64000
!
interface Serial0/1/0
 ip address 209.165.200.225 255.255.255.224
```

Autonomous Systems and Process IDs



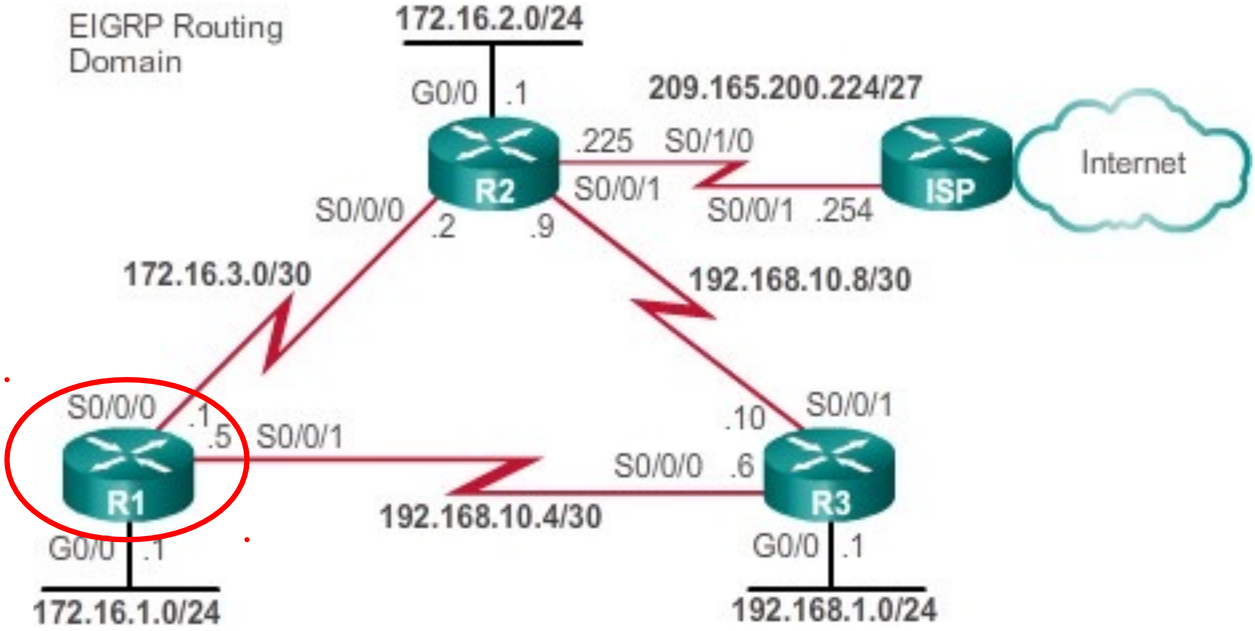
- An ***autonomous system (AS)*** is a collection of networks under the administrative control of a single entity that presents a common routing policy to the Internet.
  - Described in RFC 1930.
- AS numbers are assigned by IANA and its RIR.

Autonomous Systems and Process IDs



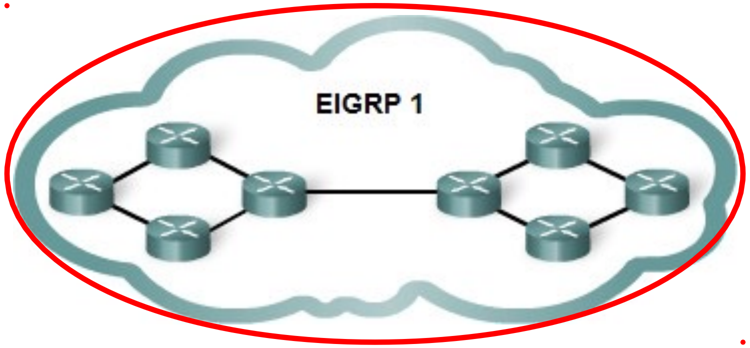
- Who needs an autonomous system number?
  - ISPs
  - Internet backbone providers
  - Large institutions connecting to other entities that also have an autonomous system number.
- What routing protocol is used between these providers?
  - **Exterior gateway routing protocol BGP**.
- ***Most companies and institutions with IP networks do not need an autonomous system number because they come under the control of a larger entity such as an ISP.***

router eigrp Command



```
R1# conf t
Enter configuration commands, one per line.   End with CNTL/Z.
R1(config)# router ?
  bgp          Border Gateway Protocol (BGP)
  eigrp        Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis         ISO IS-IS
  iso-igrp     IGRP for OSI networks
  mobile       Mobile routes
  odr          On Demand stub Routes
  ospf         Open Shortest Path First (OSPF)
  ospfv3       OSPFv3
  rip          Routing Information Protocol (RIP)

R1(config)#router eigrp 1
R1(config-router)#
```

Process ID



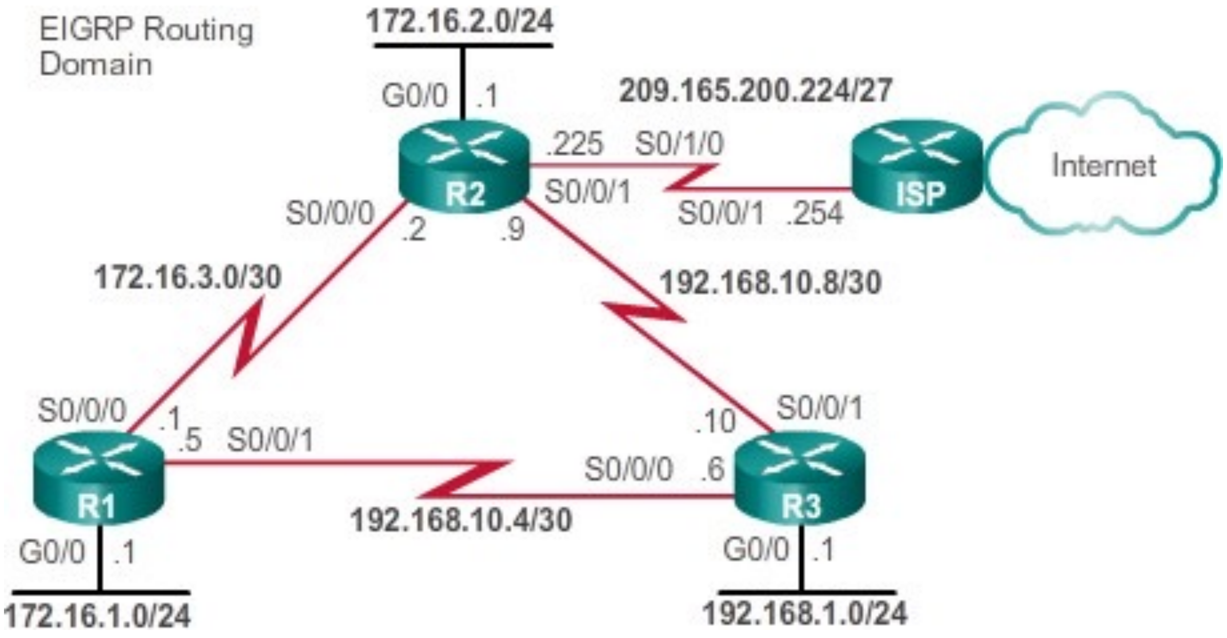
```
Router(config)# router eigrp autonomous-system
```

```
Router(config)# router eigrp 1
```

Must be same on all routers in EIGRP routing domain

- Both **EIGRP** and **OSPF** use a ***process ID*** to represent an instance of their respective routing protocol running on the router.
- **EIGRP** refers to "***autonomous-system***" number
  - Functions as a ***process ID***.
  - 1 and 65,535

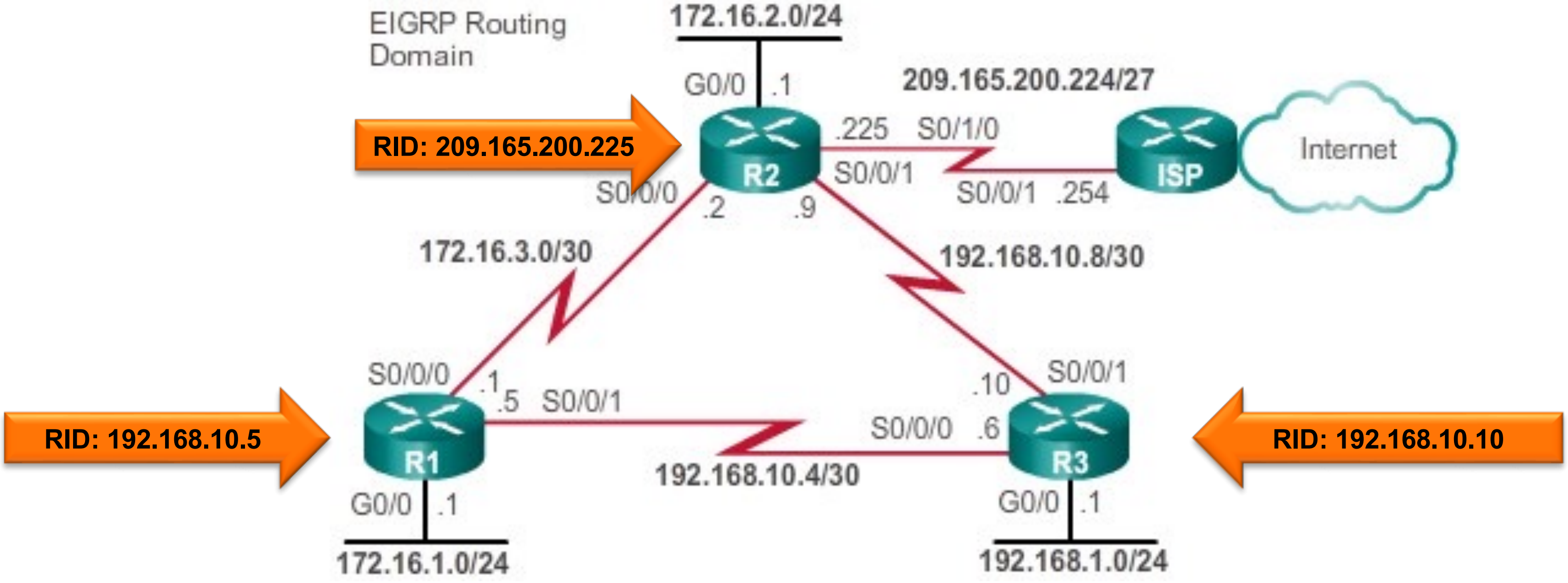
EIGRP Router ID



- The EIGRP router ID is used to uniquely identify each router in the EIGRP routing domain.

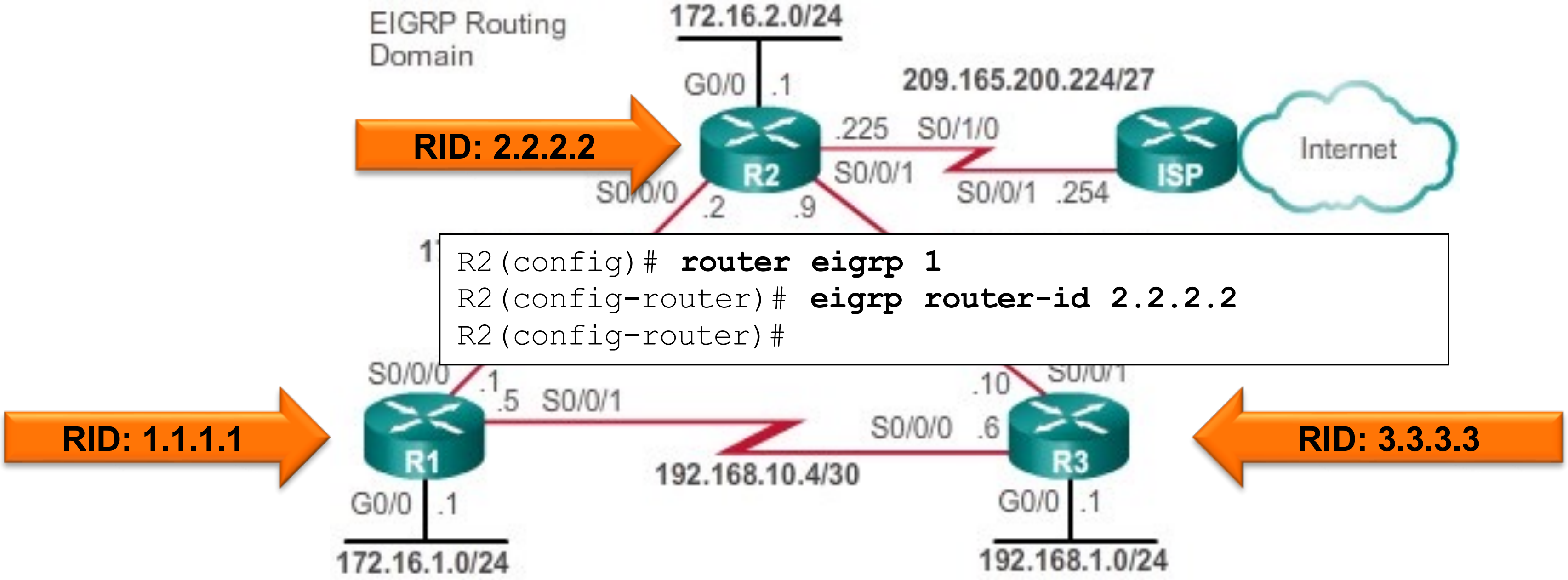
- Criteria for deriving the router ID:
  1. **Configured router ID**:
     - Configured with **`eigrp router-id`** *`router-id`* command
     - Note: Some versions of IOS will accept **`router-id`**.
  2. **Highest Loopback IPv4 address**:
  3. **Highest active interface IPv4 address**:

Current Router-IDs



- Given this topology, what would be the chosen router IDs of R1, R2, and R3?
- Configuring the router ID gives an administrator to selectively choose a specific IP address to represent the EIGRP router.

eigrp router-id Command



```
R2(config)# router eigrp 1
R2(config-router)# eigrp router-id 2.2.2.2
R2(config-router)#
```

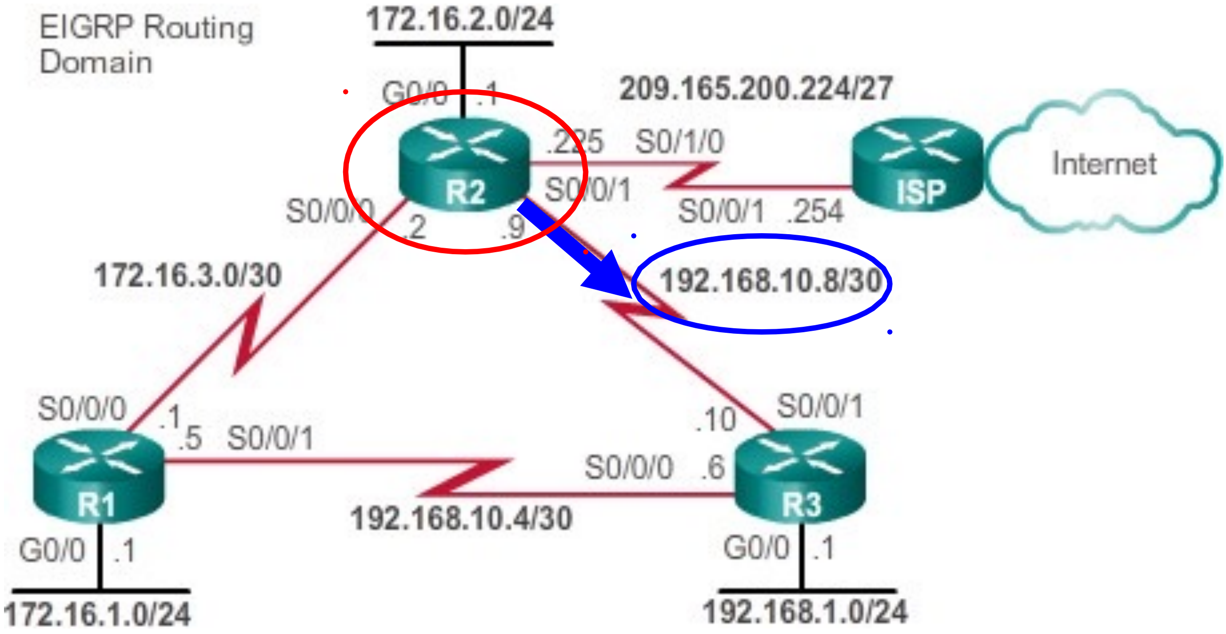
```
R1(config)# router eigrp 1
R1(config-router)# eigrp router-id 1.1.1.1
R1(config-router)#
```

```
R3(config)# router eigrp 1
R3(config-router)# eigrp router-id 3.3.3.3
R3(config-router)#
```

Verifying the Router-ID

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  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-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 1.1.1.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170

R1#
```

`network` Command

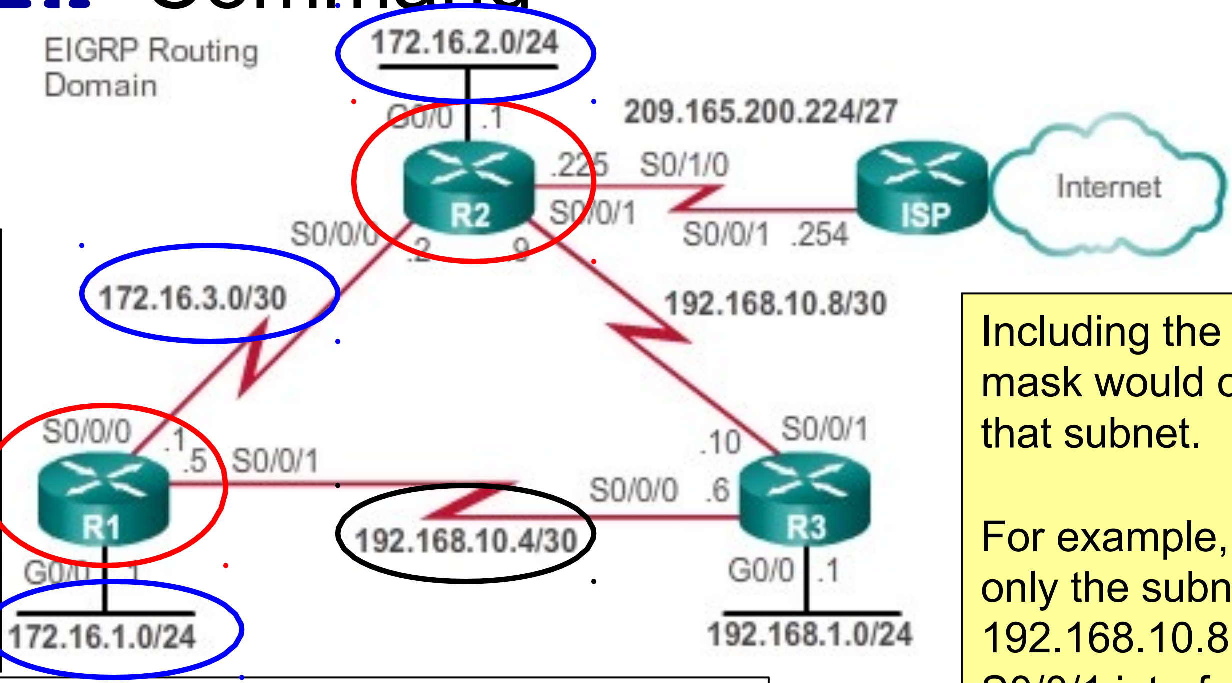


```
Router(config-router)# network network-address
```

```
R2(config-router)# network 192.168.10.0
```

- The **network** command in EIGRP has the same function as in other IGP routing protocols:
- What does it do?
  - Any interface on this router that matches the network address in the **network** command will be enabled to send and receive EIGRP updates.
  - This network (or subnet) will be included in EIGRP routing updates.

network Command



All interfaces belonging to the classful 172.16.0.0/16 address are enabled for EIGRP

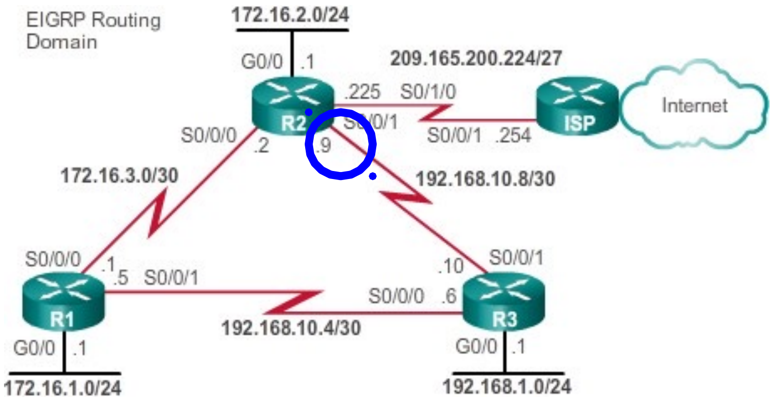
Including the wildcard mask would only advertise that subnet.

For example, to configure only the subnet 192.168.10.8 /30 on the S0/0/1 interface.

```
R1(config)# router eigrp 1
R1(config-router)# network 192.168.10.0
R1(config-router)# network 172.16.0.0
R1(config-router)#
```

```
R2(config)# router eigrp 1
R2(config-router)# network 172.16.0.0
R2(config-router)#
*Feb 28 17:51:42.543: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1:
Neighbor 172.16.3.1 (Serial0/0/0) is up: new adjacency
R2(config-router)#
```

The network Command with a Wildcard Mask

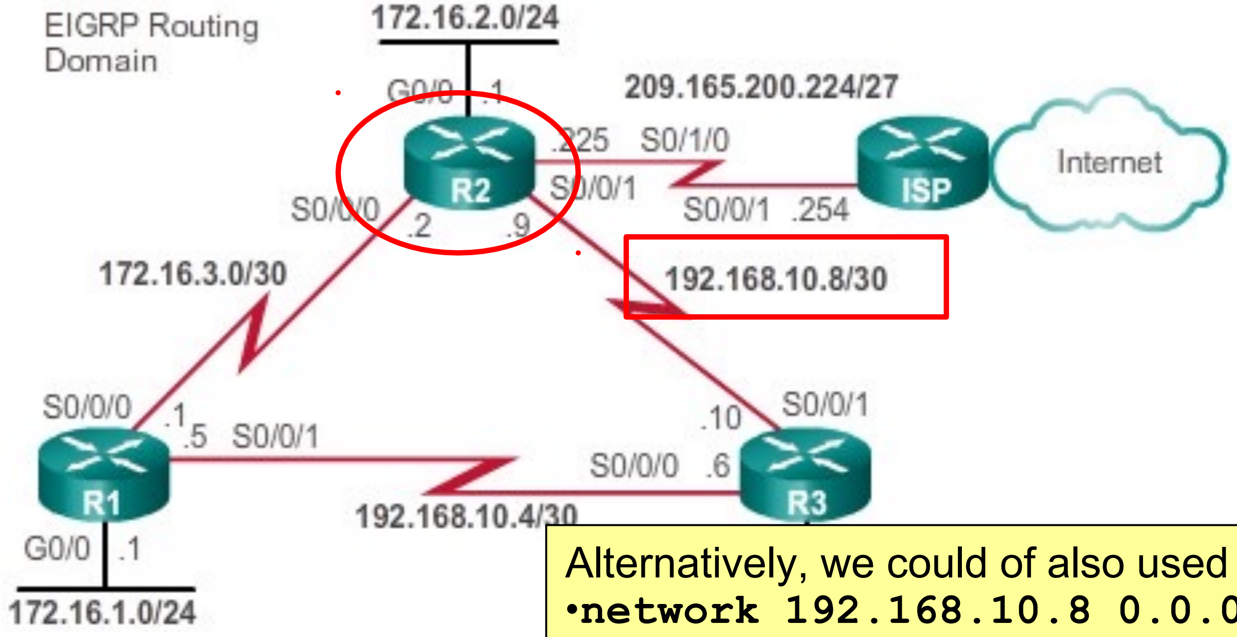


```
  255.255.255.255
- 255.255.255.252   Subtract the subnet mask
  ---------------
    0.  0.  0.  3   Wildcard mask
```

```
R2(config-router)# network 192.168.10.8 0.0.0.3
OR
R2(config-router)# network 192.168.10.8 255.255.255.252
```

- Think of a ***wildcard mask*** as the inverse of a subnet mask.
- To calculate the inverse of the subnet mask, subtract the subnet mask from 255.255.255.255.
- ***Some Cisco IOS software versions also let you just enter the subnet mask – it will correct it in the running-config.***

network Command

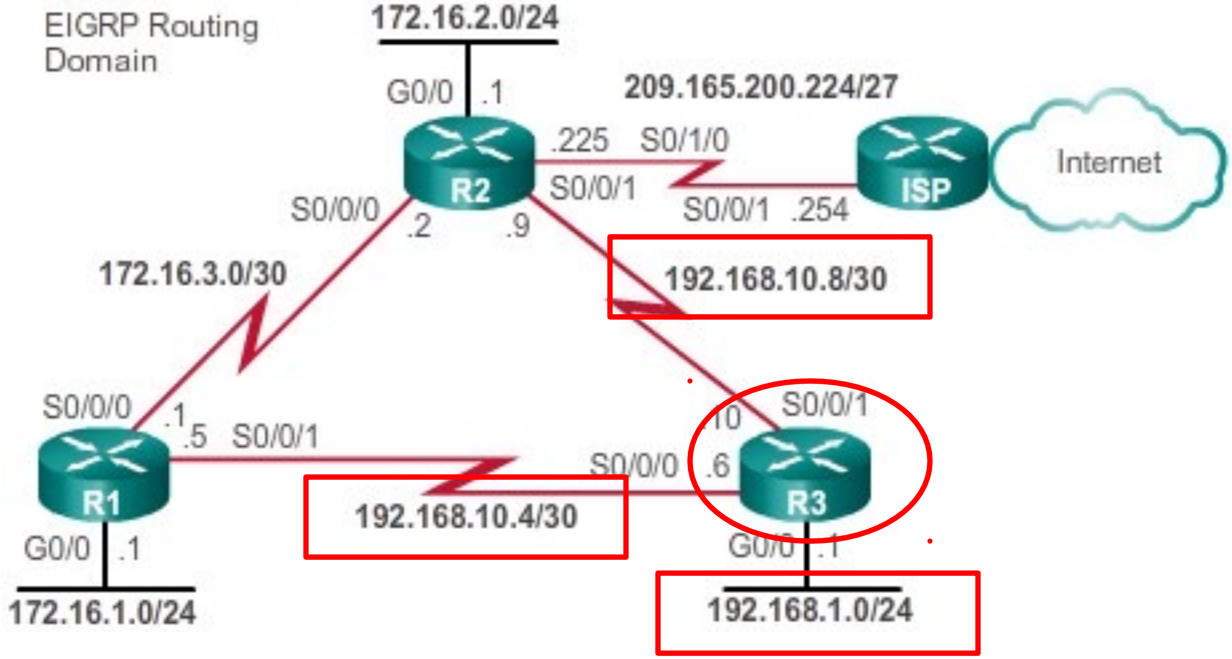


Alternatively, we could of also used either:
- **network 192.168.10.8 0.0.0.3**
- **network 192.168.10.9 0.0.0.0**

```
R2(config)# router eigrp 1
R2(config-router)# network 192.168.10.8 255.255.255.252
R2(config-router)# end
R2#
R2# show running-config | section eigrp 1
router eigrp 1
 network 172.16.0.0
 network 192.168.10.8 0.0.0.3
 eigrp router-id 2.2.2.2
R2#
```

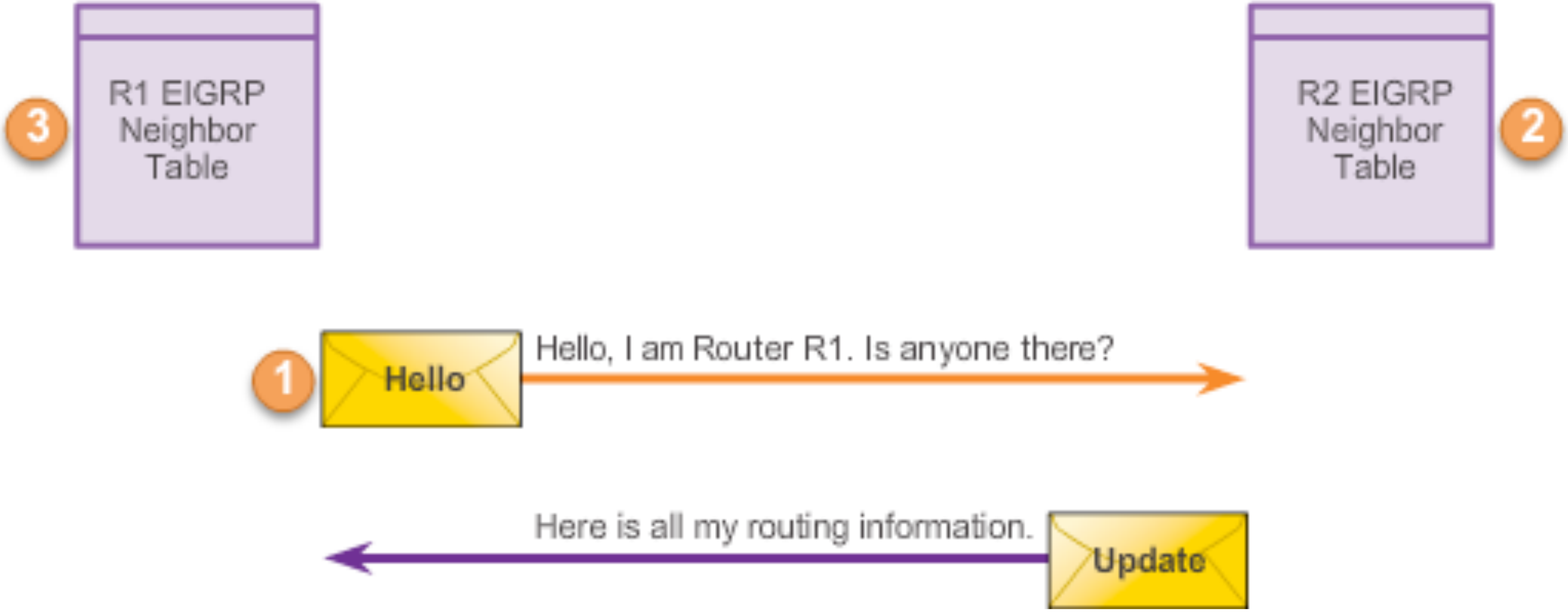
Notice how the EIGRP converts the entry into a wildcard mask.

network Command



```
R3(config)# router eigrp 1
R3(config-router)# network 192.168.1.0
R3(config-router)# network 192.168.10.4 0.0.0.3
*Feb 28 20:47:22.695: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor
192.168.10.5 (Serial0/0/0) is up: new adjacency
R3(config-router)#
R3(config-router)# network 192.168.10.8 0.0.0.3
*Feb 28 20:47:06.555: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor
192.168.10.9 (Serial0/0/1) is up: new adjacency
R3(config-router)#
```

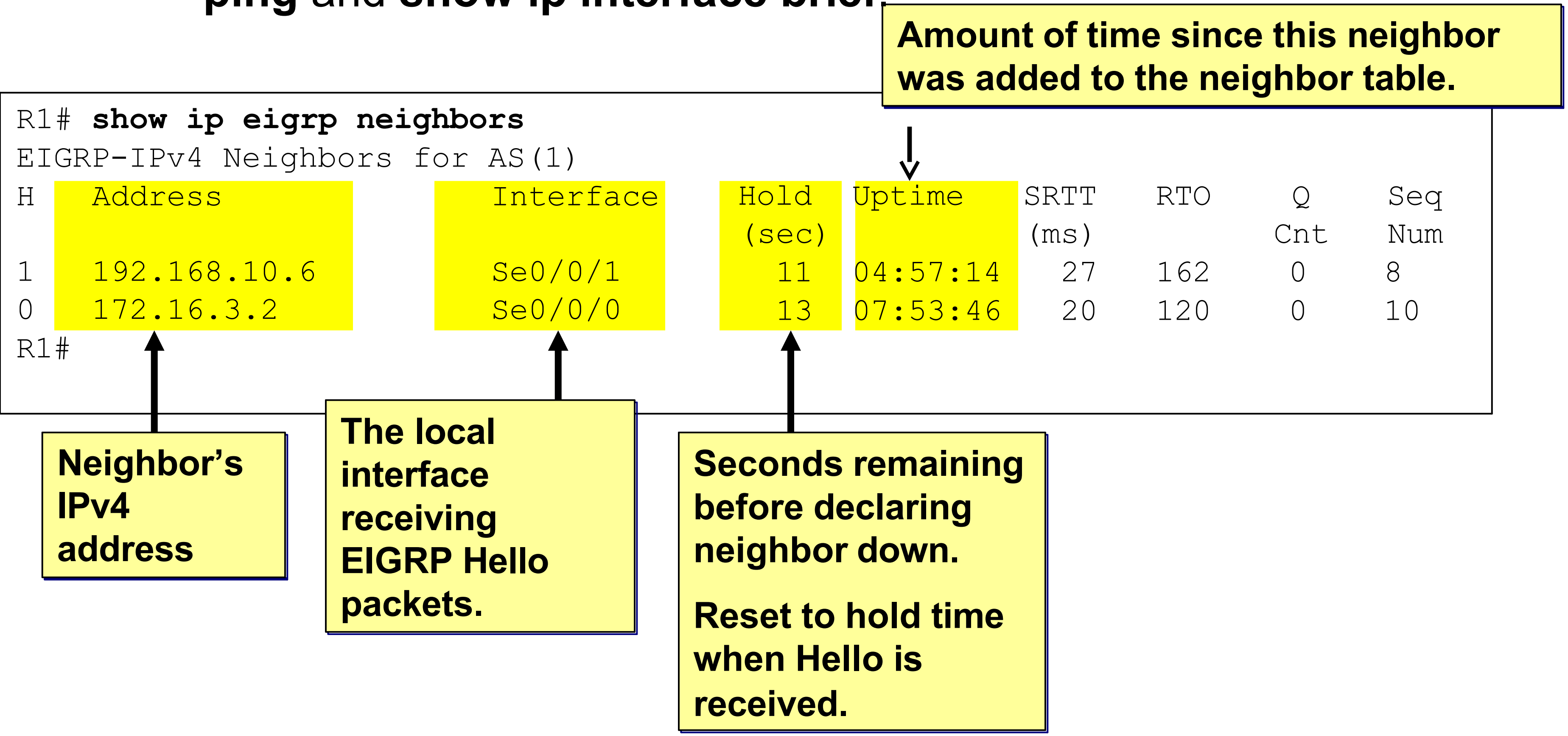
Discovering Neighbors


3
R1 EIGRP Neighbor Table
R2 EIGRP Neighbor Table
2
1
Hello
Hello, I am Router R1. Is anyone there?
Here is all my routing information.
Update

- Before any EIGRP packets can be exchanged between routers, EIGRP must first discover its neighbors.
- **EIGRP routers** discover neighbors and establish adjacencies with neighbor routers using the **hello** packet.
- Neighbors are added to the neighbor table.

Verifying Adjacencies

- Use the **`show ip eigrp neighbors`** command to view the neighbor table and verify that EIGRP has established an adjacency with its neighbors.
  - The output displays a list of each adjacent neighbor.
  - The command is very useful for troubleshooting EIGRP, followed by **ping** and **show ip interface brief**.



```
R1# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address              Interface     Hold  Uptime    SRTT   RTO    Q     Seq
                                       (sec)           (ms)          Cnt   Num
1   192.168.10.6         Se0/0/1        11   04:57:14   27    162    0     8
0   172.16.3.2           Se0/0/0        13   07:53:46   20    120    0     10
R1#
```

Amount of time since this neighbor was added to the neighbor table.

Neighbor's IPv4 address

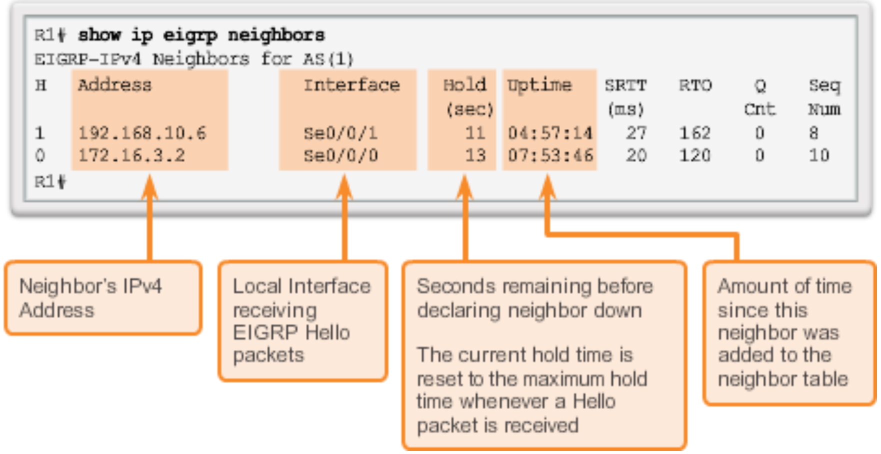
The local interface receiving EIGRP Hello packets.

Seconds remaining before declaring neighbor down.

Reset to hold time when Hello is received.

Verifying EIGRP

Verifying EIGRP



```
R1# show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address          Interface    Hold  Uptime    SRTT   RTO   Q    Seq
                                  (sec)           (ms)         Cnt  Num
1   192.168.10.6     Se0/0/1       11   04:57:14   27    162   0    8
0   172.16.3.2       Se0/0/0       13   07:53:46   20    120   0    10
R1#
```

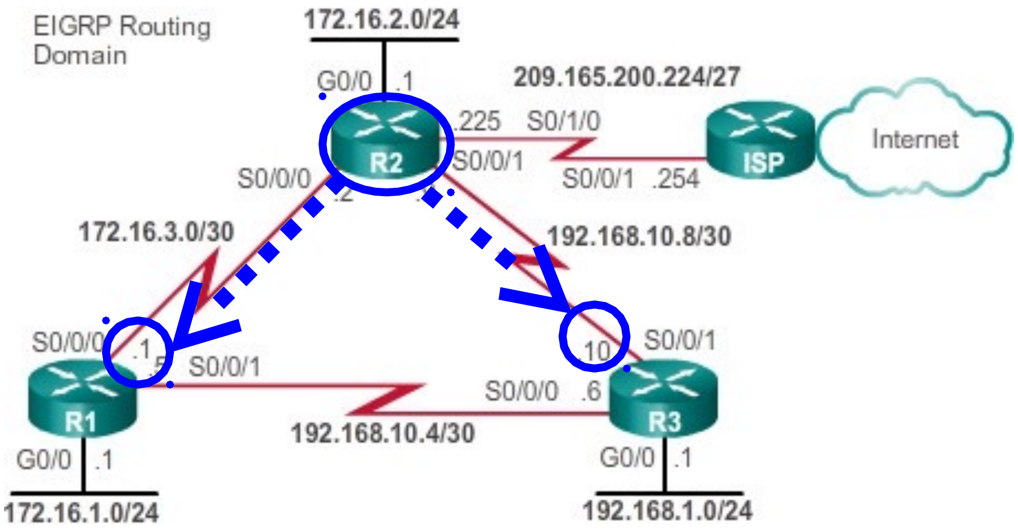
Neighbor's IPv4 Address

Local Interface receiving EIGRP Hello packets

Seconds remaining before declaring neighbor down

The current hold time is reset to the maximum hold time whenever a Hello packet is received

Amount of time since this neighbor was added to the neighbor table



- What if the **ping is successful** and EIGRP still does not see the router as a neighbor?
  - Are both routers configured with the same EIGRP process ID?
  - Is the directly connected network included in the EIGRP `network` statements?
  - Is the `passive-interface` command inappropriately configured, thus preventing EIGRP hello packets on the interface?

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  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-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 1.1.1.1
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.10.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.10.6          90      00:40:20
    172.16.3.2            90      00:40:20
  Distance: internal 90 external 170

R1#
```

1 **Routing protocol and Process ID (AS Number)**

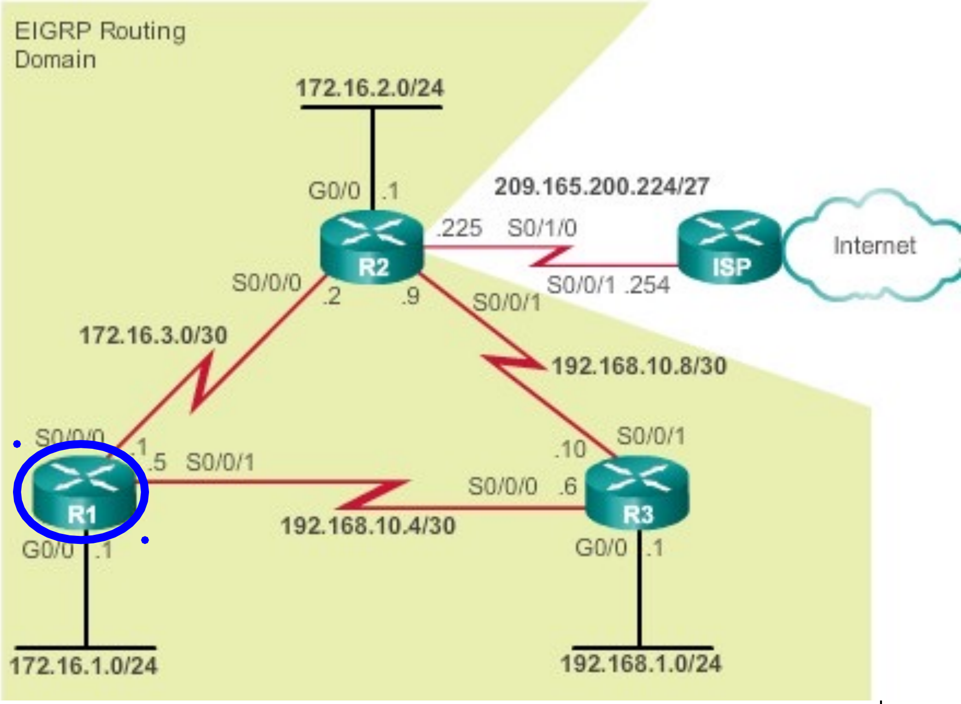
2 **K values used in composite metric**

3 **EIGRP Router ID**

4 **EIGRP Administrative Distances**

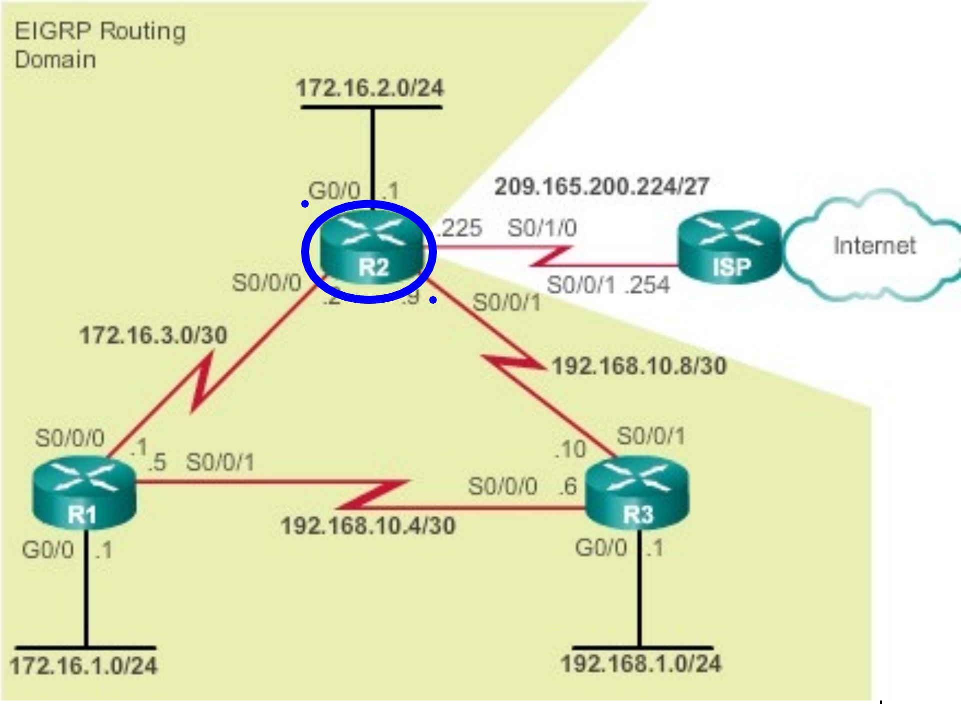
5 **Interfaces enabled for this EIGRP for IPv6.**

Verify the R1 Routing Table



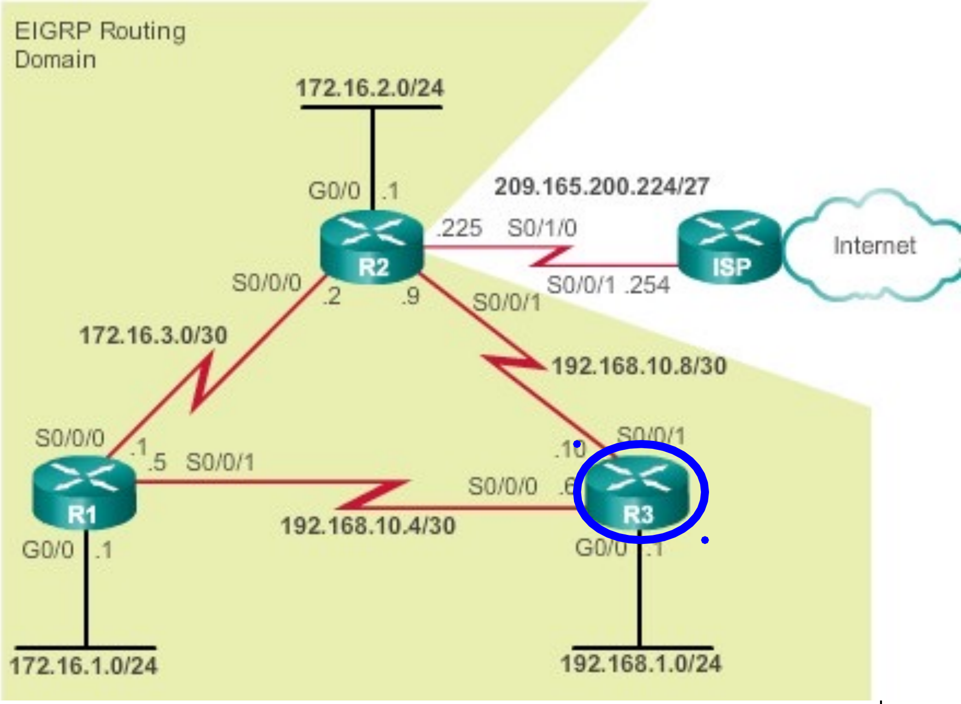
```
R1# show ip route | begin Gateway

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
C        172.16.1.0/24 is directly connected, GigabitEthernet0/0
L        172.16.1.1/32 is directly connected, GigabitEthernet0/0
D        172.16.2.0/24 [90/2170112] via 172.16.3.2, 00:14:35, Serial0/0/0
C        172.16.3.0/30 is directly connected, Serial0/0/0
L        172.16.3.1/32 is directly connected, Serial0/0/0
D     192.168.1.0/24 [90/2170112] via 192.168.10.6, 00:13:57, Serial0/0/1
      192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.10.4/30 is directly connected, Serial0/0/1
L        192.168.10.5/32 is directly connected, Serial0/0/1
D        192.168.10.8/30  [90/2681856] via 192.168.10.6, 00:50:42, Serial0/0/1
                          [90/2681856] via 172.16.3.2, 00:50:42, Serial0/0/0
R1#
```

Verify the R2 Routing Table



```
R2# show ip route | begin Gateway

Gateway of last resort is not set

 172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
D        172.16.1.0/24 [90/2170112] via 172.16.3.1, 00:11:05, Serial0/0/0
C        172.16.2.0/24 is directly connected, GigabitEthernet0/0
L        172.16.2.1/32 is directly connected, GigabitEthernet0/0
C        172.16.3.0/30 is directly connected, Serial0/0/0
L        172.16.3.2/32 is directly connected, Serial0/0/0
D     192.168.1.0/24 [90/2170112] via 192.168.10.10, 00:15:16, Serial0/0/1
      192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D        192.168.10.4/30 [90/2681856] via 192.168.10.10, 00:52:00, Serial0/0/1
                         [90/2681856] via 172.16.3.1, 00:52:00, Serial0/0/0
C        192.168.10.8/30 is directly connected, Serial0/0/1
L        192.168.10.9/32 is directly connected, Serial0/0/1
      209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.200.224/27 is directly connected, Loopback209
L        209.165.200.225/32 is directly connected, Loopback209
```

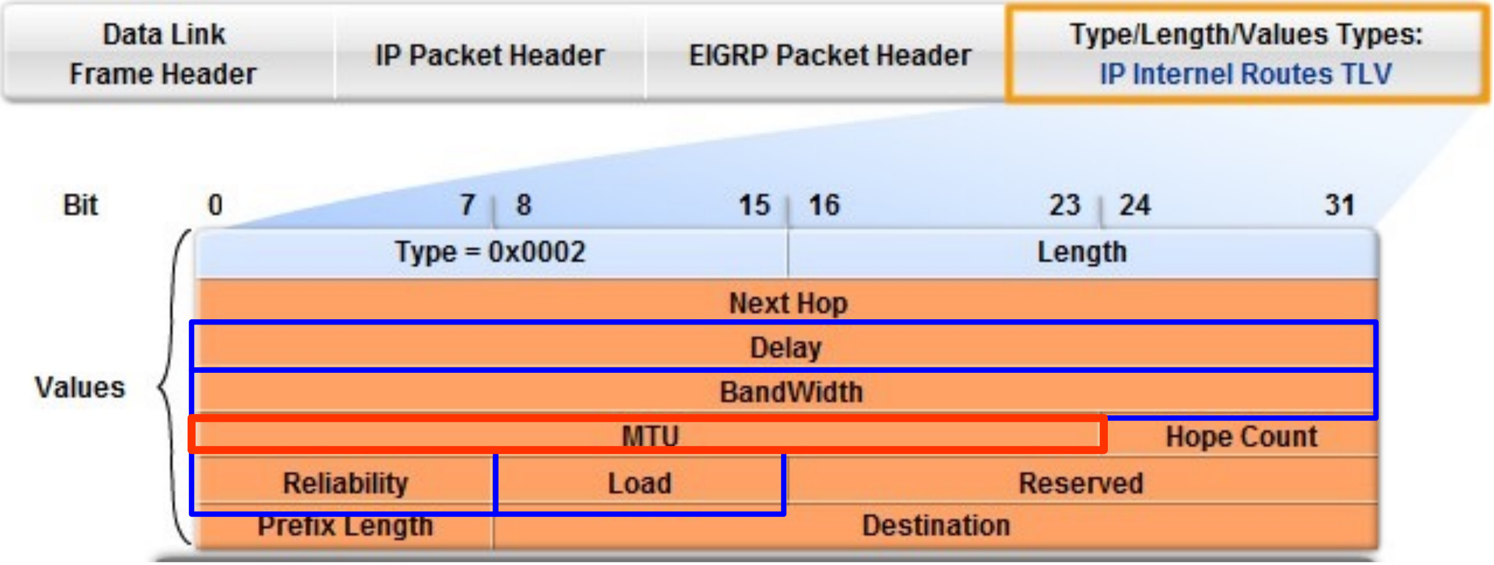
Verify the R3 Routing Table



```
R3# show ip route | begin Gateway

Gateway of last resort is not set

 172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D         172.16.1.0/24 [90/2170112] via 192.168.10.5, 00:12:00, Serial0/0/0
D         172.16.2.0/24 [90/2170112] via 192.168.10.9, 00:16:49, Serial0/0/1
D         172.16.3.0/30 [90/2681856] via 192.168.10.9, 00:52:55, Serial0/0/1
                        [90/2681856] via 192.168.10.5, 00:52:55, Serial0/0/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.10.0/24 is variably subnetted, 4 subnets, 2 masks
C        192.168.10.4/30 is directly connected, Serial0/0/0
L        192.168.10.6/32 is directly connected, Serial0/0/0
C        192.168.10.8/30 is directly connected, Serial0/0/1
L        192.168.10.10/32 is directly connected, Serial0/0/1
R3#
```

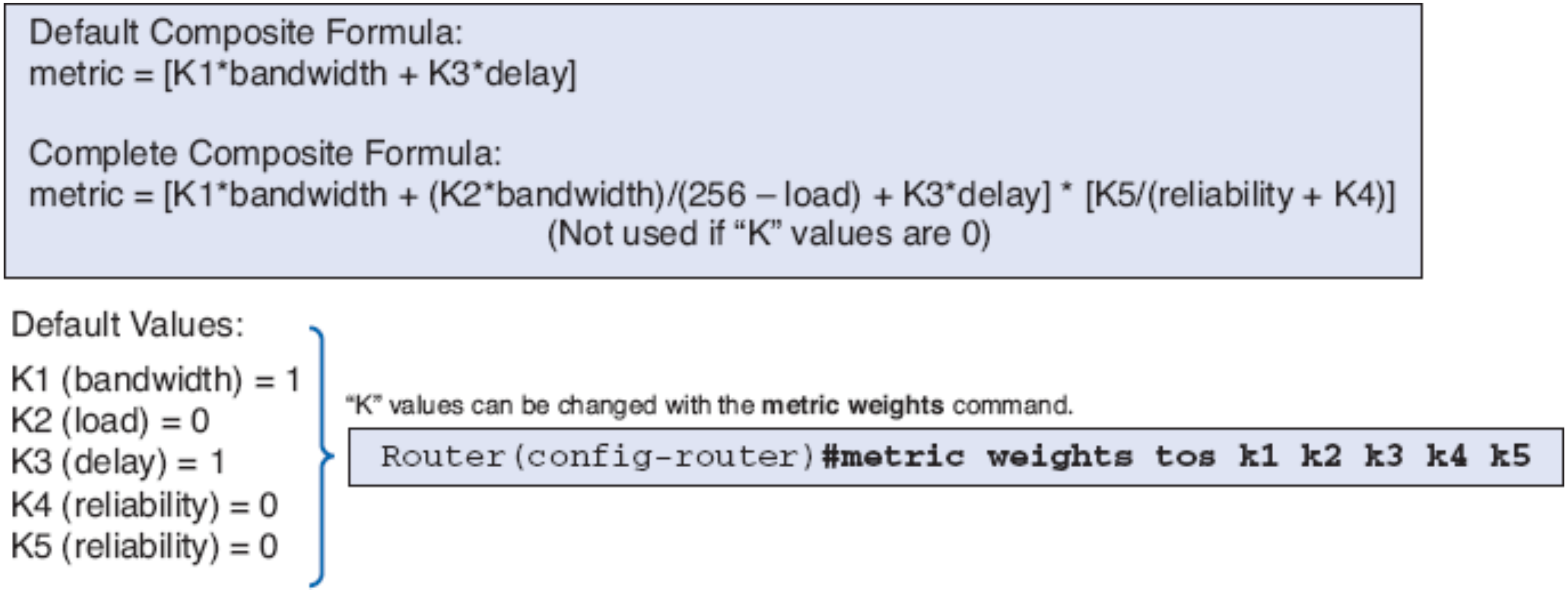
EIGRP Metrics

EIGRP Composite Metric and the K Values



- EIGRP uses the following values in its composite metric to calculate the preferred path to a network:
  - **Bandwidth:** The lowest bandwidth between source and destination.
  - **Delay:** The cumulative interface delay along the path
  - **Reliability:** Worst reliability between source and destination, based on keepalives.
  - **Load**: Worst load on a link between source and destination, based on the packet rate and the configured bandwidth of the interface.
- **Note**: Although **MTU** is included in the routing table updates, it is not a routing metric used by EIGRP or IGRP.

The Composite Metric



Default Composite Formula:
metric = [K1*bandwidth + K3*delay]

Complete Composite Formula:
metric = [K1*bandwidth + (K2*bandwidth)/(256 – load) + K3*delay] * [K5/(reliability + K4)]
(Not used if "K" values are 0)

Default Values:

K1 (bandwidth) = 1
K2 (load) = 0
K3 (delay) = 1
K4 (reliability) = 0
K5 (reliability) = 0

"K" values can be changed with the **metric weights** command.

```
Router(config-router)#metric weights tos k1 k2 k3 k4 k5
```

- By default:
  - K1 and K3 are set to 1.
  - K2, K4, and K5 are set to 0.
- The result in the default case is that only the bandwidth and delay values are used to compute the default composite metric.

Note: The formula above needs to be clarified. When K5 = 0, the reliability term is dropped entirely (to prevent multiplication by zero).

Verifying the K Values

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  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-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  <output omitted>
```

- The K values on R1 are set to the default.
- Changing these values to other than the default is not recommended unless the network administrator has a very good reason to do so

Examining the Metric Values

```
R1# show interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Description: Link to R2
  Internet address is 172.16.3.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  <output omitted>
```

- **`show interface`** command, lets you can examine the actual values used for bandwidth, delay, reliability, and load in the computation of the routing metric.
- Default values:
  - bandwidth
  - delay

Bandwidth

```
R1# show interface serial 0/0/0
  <output omitted>
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  <output omitted>
```

- The ***bandwidth*** metric (1544 Kbps) is a static value used by some routing protocols such as EIGRP and OSPF to calculate their routing metric.
  - Kilobits per second (Kbps).
  - Most **serial interfaces** use the default bandwidth value of 1544 Kbps or 1,544,000 bps (1.544 Mbps).

- The **value of the bandwidth** might or might not reflect the actual physical bandwidth of the interface.
  - **Modifying the bandwidth** value does not change the actual bandwidth of the link.
  - Should reflect actual bandwidth of the link. (coming).

Delay

```
R1# show interface serial 0/0/0
  <output omitted>
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  <output omitted>
```

- ***Delay*** is a measure of the time it takes for a packet to traverse a route.
  - Based on the type of link, or interface
  - Expressed in microseconds (millionths of a second).
  - The router does not actually track how long packets are taking to reach the destination.
  - Like the bandwidth value, delay is a default value that can be changed by the network administrator.

Delay

```
R1# show interface serial 0/0/0
  <output omitted>
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  <output omitted>
```

- 100 microseconds for Fast Ethernet interfaces.
- Default value is 20,000 microseconds for serial interfaces

Media	Delay In usec
Gigabit Ethernet	10
Fast Ethernet	100
FDDI	100
16M Token Ring	630
Ethernet	1,000
T1 (Serial Default)	20,000
DS0 (64 Kbps)	20,000
1024 Kbps	20,000
56 Kbps	20,000

Reliability – Optional Metric

```
R1# show interface serial 0/0/0
  <output omitted>
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  <output omitted>
```

- ***Reliability*** is a measure of the probability that the link will fail or how often the link has experienced errors.
  - A value between 0 and 255,
    - 1 = a minimally reliable link
    - 255 = 100 percent reliable.
  - By **default,** EIGRP does not use reliability in its metric calculation.

Load – Optional Metric

```
R1# show interface serial 0/0/0
  <output omitted>
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  <output omitted>
```

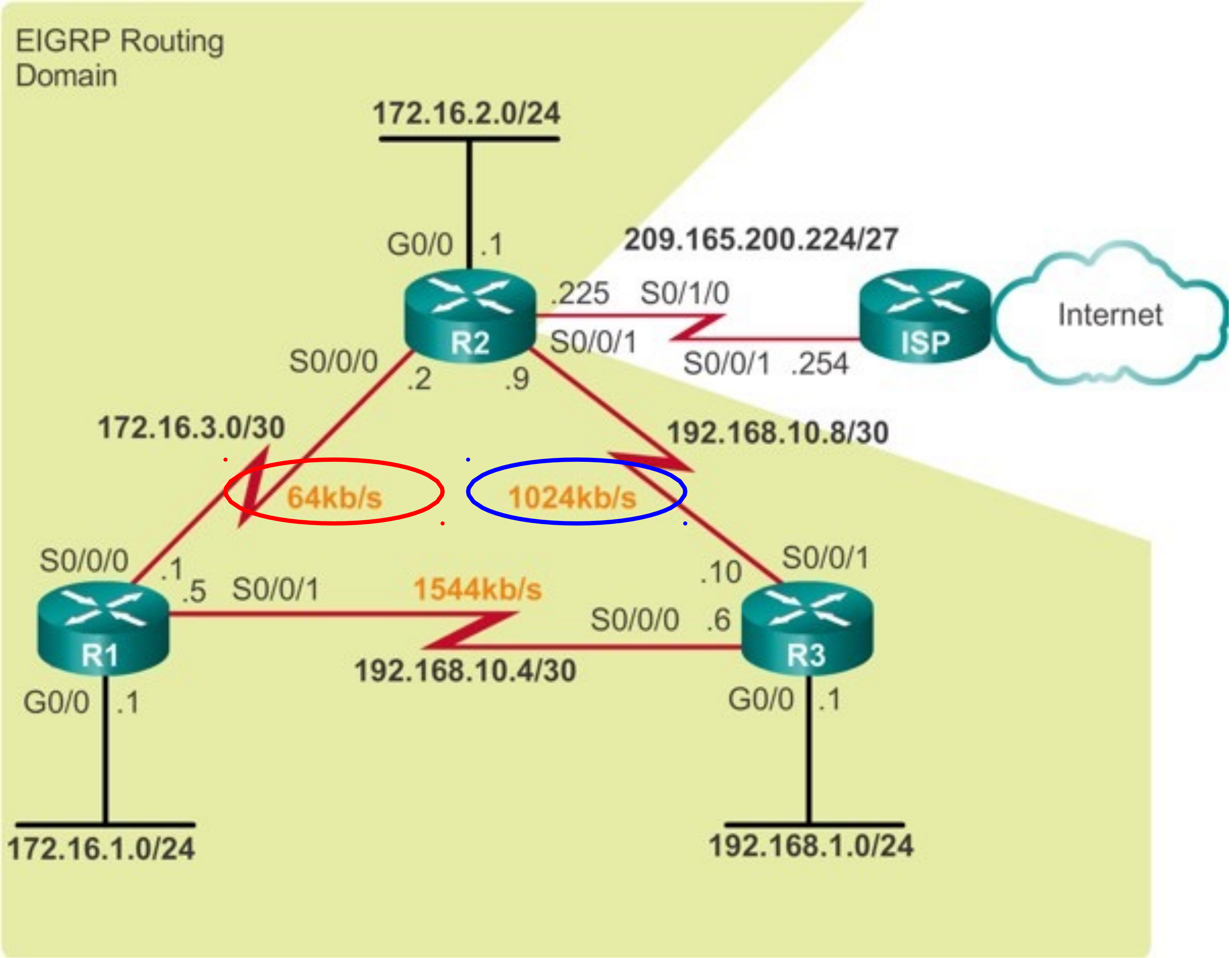
- ***Load*** reflects the amount of traffic using the link.
  - A value between 0 and 255.
  - A lower load value is more desirable because it indicates less load on the link.
    - 1/255 would be a minimally loaded link.
    - 40/255 is a link at 16 percent capacity
    - 255/255 is a link that is 100 percent saturated
- By **default,** EIGRP does not use load in its metric calculation.

Using the bandwidth Command

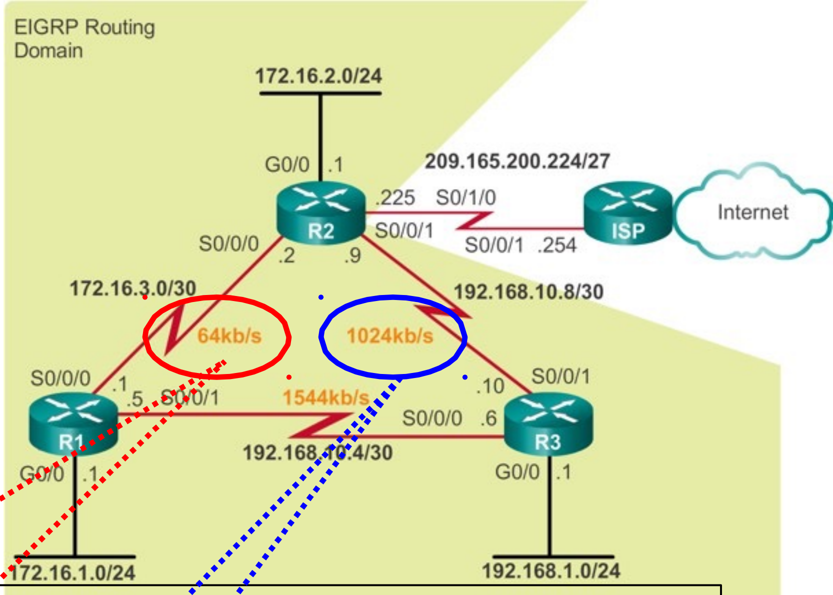
```
Router(config-if)# bandwidth kilobits
```

- Most serial links, the bandwidth metric defaults to 1544 Kbps.
- Correct value for bandwidth is very important to the accuracy of routing information
  - **`bandwidth`** command **-** modifies the bandwidth metric.
  - **`no bandwidth`** - restores the default value.

Adjusting the EIGRP Bandwidth Example



Using the `bandwidth` Command



Configure the bandwidth commands for R1, R2, and R3.

```
R1(config)# inter s0/0/0
R1(config-if)# bandwidth 64
```

```
R2(config)# inter s0/0/0
R2(config-if)# bandwidth 64

R2(config)# inter s0/0/1
R2(config-if)# bandwidth 1024
```

```
R3(config)# inter s0/0/1
R3(config-if)# bandwidth 1024
```

- Modify the bandwidth on the appropriate serial interfaces.
- Be sure to modify both ends of the link.

Verify the EIGRP Bandwidth

- It is important that the bandwidth parameters are the same on both sides of the link to ensure proper routing in both directions.

```
R1# show interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Internet address is 172.16.3.1/30
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255

<Output omitted>
R1#

R2# show interface s0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Internet address is 172.16.3.2/30
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255

<Output omitted>
R2#
```

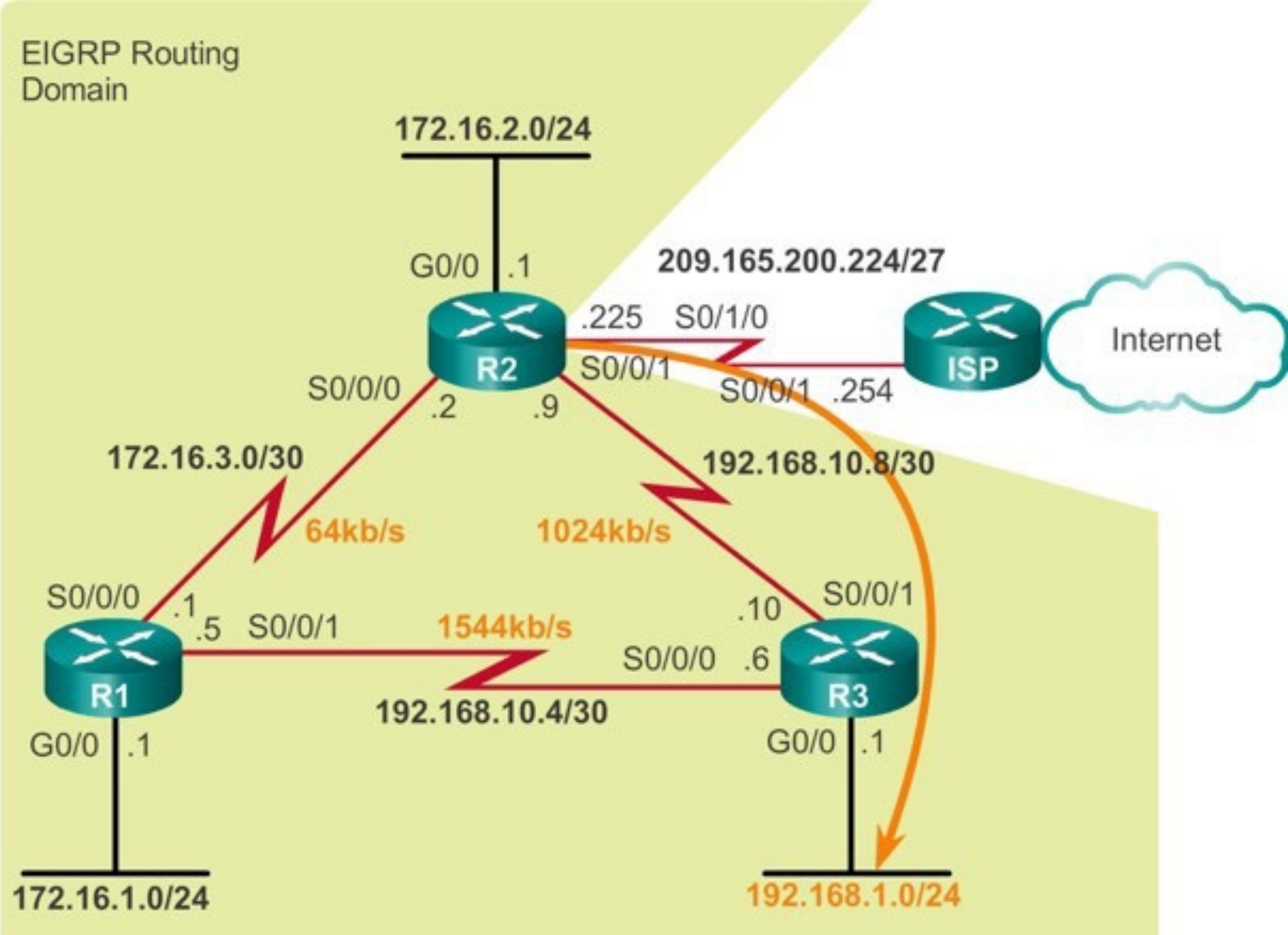
EIGRP Composite Metric Formula

[1 x Bandwidth + (0 x Bandwidth) / (256 - load) + 1 x Delay]) x 256

- The multiplication by 256 is a legacy feature to maintain backward compatibility with IGRP.

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  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-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 1.1.1.1

<Output omitted>

R1#
```

Example Metric Calculation



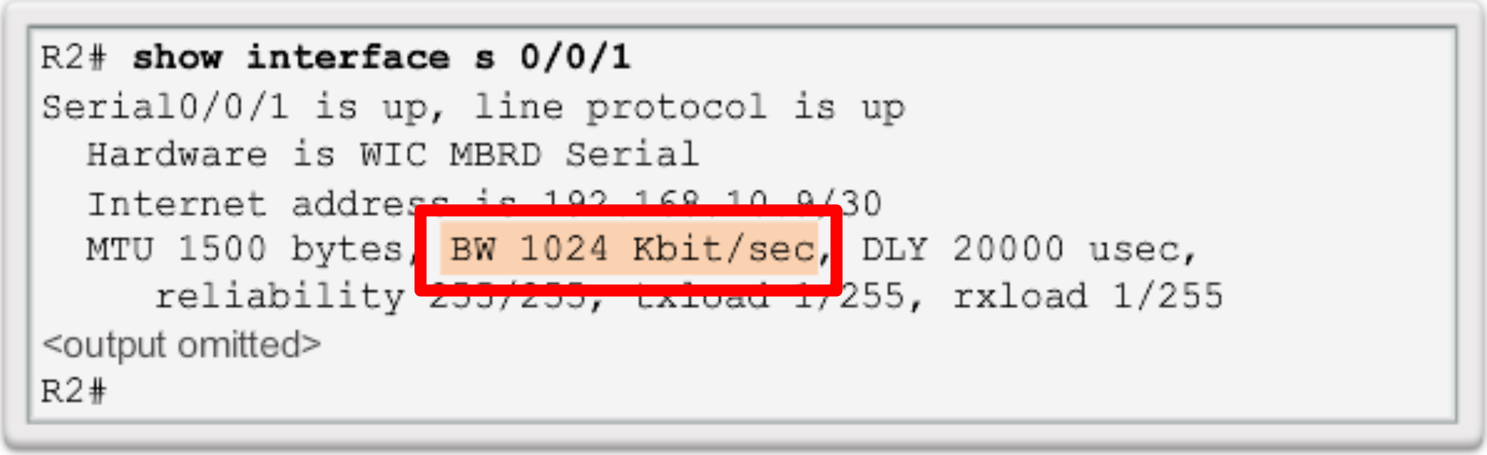
- Using the default values for K1 and K3, you can simplify this calculation to:

slowest bandwidth (or minimum bandwidth)

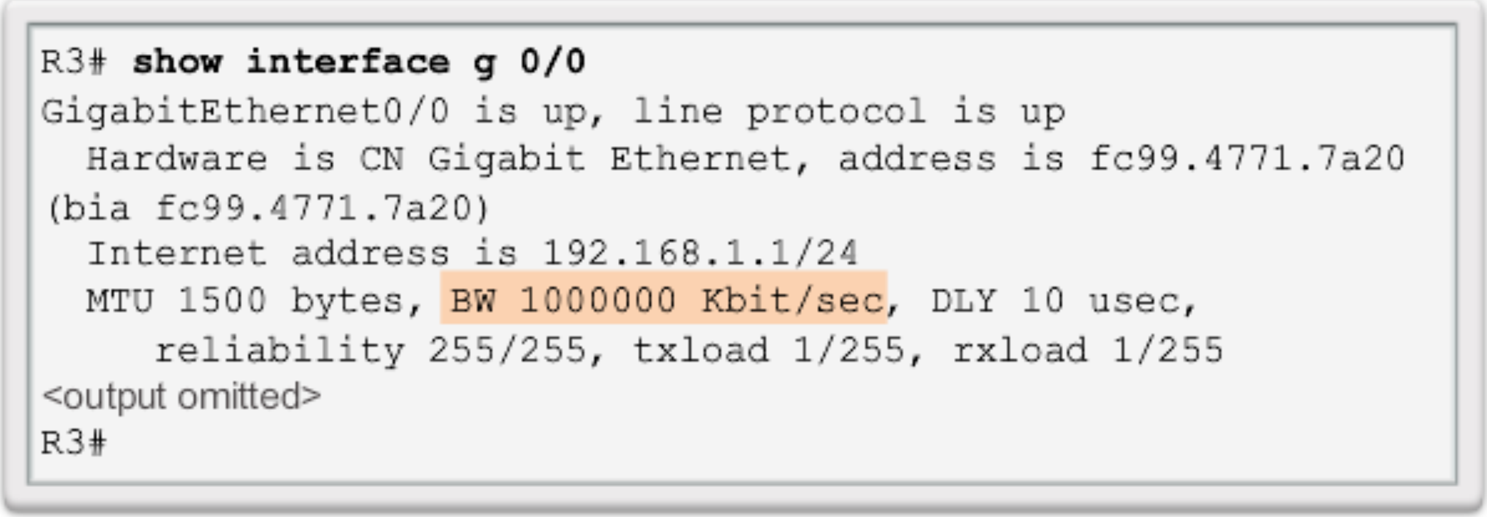
plus *the cumulative sum of all the delays*

EIGRP route metric

Slowest bandwidth of outgoing interfaces



```
R2# show interface s 0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Internet address is 192.168.10.9/30
  MTU 1500 bytes, BW 1024 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<output omitted>
R2#
```

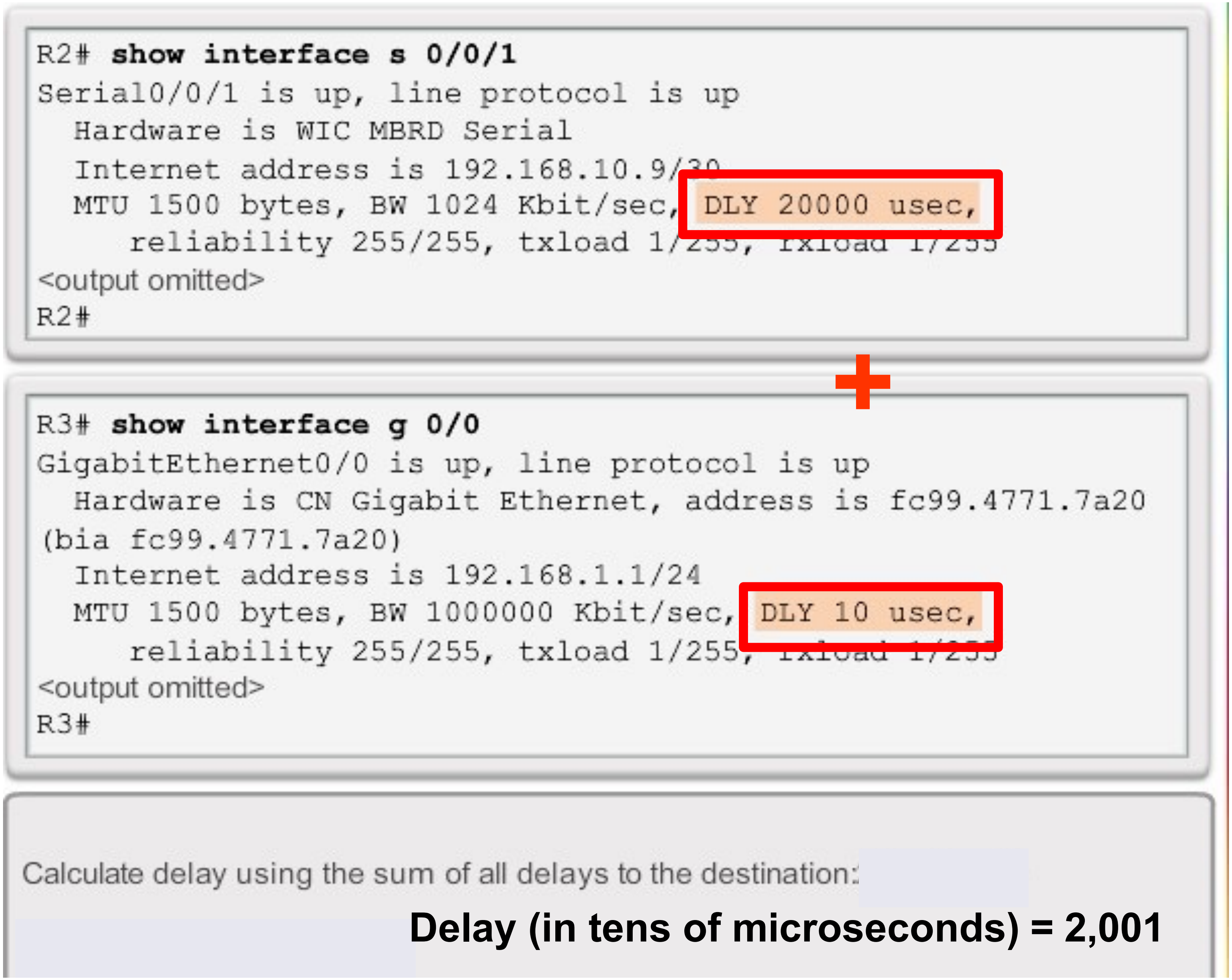


```
R3# show interface g 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is fc99.4771.7a20
(bia fc99.4771.7a20)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<output omitted>
R3#
```

Calculate bandwidth using the slowest bandwidth to the destination:

Normalized BW (10^7/MinimumBW) = 9,765

Sum of the delays



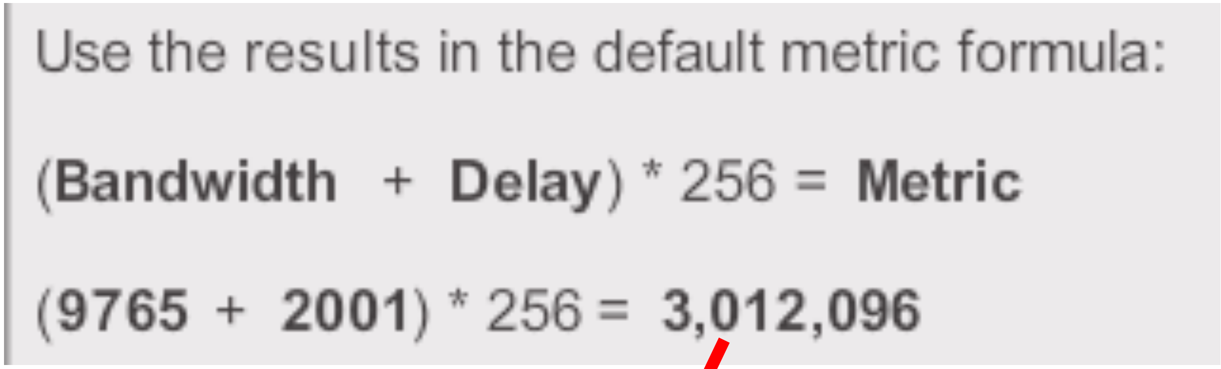
```
R2# show interface s 0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is WIC MBRD Serial
  Internet address is 192.168.10.9/30
  MTU 1500 bytes, BW 1024 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<output omitted>
R2#
```

```
R3# show interface g 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is fc99.4771.7a20
(bia fc99.4771.7a20)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<output omitted>
R3#
```

Calculate delay using the sum of all delays to the destination:

Delay (in tens of microseconds) = 2,001

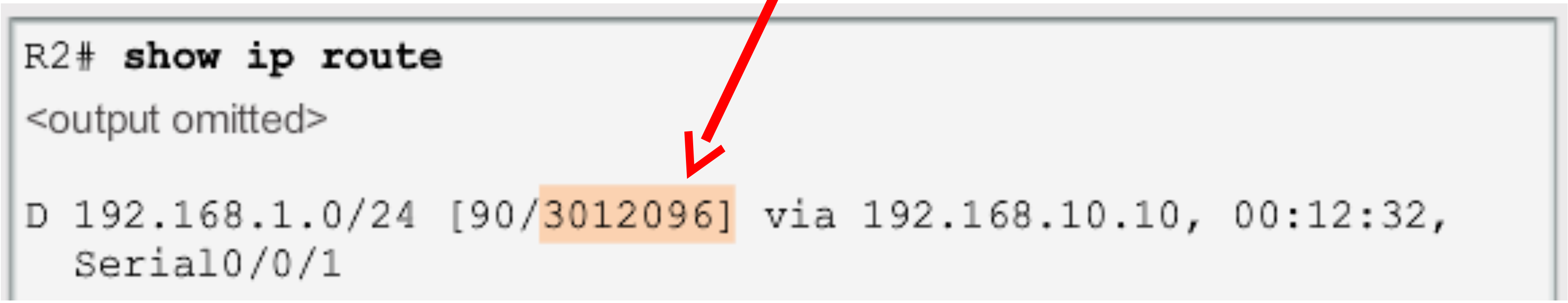
(Slowest BW + Sum of Delays) * 256



Use the results in the default metric formula:

(**Bandwidth** + **Delay**) * 256 = **Metric**

(**9765** + **2001**) * 256 = **3,012,096**



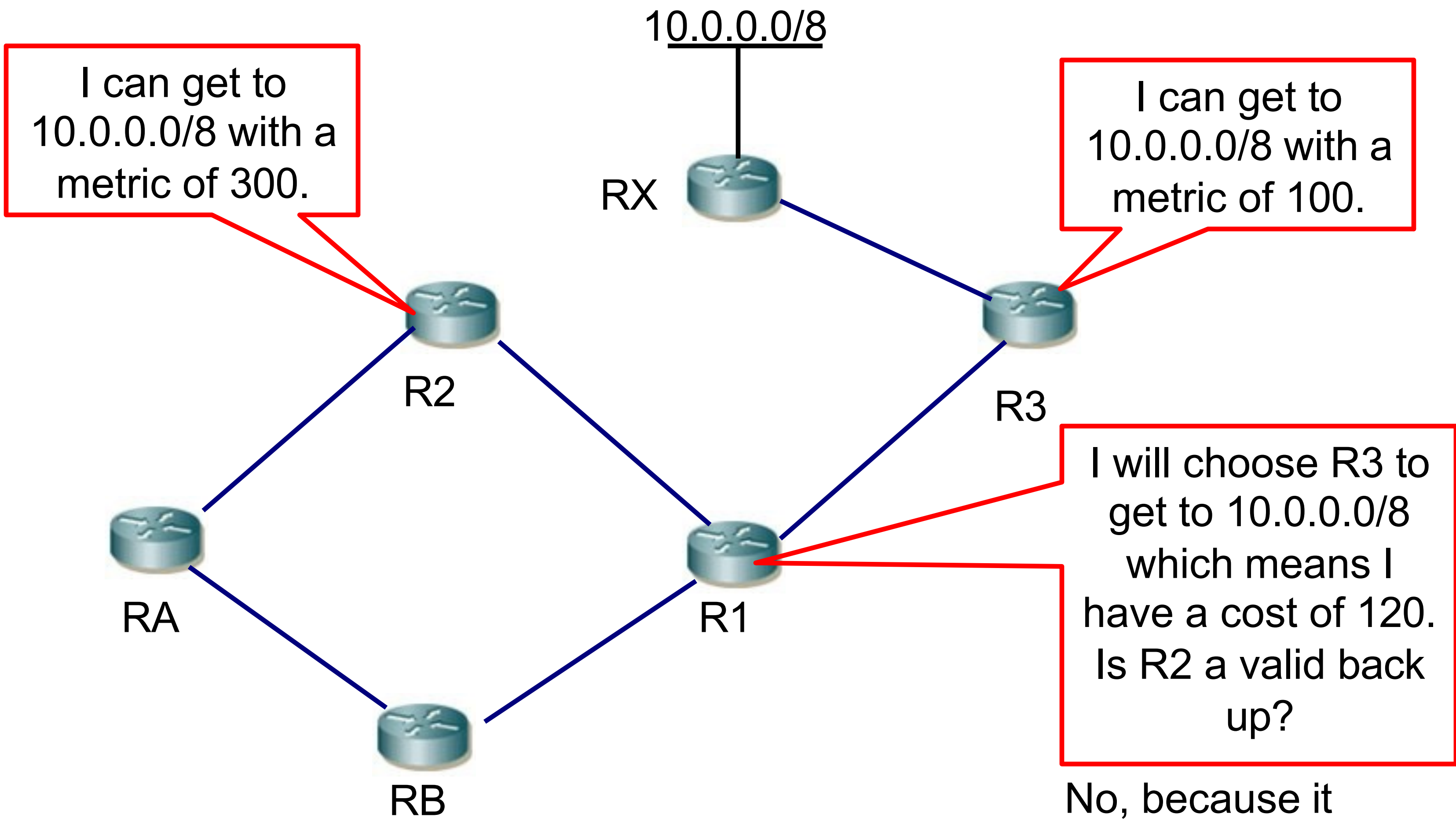
```
R2# show ip route
<output omitted>

D 192.168.1.0/24 [90/3012096] via 192.168.10.10, 00:12:32,
  Serial0/0/1
```

DUAL

- Successor
- Feasible Distance (FD)
- Feasible Successor (FS)
- Reported Distance (RD) or Advertised Distance (AD)
- Feasible Condition or Feasibility Condition (FC)

DUAL Concepts

- **Diffusing Update Algorithm** is the algorithm used by EIGRP.
- Determines:
  - Best loop-free path
  - Loop-free backup paths (which can be used immediately)
- DUAL also provides the following:
  - Fast convergence
  - Minimum bandwidth usage with bounded updates
- DUAL uses several terms that are discussed in more detail throughout this section:
  - **Successor**
  - **Feasible distance**
  - **Feasible successor**
  - **Reported distance** or advertised distance
  - **Feasible condition** or feasibility condition

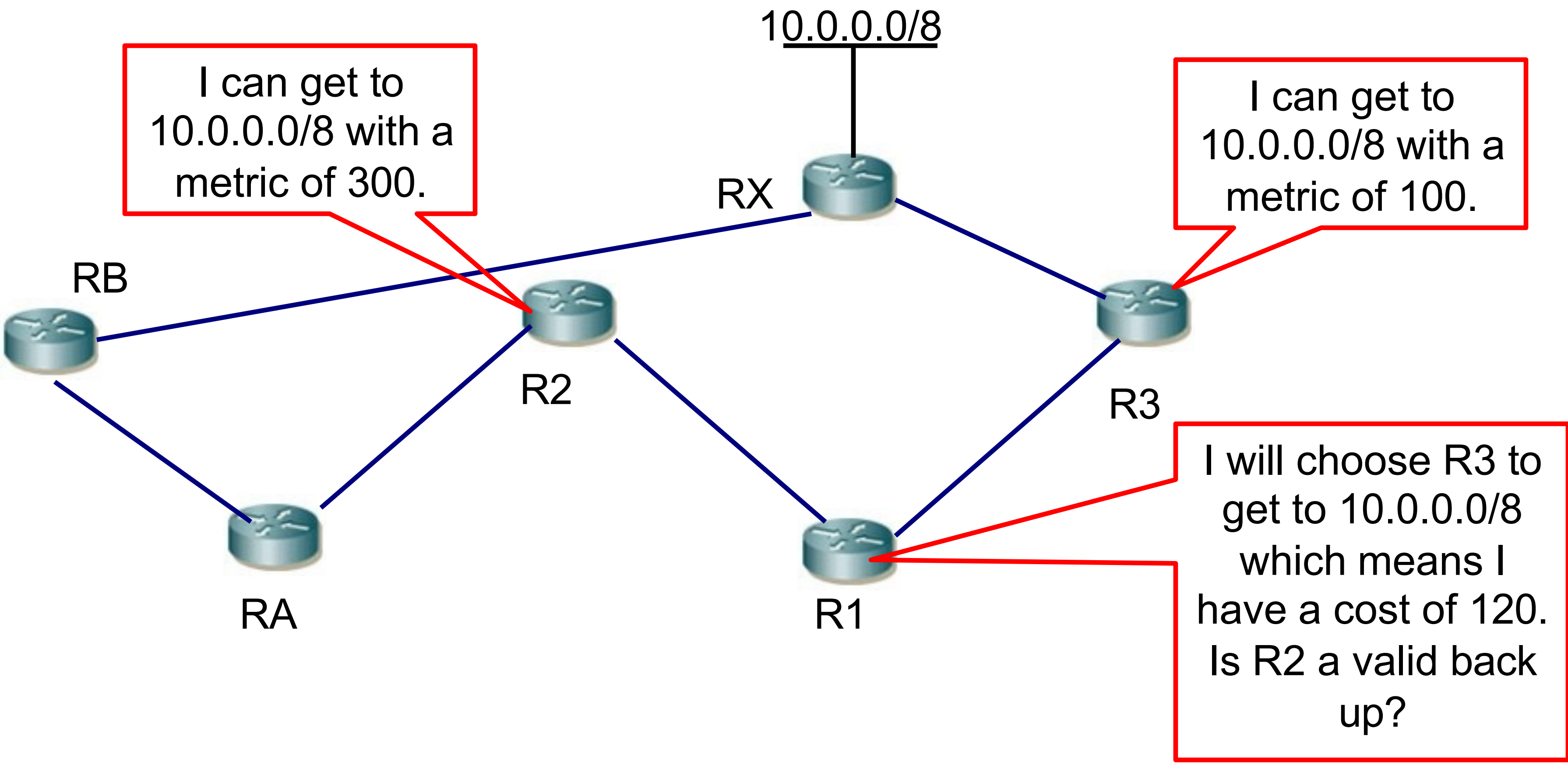


I can get to 10.0.0.0/8 with a metric of 300.

I can get to 10.0.0.0/8 with a metric of 100.

I will choose R3 to get to 10.0.0.0/8 which means I have a cost of 120. Is R2 a valid back up?

No, because it comes back through me R1 (loop). It only knows this route because of me.

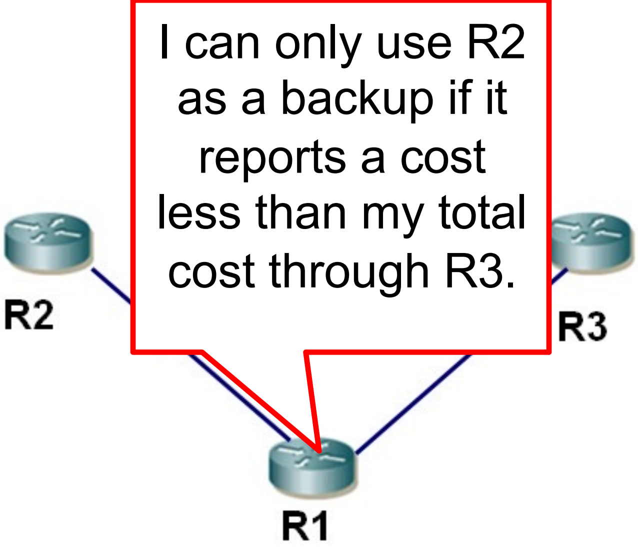


I can get to 10.0.0.0/8 with a metric of 300.

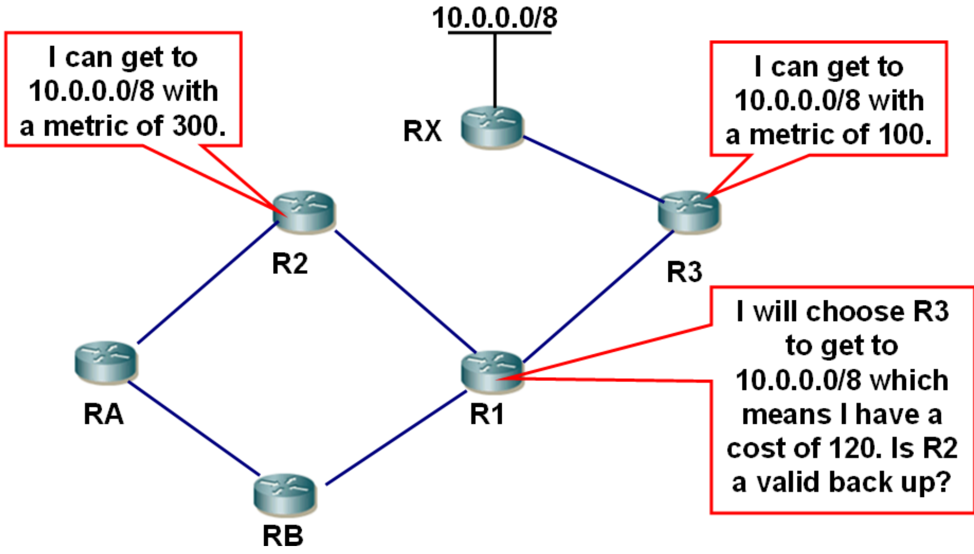
I can get to 10.0.0.0/8 with a metric of 100.

I will choose R3 to get to 10.0.0.0/8 which means I have a cost of 120. Is R2 a valid back up?

Yes, because it has its own path to 10.0.0.0/8. (no loop)



I can only use R2 as a backup if it reports a cost less than my total cost through R3.

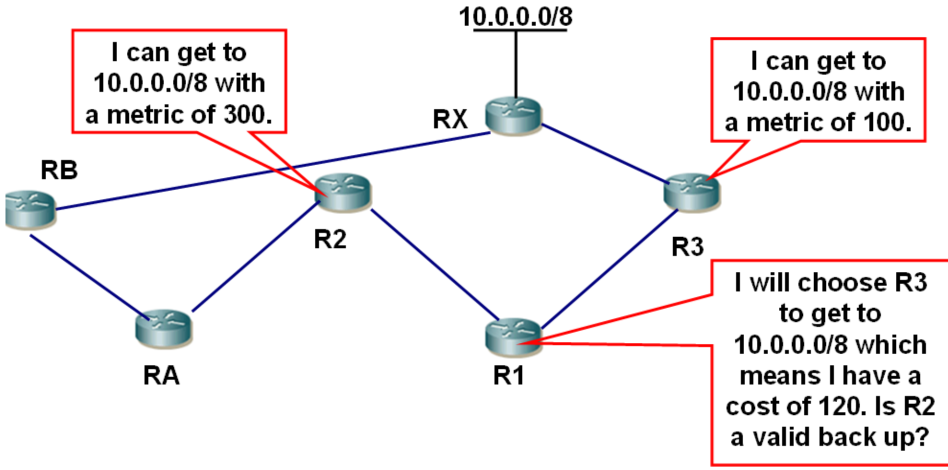


I can get to 10.0.0.0/8 with a metric of 300.

I can get to 10.0.0.0/8 with a metric of 100.

I will choose R3 to get to 10.0.0.0/8 which means I have a cost of 120. Is R2 a valid back up?

- EIGRP is a distance vector routing protocol.
- Does not see any topology map.
- Can’t tell if there is a loop or not?
- To play it safe, EIGRP only accepts a backup route if it meets the Feasibility Condition (coming).
- R1 will only use R2 if R2’s metric to 10.0.0.0/8 is less than R1’s cost through R3.



I can get to 10.0.0.0/8 with a metric of 300.

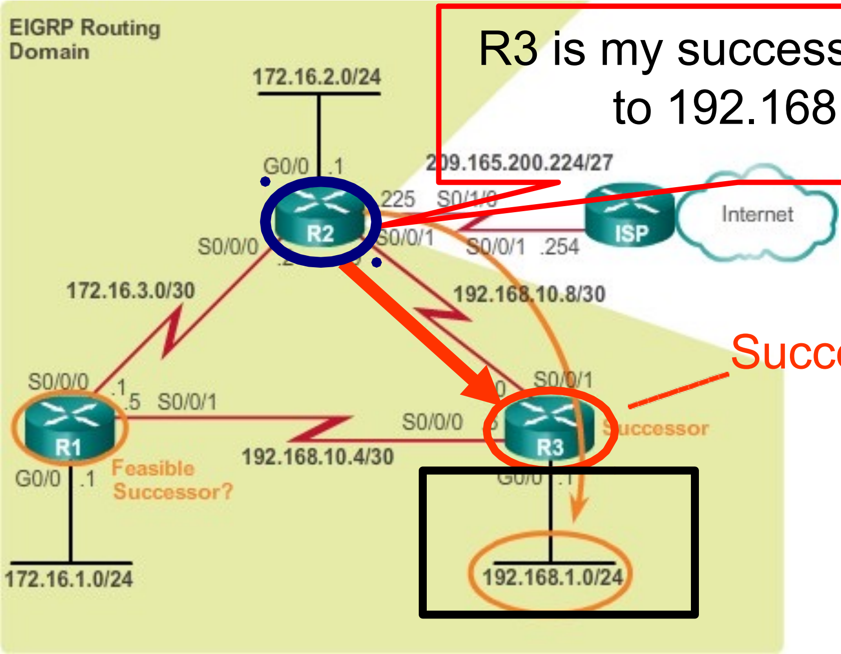
I can get to 10.0.0.0/8 with a metric of 100.

I will choose R3 to get to 10.0.0.0/8 which means I have a cost of 120. Is R2 a valid back up?

Successor and Feasible Distance

```
R2# show ip route
<code output omitted>

D   192.168.1.0/24 [90/3012096] via 192.168.10.10, 00:12:32, Serial0/0/1
```

IP address of the successor



R3 is my successor for getting to 192.168.1.0/24

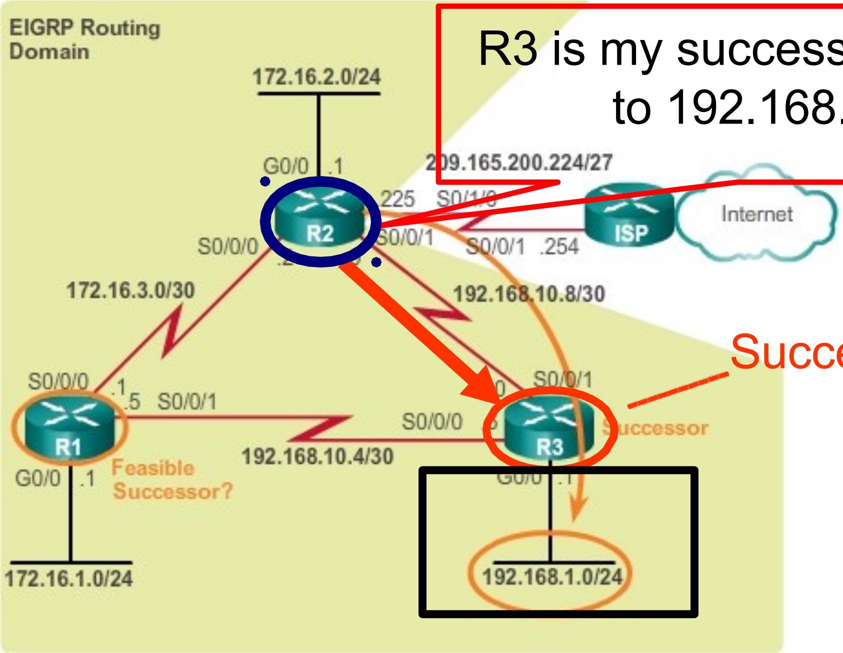
Successor

- A ***successor*** is a neighboring router that is used for packet forwarding and is the least-cost route to the destination network.
- What router is the successor for R2 for this network?
  - R3

Successor and Feasible Distance

```
R2# show ip route
<code output omitted>

D   192.168.1.0/24 [90/3012096] via 192.168.10.10, 00:12:32, Serial0/0/1
```

Feasible Distance

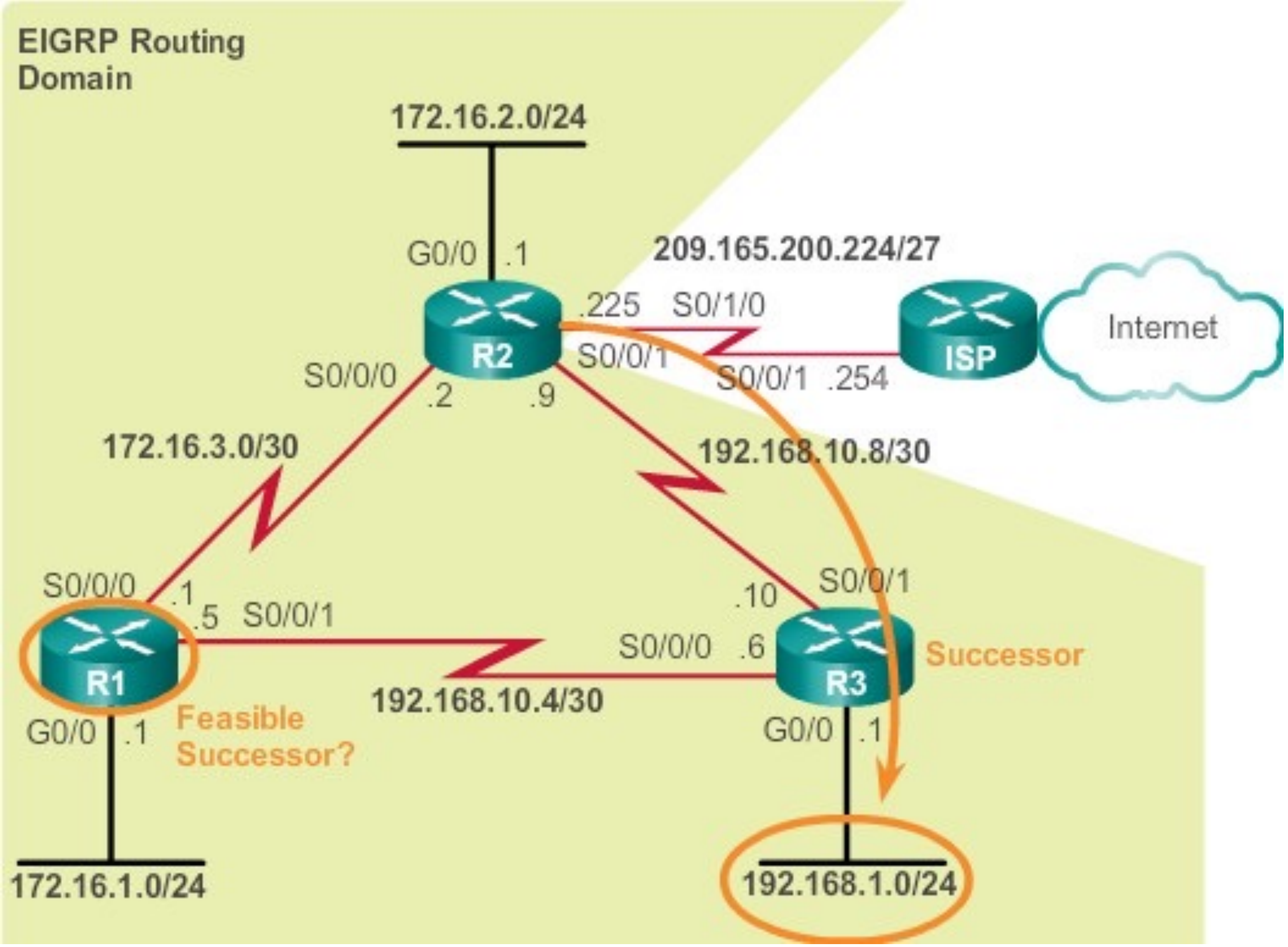


R3 is my successor for getting to 192.168.1.0/24

Successor

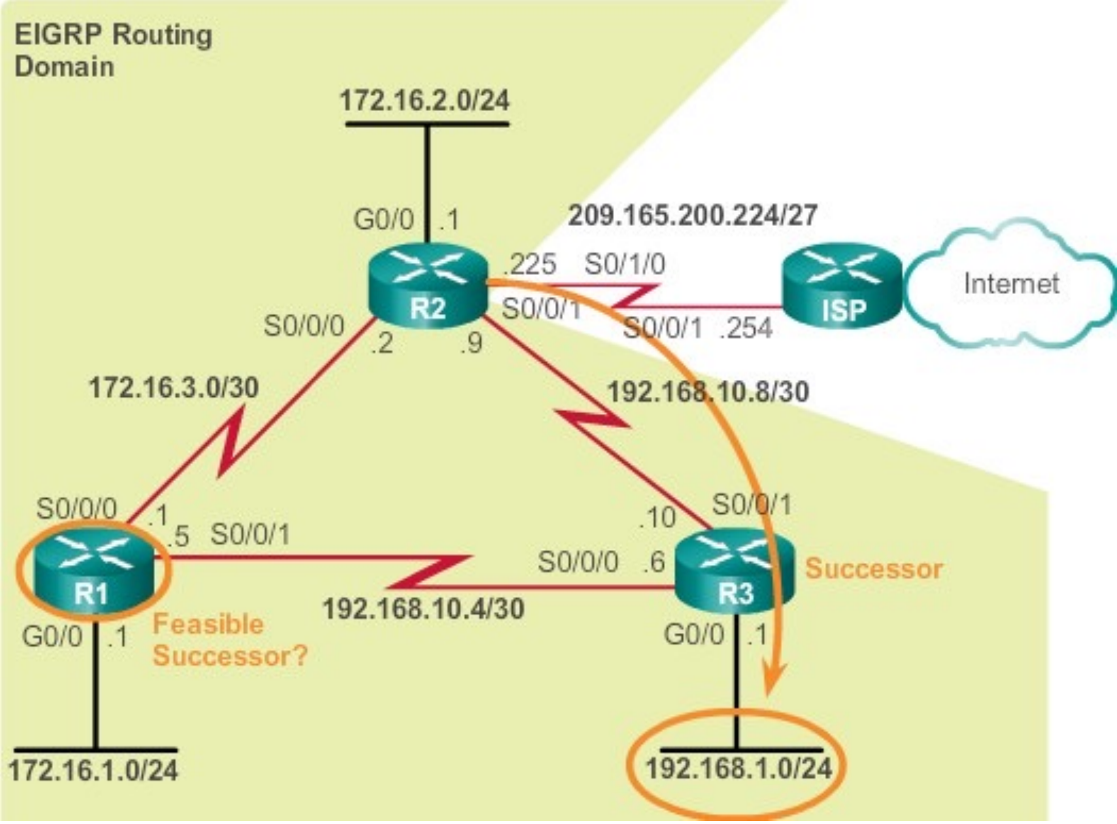
- ***Feasible distance (FD)*** is the lowest calculated metric to reach the destination network.
  - The metric for the route.

Is R1 a Feasible Successor?



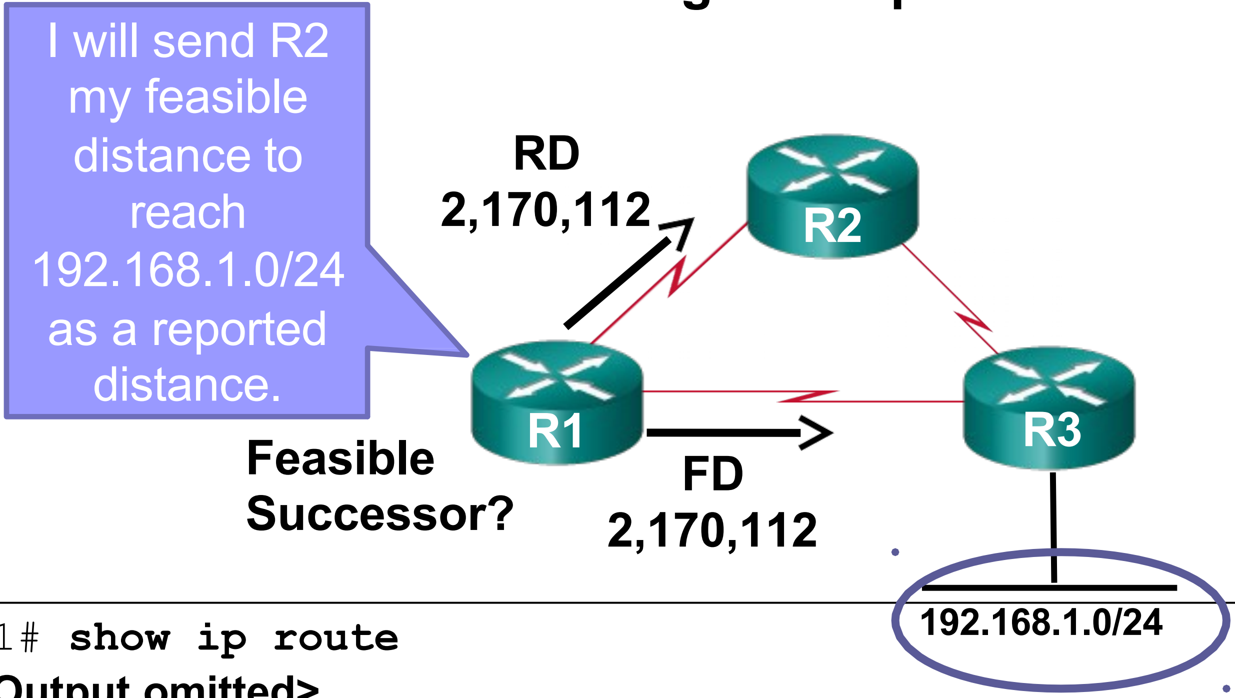
- *Is R1 a Feasible Successor?*
- *Does R2 know if R1 has a loop-free backup path to 192.168.1.0/24?*
- Remember, EIGRP is a Distance Vector Routing protocol.

Feasible Successor



- A ***feasible successor (FS)*** is a neighbor who has a loop-free backup path to the same network as the successor by satisfying the **feasibility condition**.
- Would R2 consider R1 to be a feasible successor to network 192.168.1.0/24?
- To be a feasible successor, **R1** must satisfy the ***feasibility condition (FC)***.

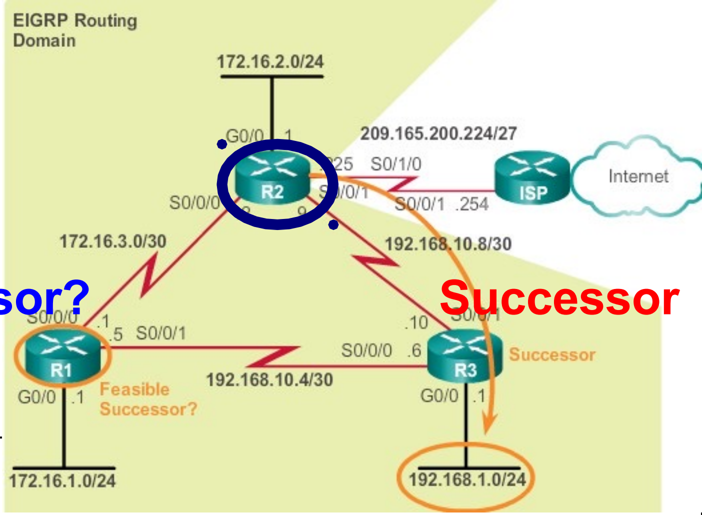
Sending the Reported Distance


I will send R2 my feasible distance to reach 192.168.1.0/24 as a reported distance.
RD
2,170,112
R2
R1
R3
Feasible Successor?
FD
2,170,112
192.168.1.0/24

```
R1# show ip route
<Output omitted>

D     192.168.1.0/24 [90/2170112] via 192.168.10.6, 02:44:50, Serial0/0/1
```

- The **reported distance** (advertised distance) - EIGRP neighbor's FD to the same destination network.
  - The metric that a router reports to a neighbor about its own cost to that network.
- **Feasibility Condition**: The FC is met when a neighbor's ***reported distance (RD)*** to a network is less than the local router's FD to the same destination network.

- **Feasibility Condition**: The FC is met when a neighbor's ***reported distance (RD)*** to a network is less than the local router's FD to the same destination network.



Feasible Successor? **Successor**

Does R1 meet the feasibility condition?

```
R2# show ip route

D     192.168.1.0/24 [90/3012096] via 192.168.10.10, 00:12:32, Serial0/0/1
```

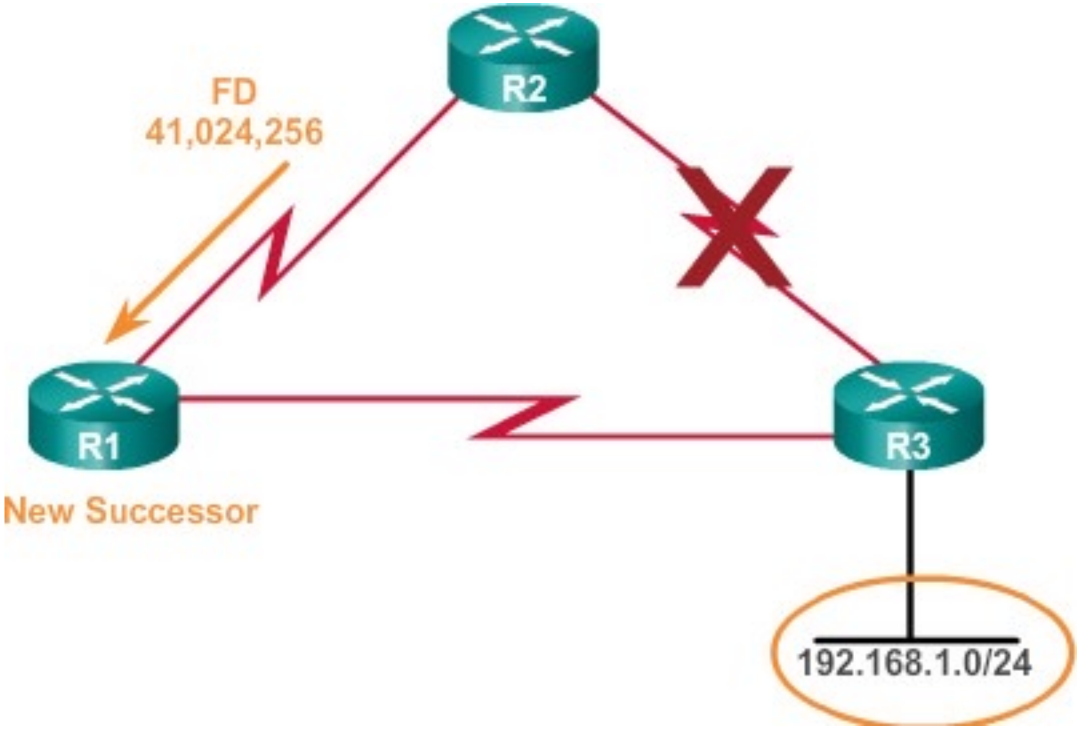
Feasible Distance **Successor (R3)**

```
R1# show ip route

D     192.168.1.0/24 [90/2170112] via 192.168.10.6, 02:44:50, Serial0/0/1
```

Feasible Distance
Sent to R2 as R1's Reported Distance

- **R2's feasible distance to 192.168.1.0 is 3,012,096**
- **R1's reported distance to 192.168.1.0 is 2,170,112**
- **R1's reported distance 2170112 is less that R2's feasible distance 3012096 so…**
- **R1 meets the feasibility condition.**

Using the Feasible Successor as New Successor



```
R2# show ip route

D      192.168.1.0/24 [90/41024256] via 172.16.3.1, 00:00:13, Serial0/0/0
```

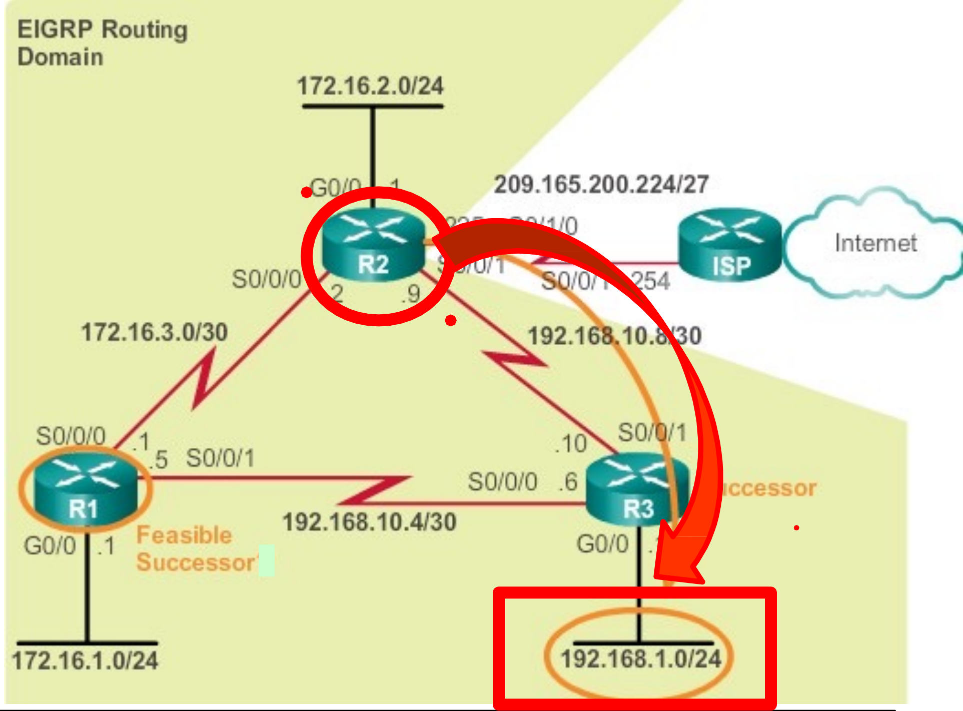
Feasible Distance   Successor (R1)

R2’s Topology Table

```
R2# show ip eigrp topology
EIGRP-IPv4 Topology Table for AS(1)/ID(2.2.2.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 172.16.2.0/24, 1 successors, FD is 2816
        via Connected, GigabitEthernet0/0
P 192.168.10.4/30, 1 successors, FD is 3523840
        via 192.168.10.10 (3523840/2169856), Serial0/0/1
        via 172.16.3.1 (41024000/2169856), Serial0/0/0
P 192.168.1.0/24, 1 successors, FD is 3012096
        via 192.168.10.10 (3012096/2816), Serial0/0/1
        via 172.16.3.1 (41024256/2170112), Serial0/0/0
P 172.16.3.0/30, 1 successors, FD is 40512000
        via Connected, Serial0/0/0
P 172.16.1.0/24, 1 successors, FD is 3524096
        via 192.168.10.10 (3524096/2170112), Serial0/0/1
        via 172.16.3.1 (40512256/2816), Serial0/0/0
P 192.168.10.8/30, 1 successors, FD is 3011840
        via Connected, Serial0/0/1

R2#
```

- The **successor**, **feasible distance**, and any **feasible successors** with their **reported distances** are kept by a router in its EIGRP topology table or ***topology database***.

Examining an Entry in the Topology Table



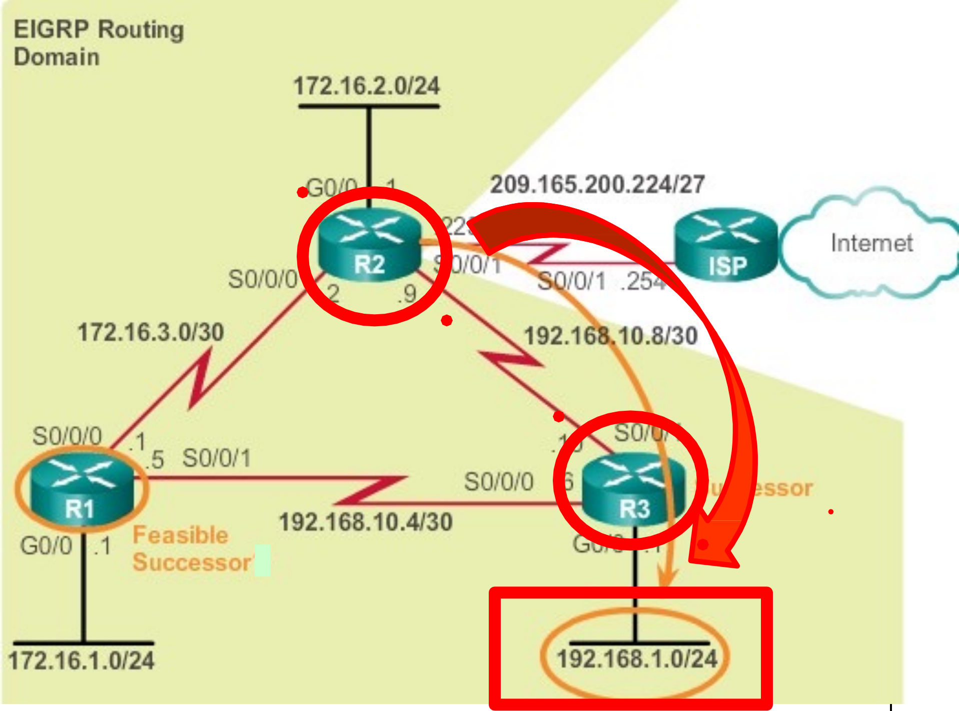
```
R2# show ip eigrp topology
<Output omitted>

P 192.168.1.0/24, 1 successors, FD is 3012096
        via 192.168.10.10 (3012096/2816), Serial0/0/1
        via 172.16.3.1 (41024256/2170112), Serial0/0/0
```

Destination network

Feasible distance

Indicates passive or active state

Number of successors



The Successor

```
R2# show ip eigrp topology
<Output omitted>

P 192.168.1.0/24, 1 successors, FD is 3012096
        via 192.168.10.10 (3012096/2816), Serial0/0/1
        via 172.16.3.1 (41024256/2170112), Serial0/0/0
```

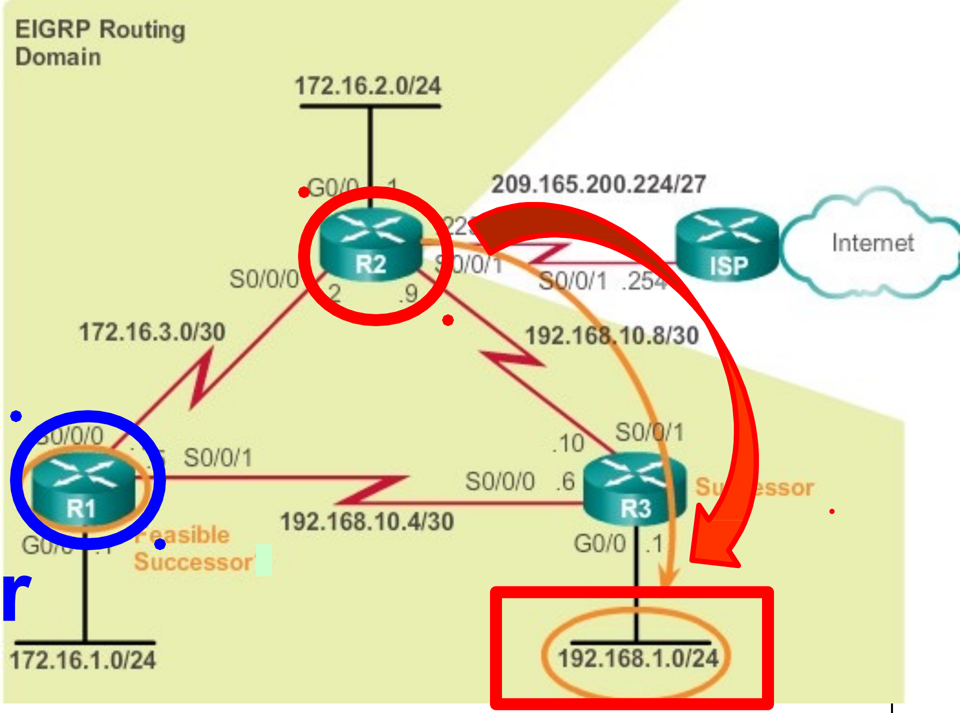
Next hop address of the successor

Feasible distance

Successor's (R3) Reported Distance

Outbound interface to reach this network

The Feasible Successor



```
R2#show ip eigrp topology
<Output omitted>

P 192.168.1.0/24, 1 successors, FD is 3012096
        via 192.168.10.10 (3012096/2816), Serial0/0/1
        via 172.16.3.1 (41024256/2170112), Serial0/0/0
```

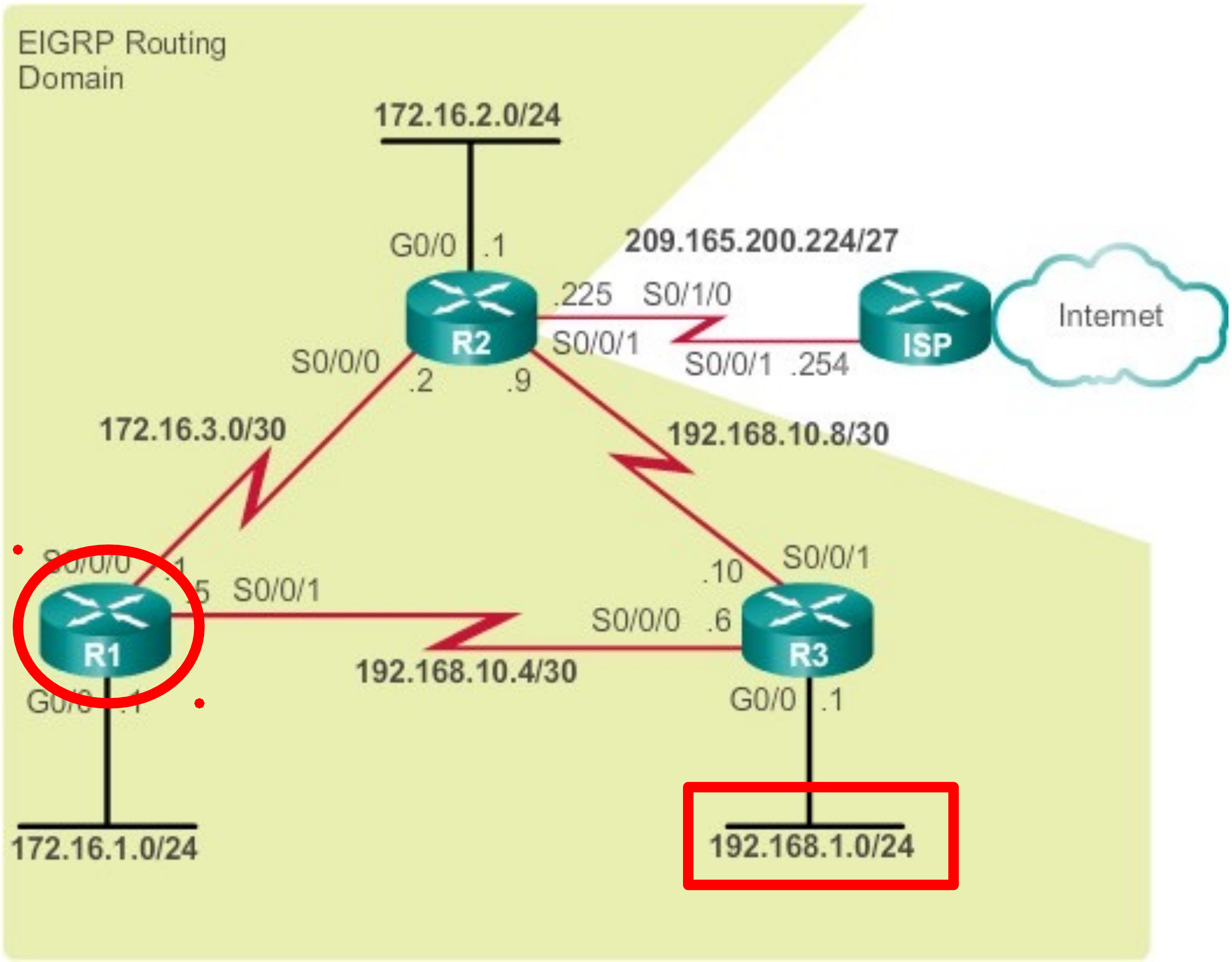
Next hop address of the feasible successor (R1)

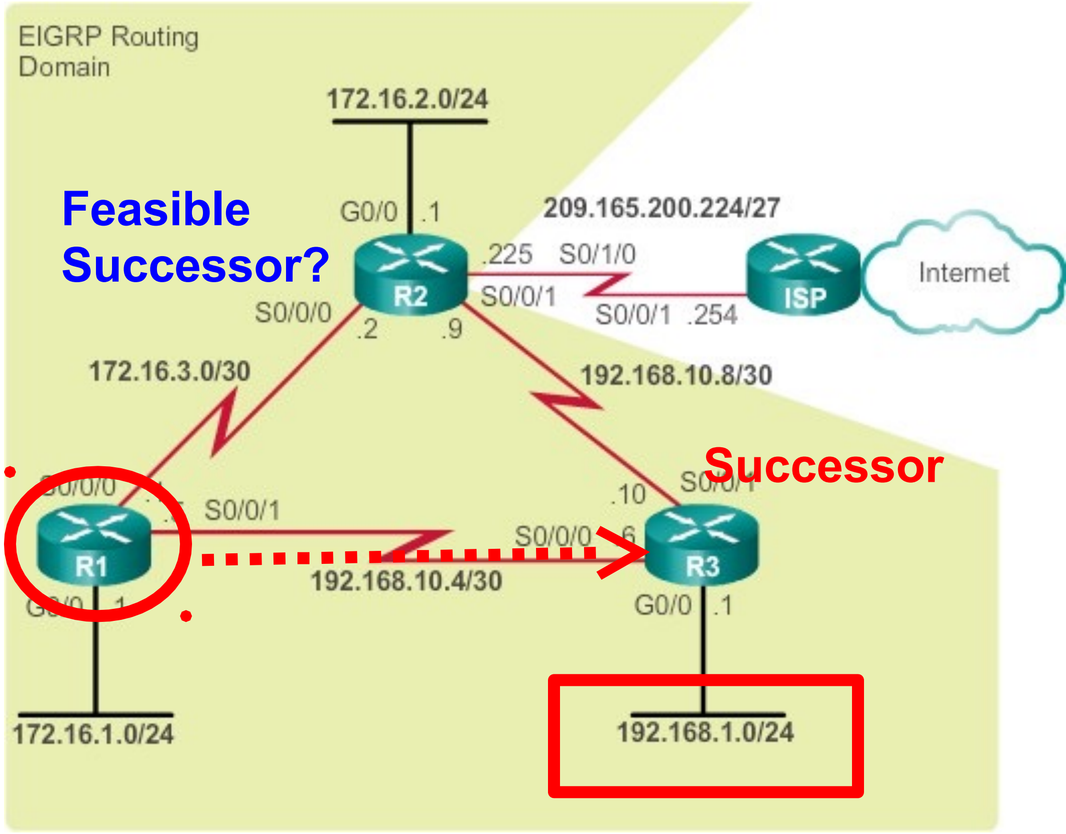
Feasible distance if the feasible successor (R1) was the successor

Feasible Successor's (R1) Reported Distance

Outbound interface to reach this network

Example: No Feasible Successor

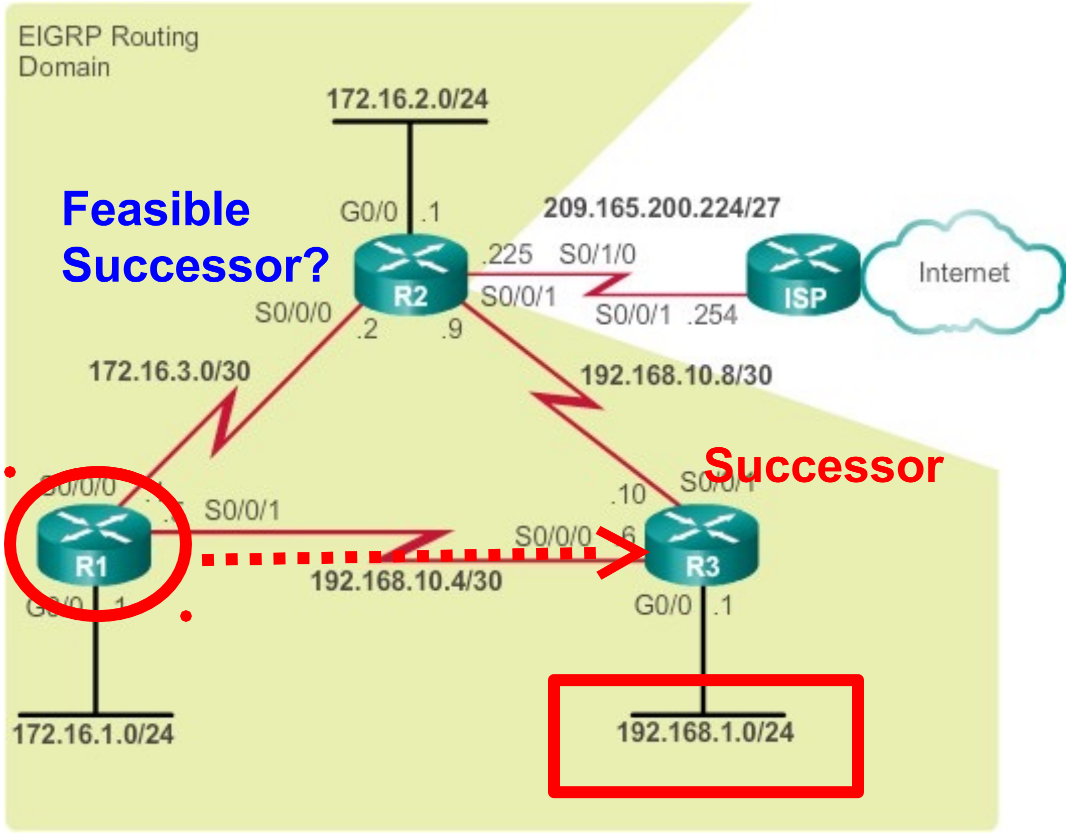




```
R1#show ip route
<Output omitted>

D      192.168.1.0/24 [90/2170112] via 192.168.10.6, 01:23:13, Serial0/0/1
```

Feasible Distance

Next-hop router (R3) is the successor



Feasible Successor?

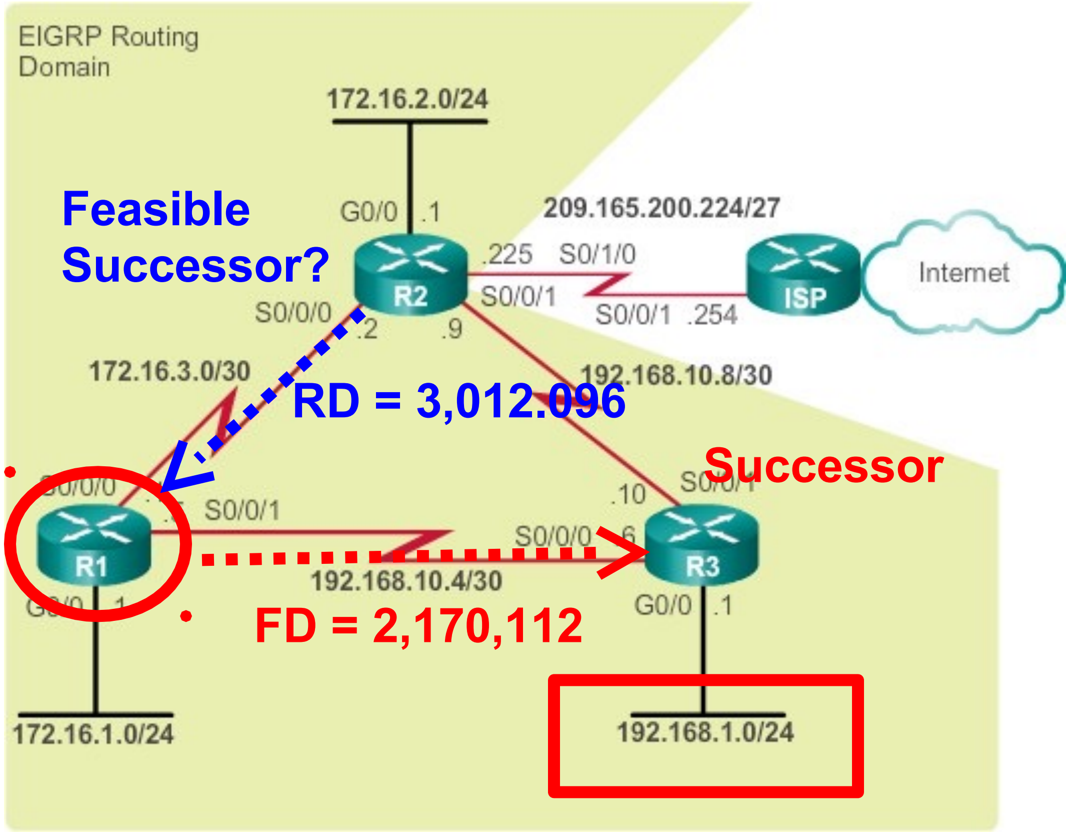
Successor

Topology Table:
No Feasible Successor

```
R1# show ip eigrp topology
<Output omitted>

P 192.168.1.0/24, 1 successors, FD is 2170112
        via 192.168.10.6 (2170112/2816), Serial0/0/1
```

Successor

- You can view all possible links whether they satisfy the feasible condition or not by adding the `[all-links]` option.
  - Even those routes that are not FSs.
- ***Is R2 a Feasible Successor?***
  - Does R2 meet the Feasibility Condition?
  - **Is R2's RD less than R1's FD?**
  - **No! So does not meet Feasibility Condition.**



Feasible Successor?

RD = 3,012,096

Successor

FD = 2,170,112

```
R1# show ip eigrp topology all-links

P 192.168.1.0/24, 1 successors, FD is 2170112, serno 9
        via 192.168.10.6 (2170112/2816), Serial0/0/1
        via 172.16.3.2 (41024256/3012096), Serial0/0/0
```

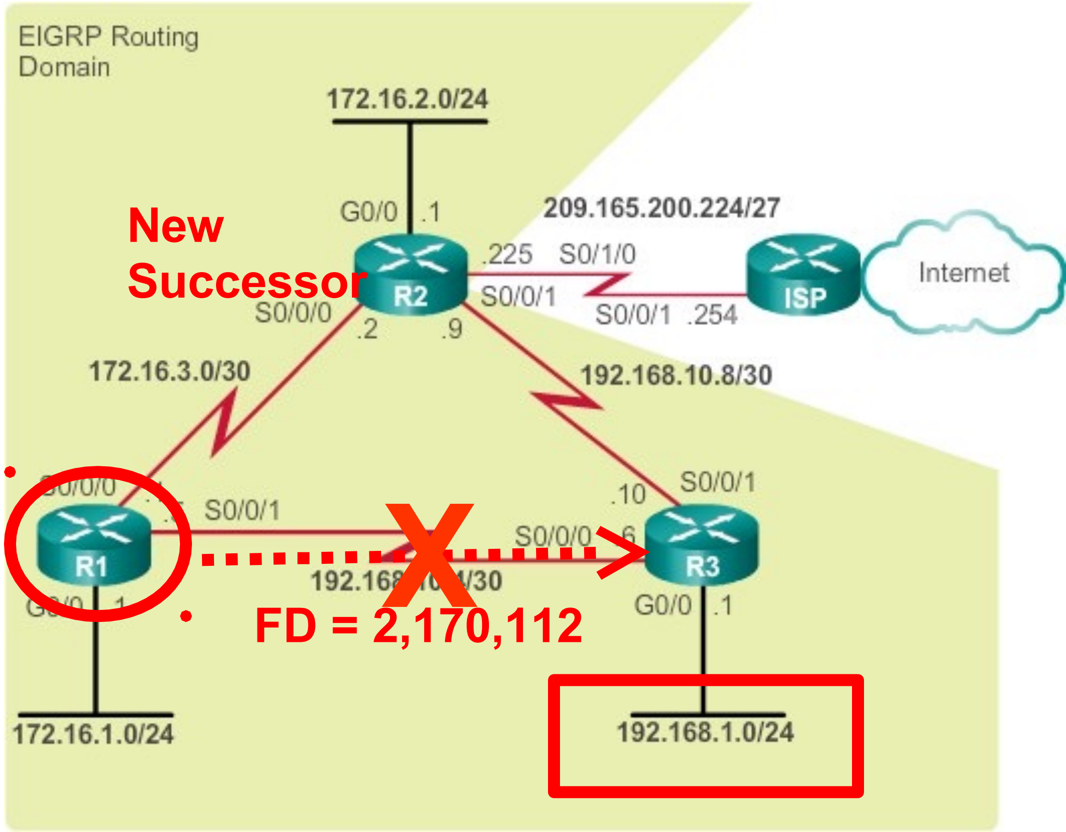
Successor

Not a feasible successor

R1's Feasible Distance

R2's Reported Distance

- ***Even though R2 looks like a viable backup path to 192.168.1.0/24, R1 has no idea that its path is not a potential loop back through itself.***
- *Does this mean R2 cannot be used if the successor fails?*
  - R2 can be used, but there will be a longer delay before adding it to the routing table.
  - Before this can happen, DUAL will need to do some further processing, which is explained in the next topic.



New Successor

FD = 2,170,112

```
R1# show ip eigrp topology all-links

P 192.168.1.0/24, 1 successors, FD is 2170112, serno 9
        via 192.168.10.6 (2170112/2816), Serial0/0/1
        via 172.16.3.2 (41024256/3012096), Serial0/0/0
```

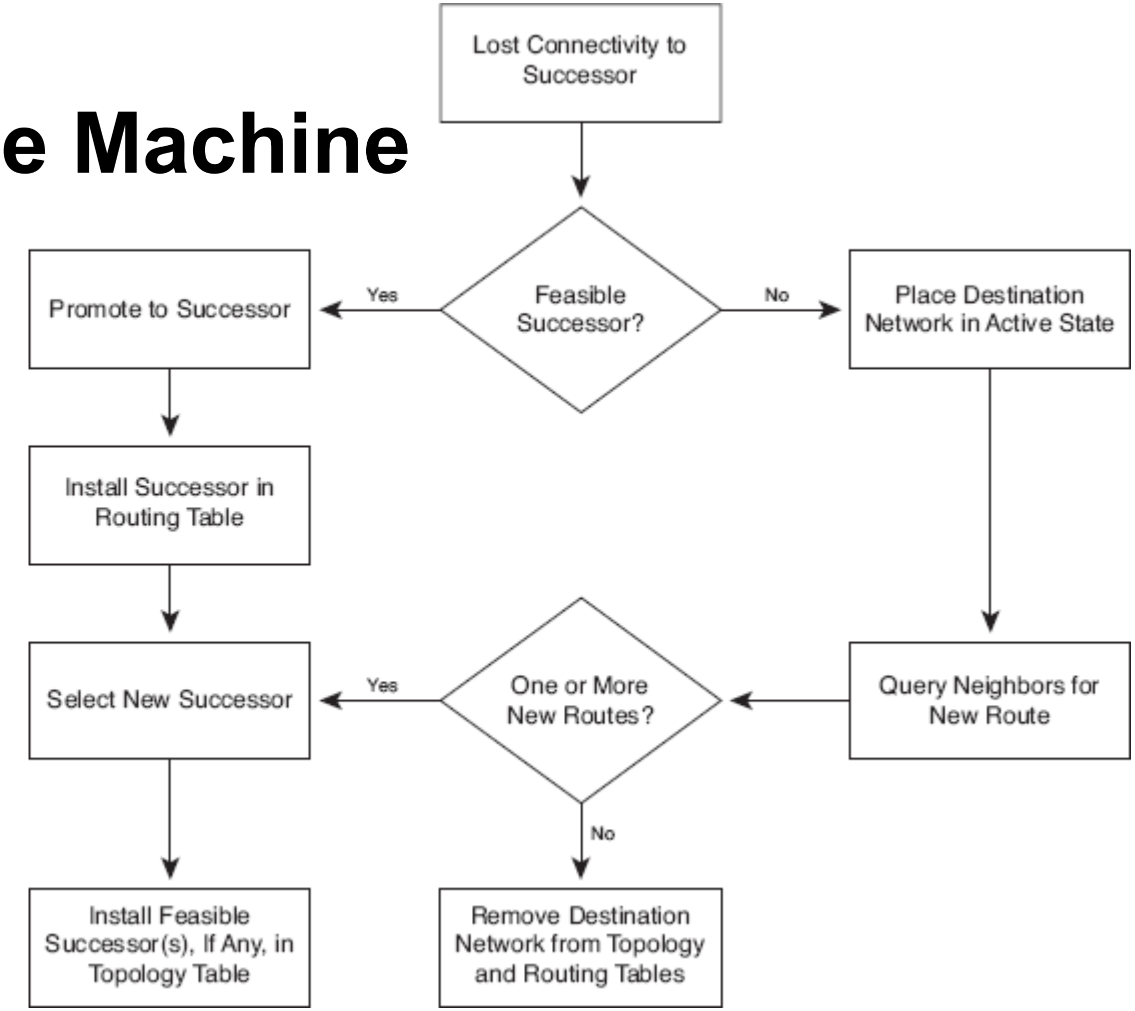
Successor

Not a feasible successor

R1's Feasible Distance

R2's Reported Distance

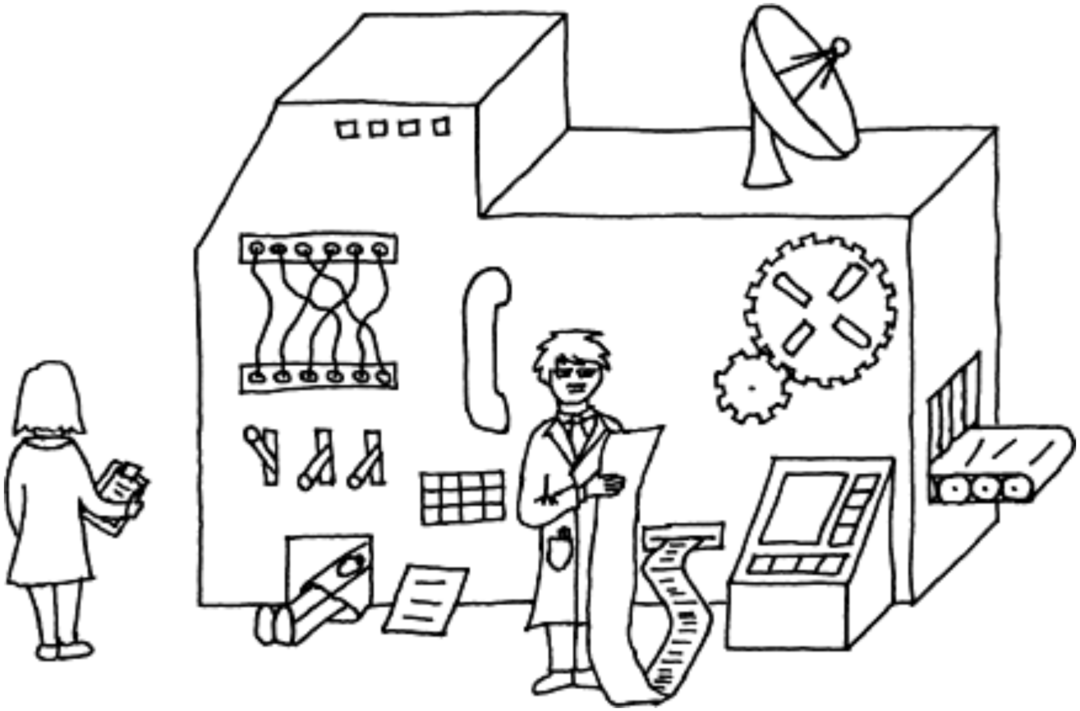
DUAL Finite State Machine



Lost Connectivity to Successor
Feasible Successor?
Yes
No
Promote to Successor
Place Destination Network in Active State
Install Successor in Routing Table
Query Neighbors for New Route
One or More New Routes?
Yes
Select New Successor
No
Install Feasible Successor(s), If Any, in Topology Table
Remove Destination Network from Topology and Routing Tables

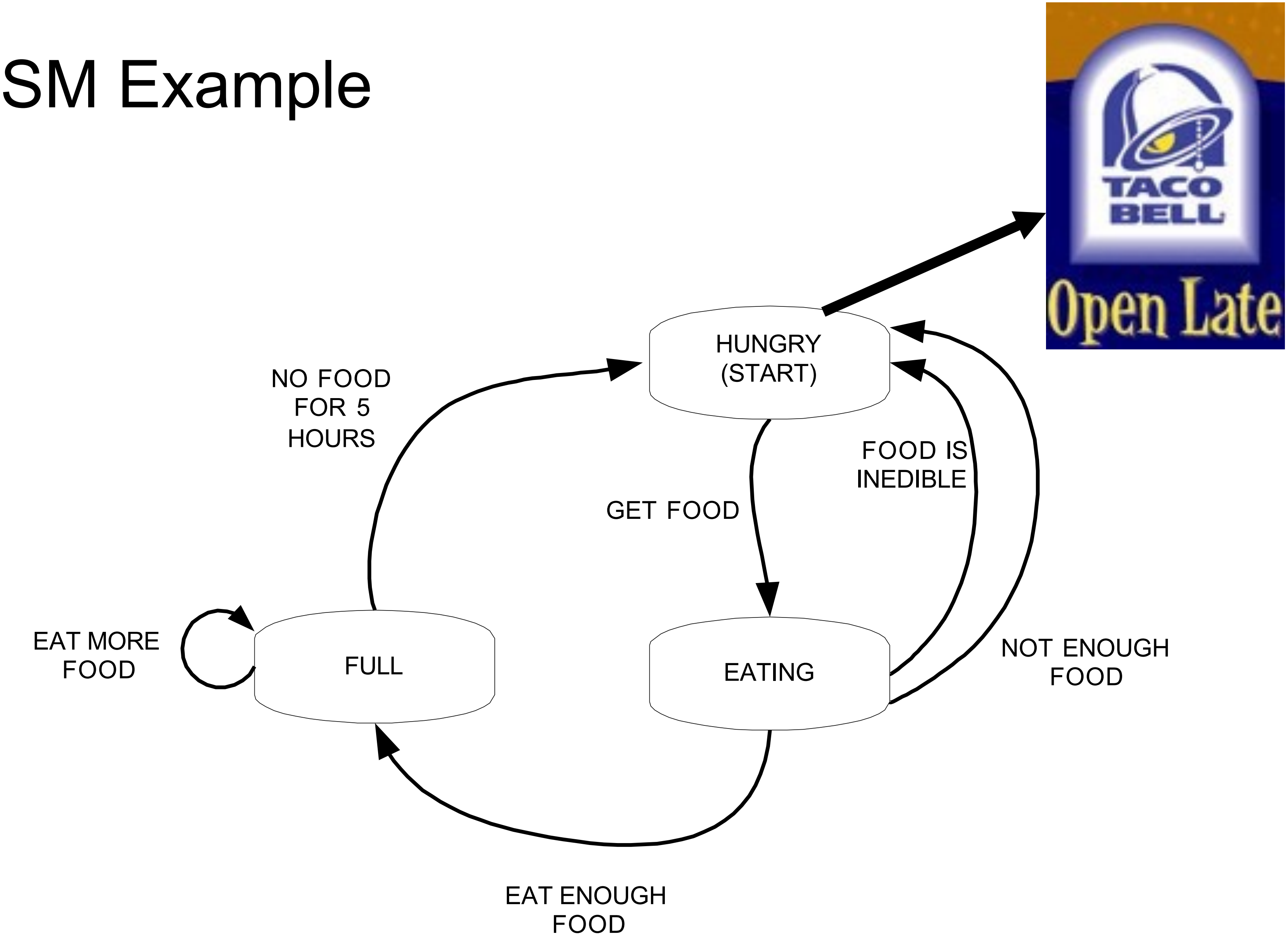
- The centerpiece of EIGRP is DUAL (EIGRP route-calculation engine).
  - **DUAL Finite State Machine (FSM)**
- This FSM contains all the logic used to calculate and compare routes in an EIGRP network.

DUAL FSM



- **FSMs** defines:
  - The set of possible states that something can go through
  - What events cause those states
  - What events result from those states
- *Beyond the scope of this course.*

FSM Example



HUNGRY (START)

NO FOOD FOR 5 HOURS

GET FOOD

FOOD IS INEDIBLE

EAT MORE FOOD

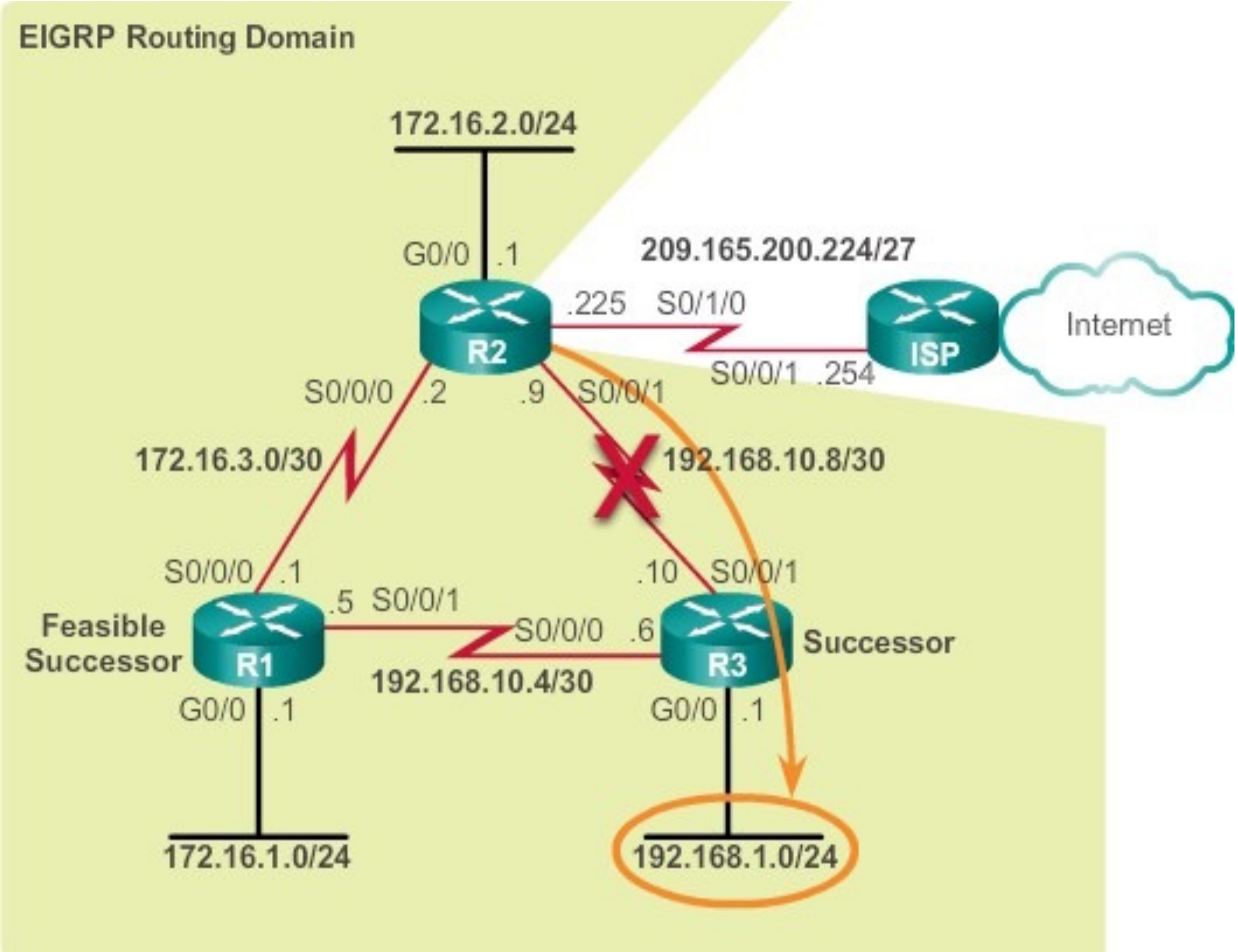
FULL

EATING

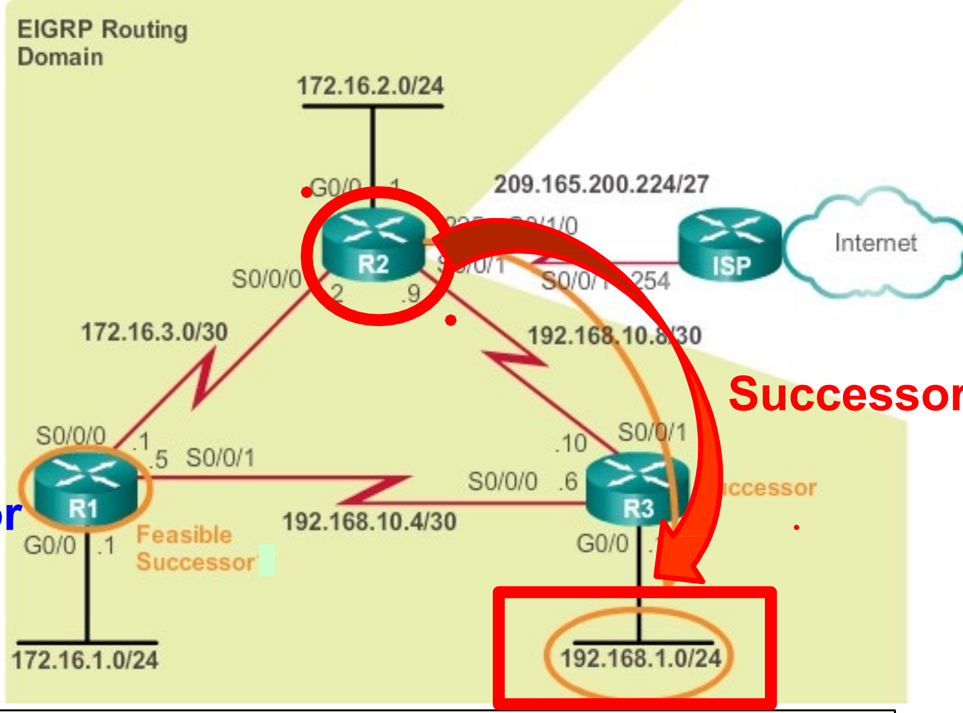
NOT ENOUGH FOOD

EAT ENOUGH FOOD

When there is a Feasible Successor –DUAL not needed



Successor and Feasible Successor

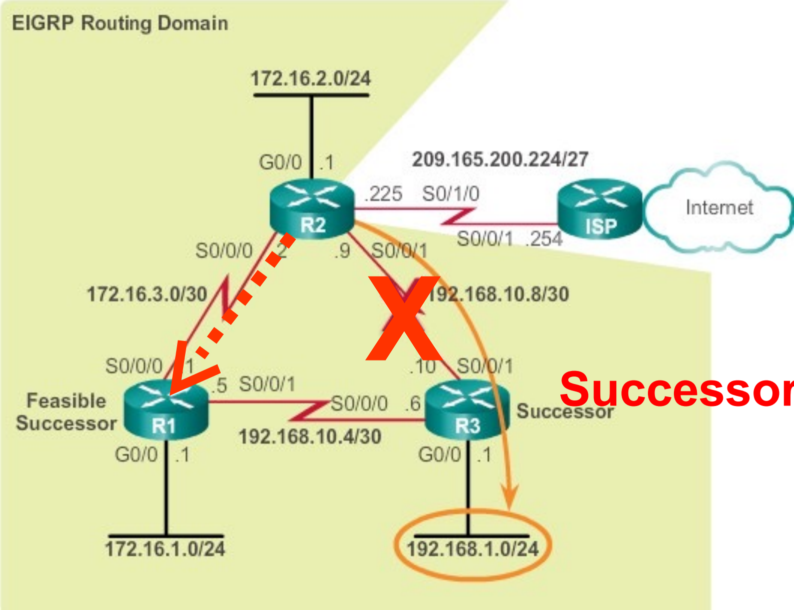


Successor

Feasible Successor

```
R2# show ip eigrp topology
<Output omitted>

P 192.168.1.0/24, 1 successors, FD is 3012096
        via 192.168.10.10 (3012096/2816), Serial0/0/1
        via 172.16.3.1 (41024256/2170112), Serial0/0/0
```

Creating a failure…
Has FS

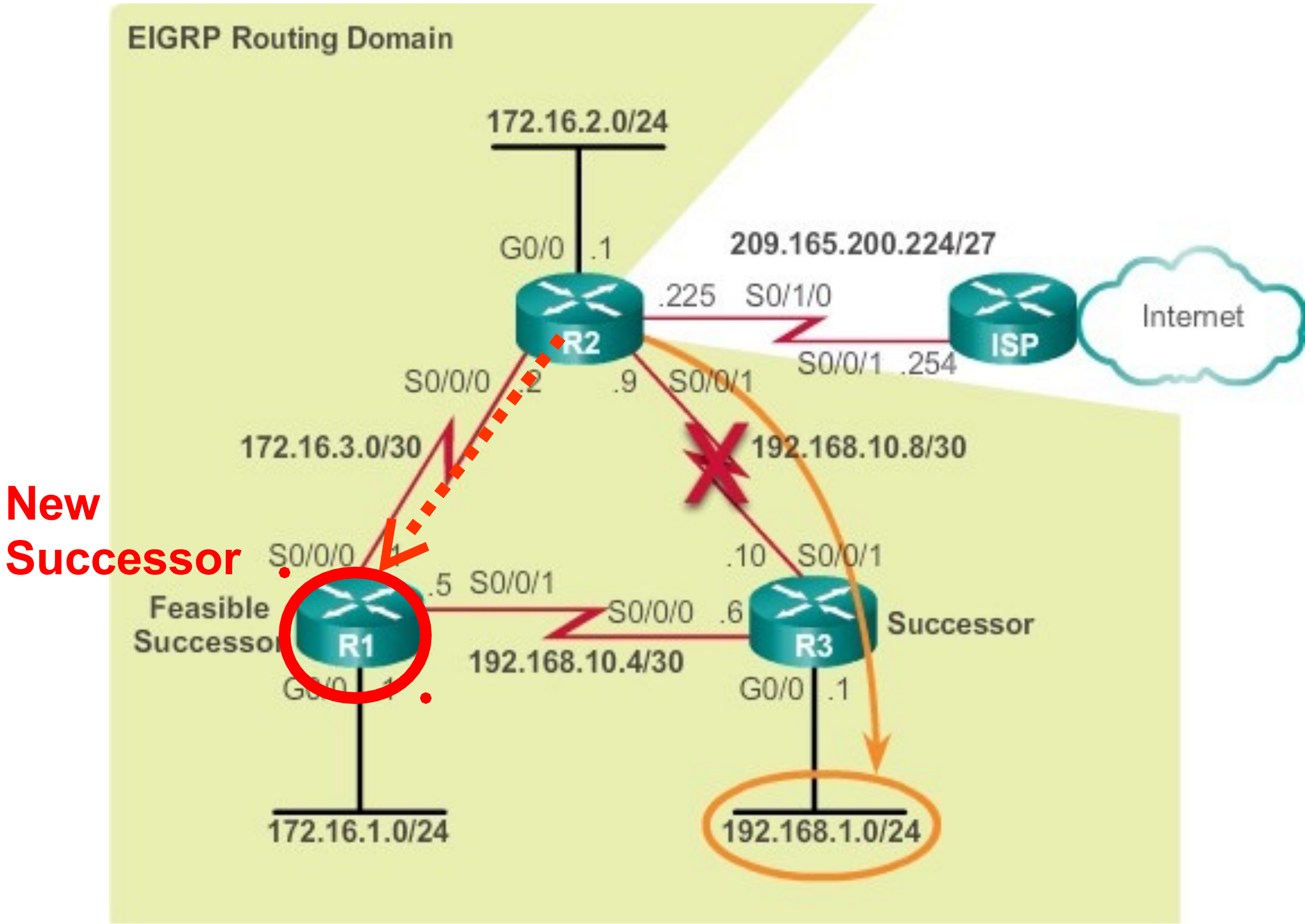


Feasible Successor

New Su

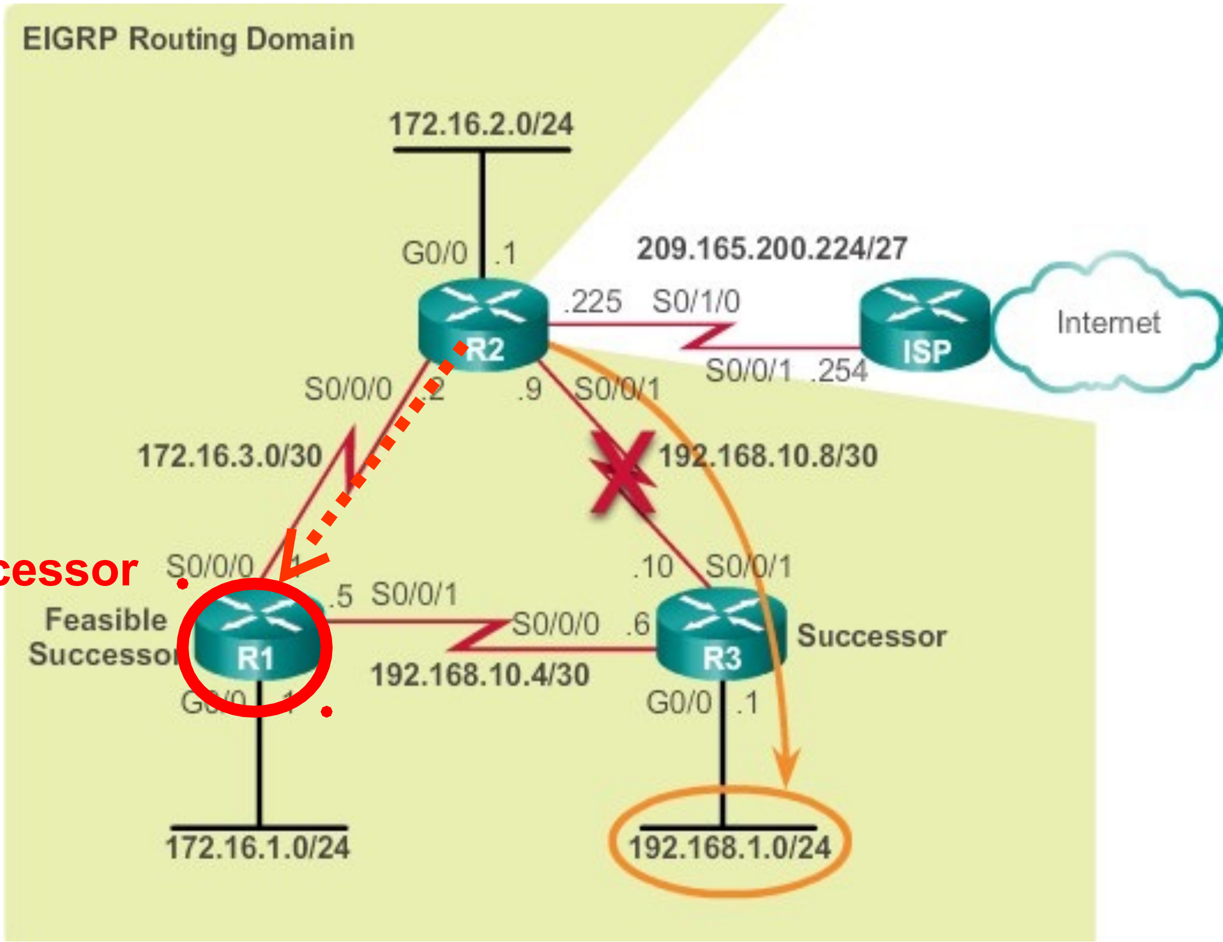
Successor

```
R2#debug eigrp fsm
EIGRP Finite State Machine debugging
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#interface s 0/0/1
R2(config-if)#shutdown
<Output omitted>
EIGRP-IPv4(1):Find FS for dest 192.168.1.0/24. FD is 3012096,
RD is 3012096 on tid 0
DUAL: AS(1) Removing dest 172.16.1.0/24, nexthop
192.168.10.10
DUAL: AS(1) RT installed 172.16.1.0/24 via 172.16.3.1
<Output omitted>
R2(config-if)#end
R2#undebug all
```



```
R2# show ip route

D      192.168.1.0/24 [90/41024256] via 172.16.3.1, 00:15:51, Serial0/0/0
```

New Successor (R1)



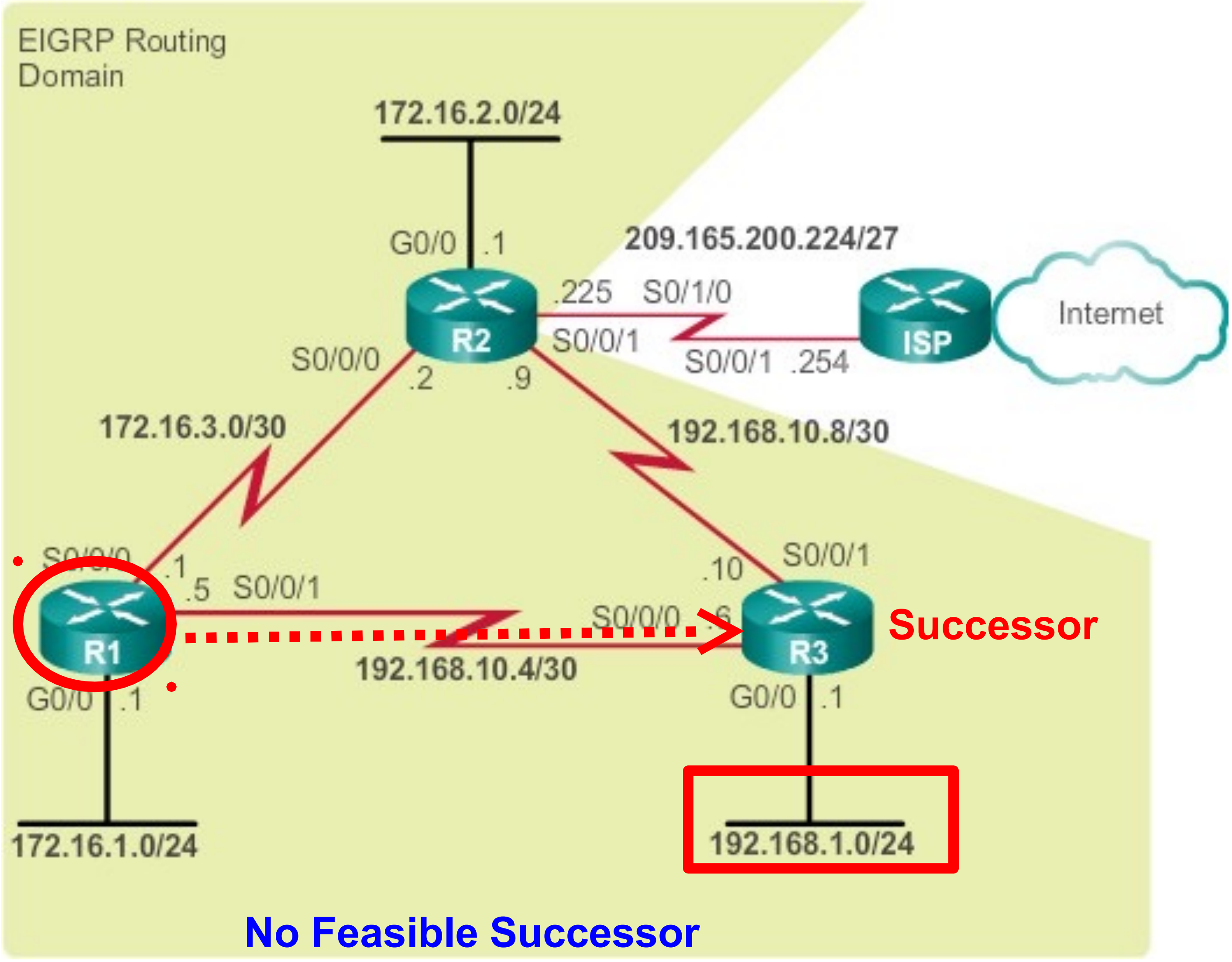
New Successor

```
R2# show ip eigrp topology

P 192.168.1.0/24, 1 successors, FD is 41024256
        via 172.16.3.1 (41024256/2170112), Serial0/0/0
```

Successor (R1)

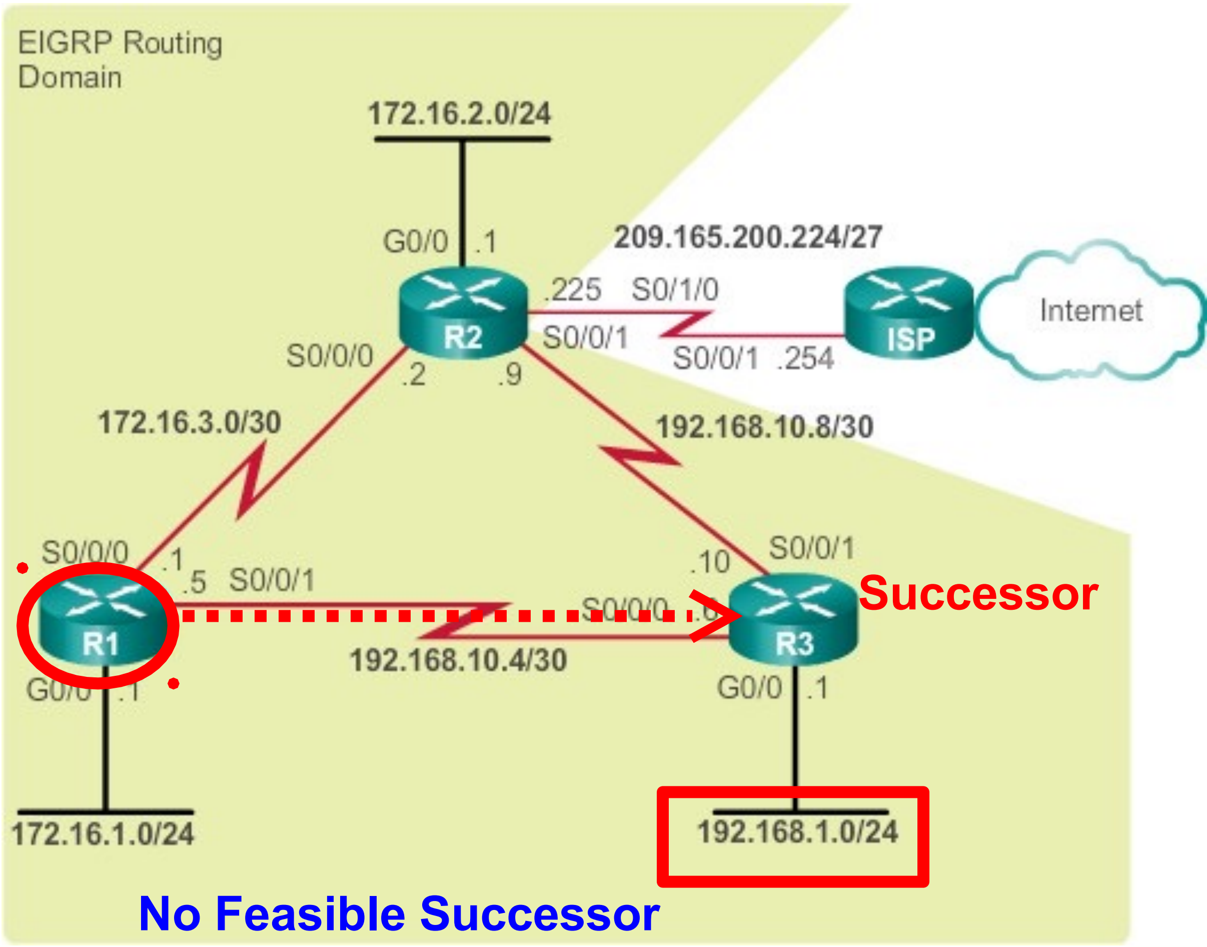
No feasible successor

When there is No Feasible Successor – DUAL needed



Successor

No Feasible Successor



Successor

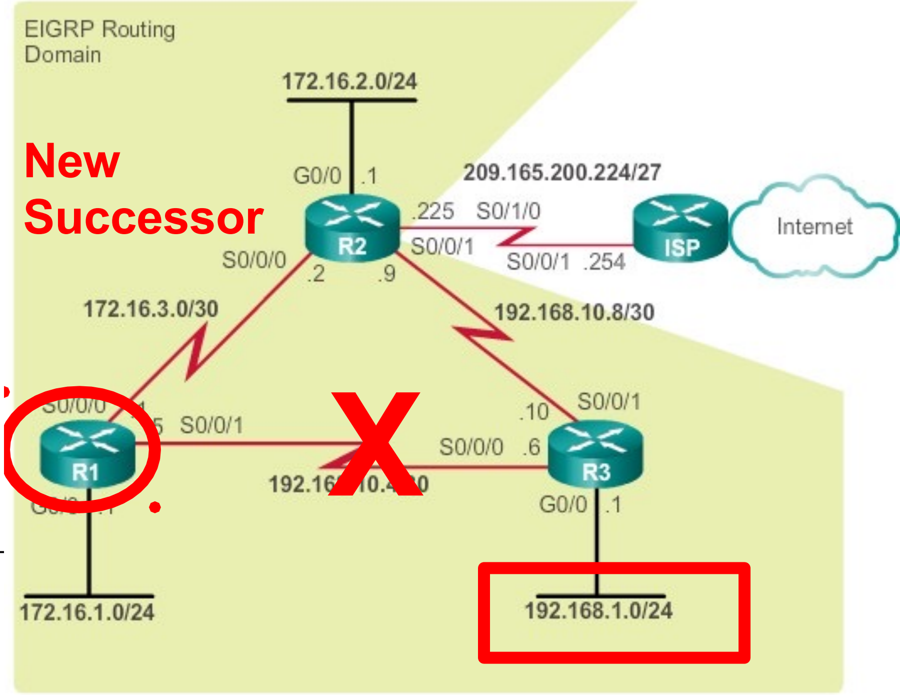
No Feasible Successor

```
R1# show ip eigrp topology

P 192.168.1.0/24, 1 successors, FD is 2170112
        via 192.168.10.6 (2170112/2816), Serial0/0/1
```

Successor (R3)

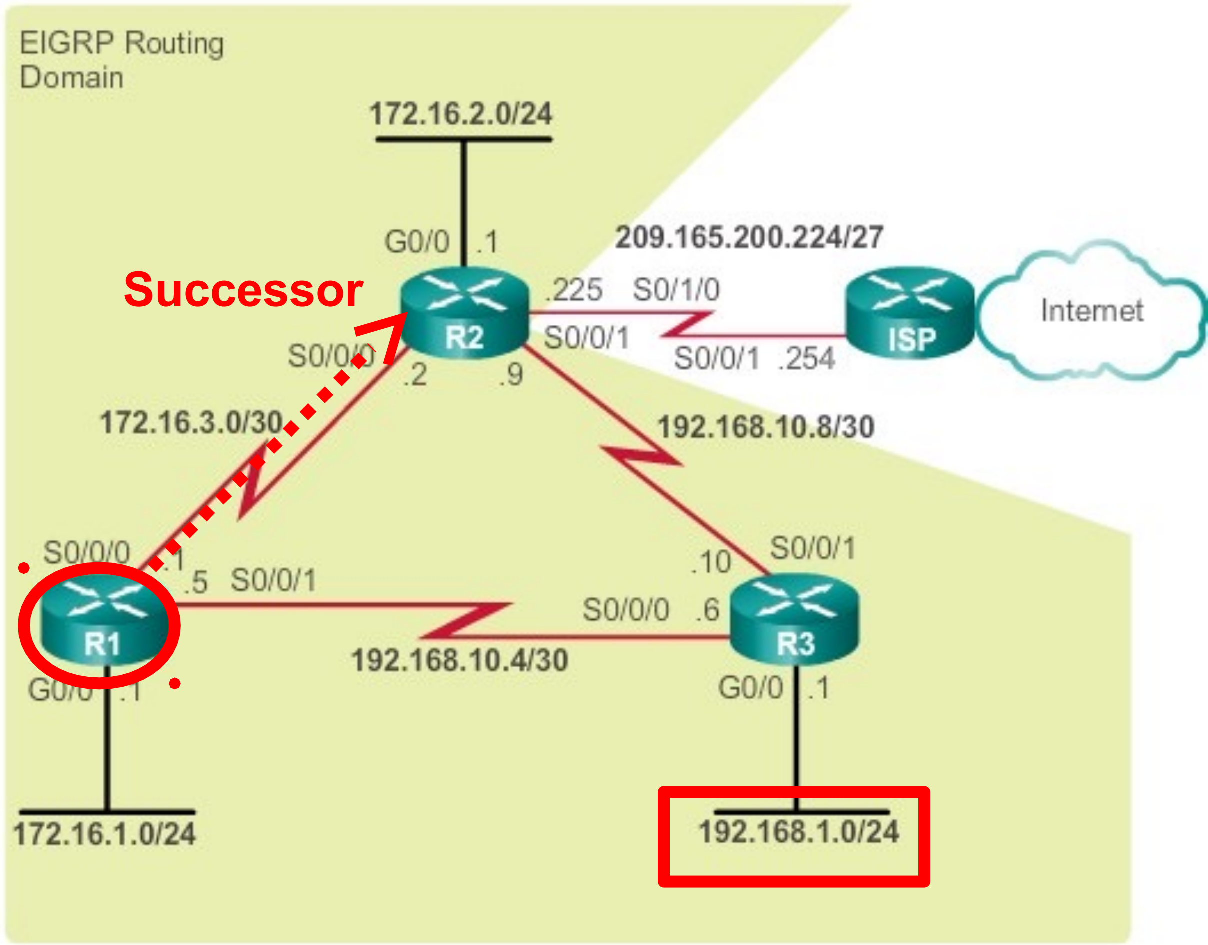
No feasible successor

- **No Feasible Successor so DUAL is initiated**
- **192.168.1.0/24 network** put into the **active state** and shows that EIGRP **queries** are sent to other neighbors.
- R2 **replies** with a path to this network
- This becomes the new successor
- Installed into the routing table.



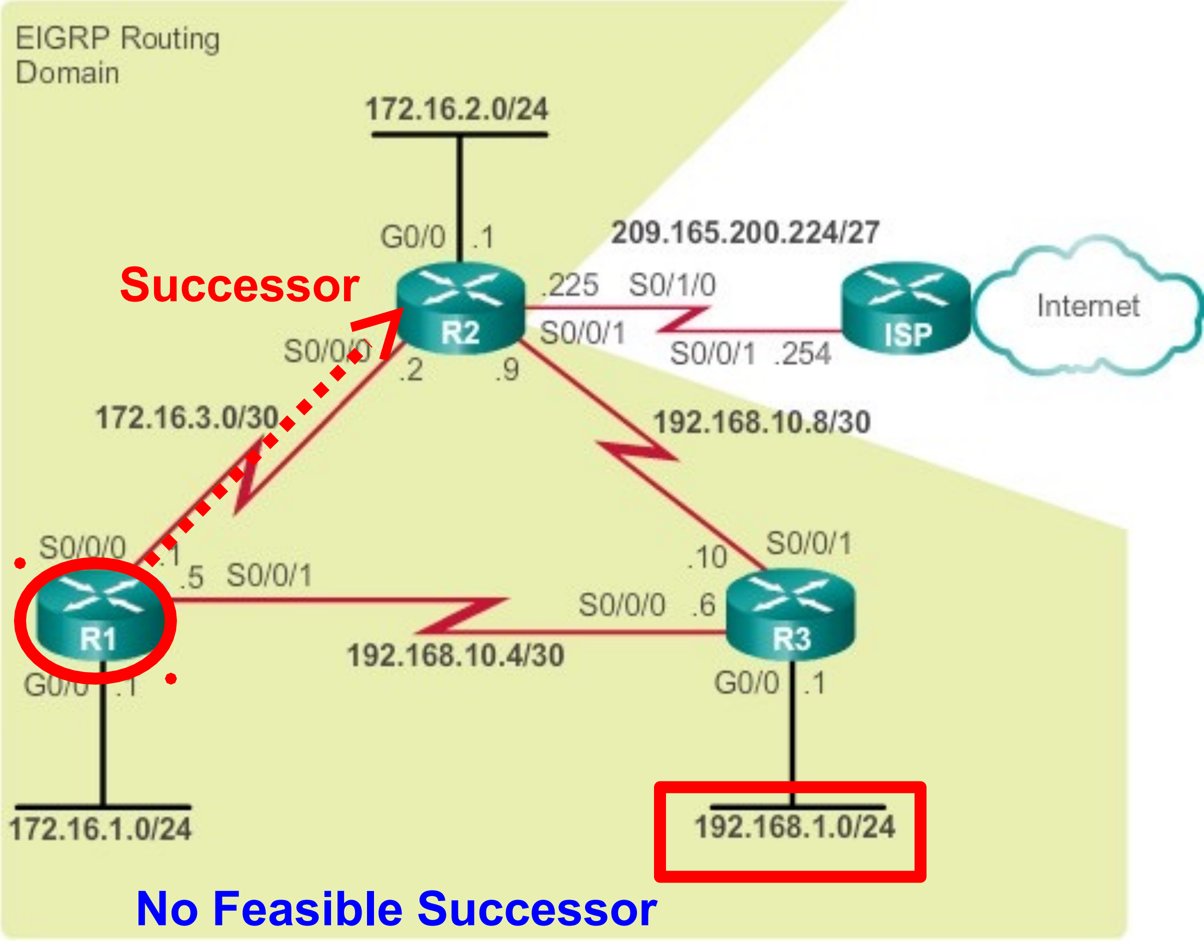
New Successor

```
R1#debug eigrp fsm
EIGRP Finite State Machine debugging is on
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface s 0/0/1
R1(config-if)#shutdown
<Output omitted>
EIGRP-IPv4(1): Find FS for dest 192.168.1.0/24. FD is 2170112, RD is 2170112
DUAL: AS(1) Dest 192.168.1.0/24 entering active state for tid 0.
EIGRP-IPv4(1): dest(192.168.1.0/24) active
EIGRP-IPv4(1): rcvreply: 192.168.1.0/24 via 172.16.3.2 metric 41024256/3012096
EIGRP-IPv4(1): reply count is 1
EIGRP-IPv4(1): Find FS for dest 192.168.1.0/24. FD is 72057594037927935, RD is
72057594037927935
DUAL: AS(1) Removing dest 192.168.1.0/24, nexthop 192.168.10.6
DUAL: AS(1) RT installed 192.168.1.0/24 via 172.16.3.2
<Output omitted>
R1(config-if)#end
R1#undebug all
```



```
R1# show ip route

D      192.168.1.0/24 [90/41024256] via 172.16.3.2, 00:05:25, Serial0/0/0
```

New Successor (R2)



Successor

No Feasible Successor

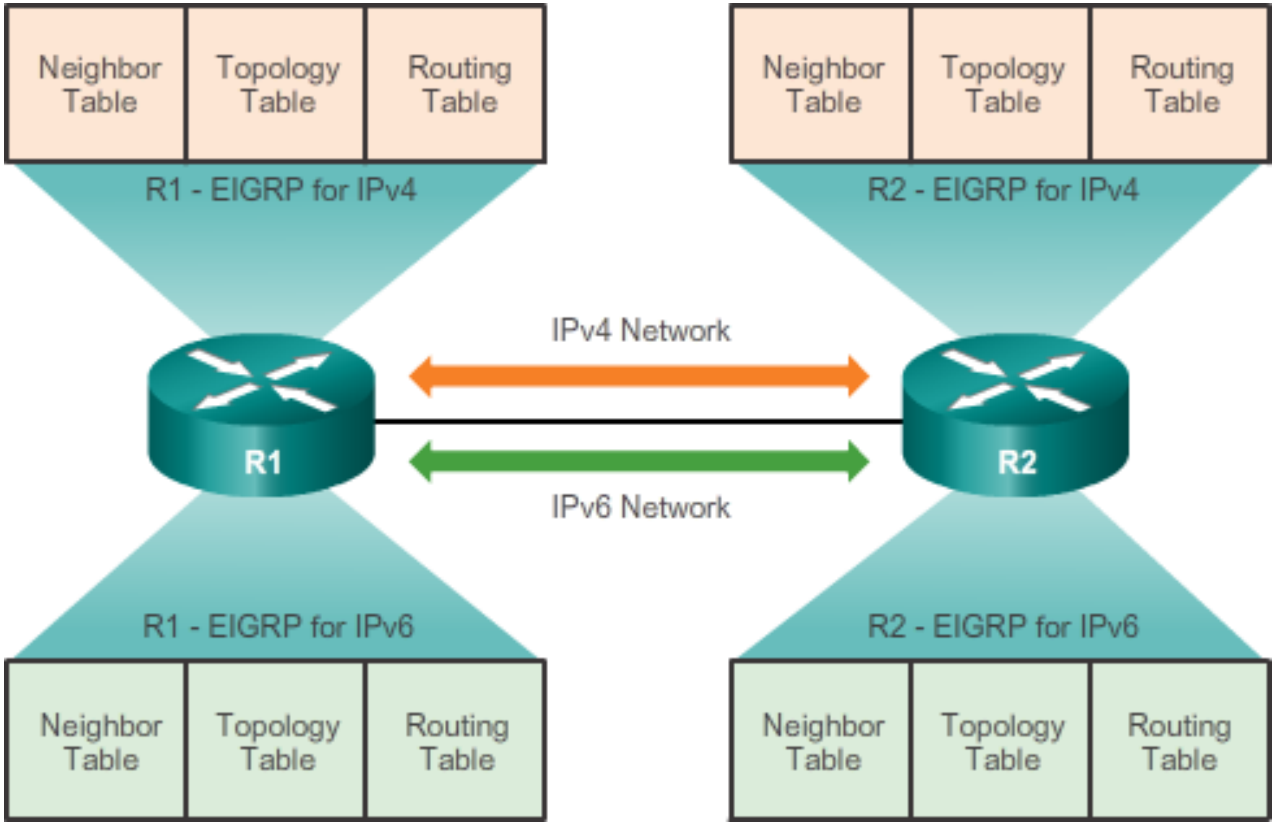
```
R1# show ip eigrp topology

P 192.168.1.0/24, 1 successors, FD is 41024256
        via 172.16.3.2 (41024256/3012096), Serial0/0/0
```

Successor (R2)

No feasible successor

EIGRP for IPv6

EIGRP for IPv6

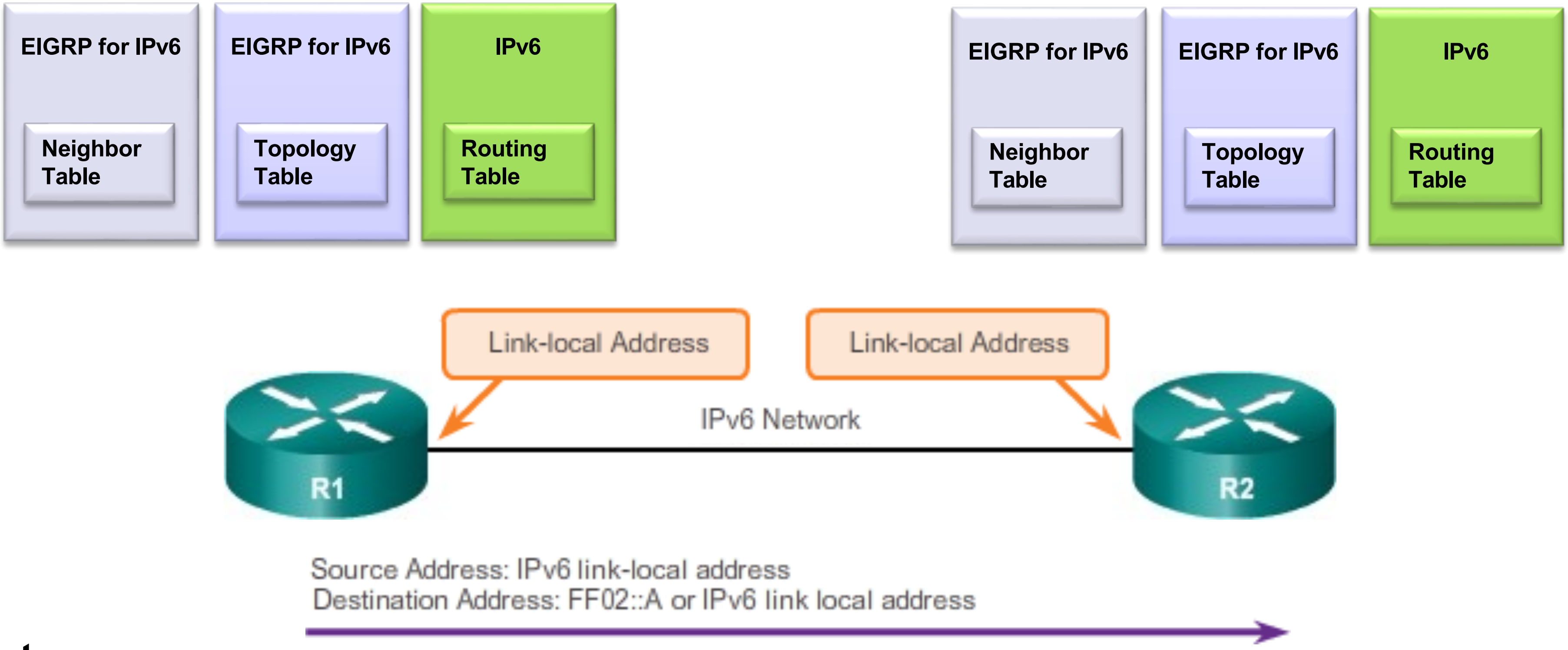


- EIGRP for IPv6 is a distance-vector routing protocol.
  - The configuration and operation is similar to EIGRP for IPv4.
- The following remained the same as EIGRP for IPv4:
  - Uses the same protocol number (88)
  - Maintains a topology table and queries if no feasible successors are available.
  - Uses DUAL to calculate the successor routes

EIGRP for IPv4 and EIGRP for IPv6

	EIGRP for IPv4	EIGRP for IPv6
Advertised routes		
Distance vector		
Convergence technology		
Metric		
Transport protocol		
Update messages		
Neighbor discovery		
Source address; destination addresses		
Authentication		
Router ID		32-bit router ID

EIGRP for IPv4 and EIGRP for IPv6



Source Address: IPv6 link-local address
Destination Address: FF02::A or IPv6 link local address

Note:

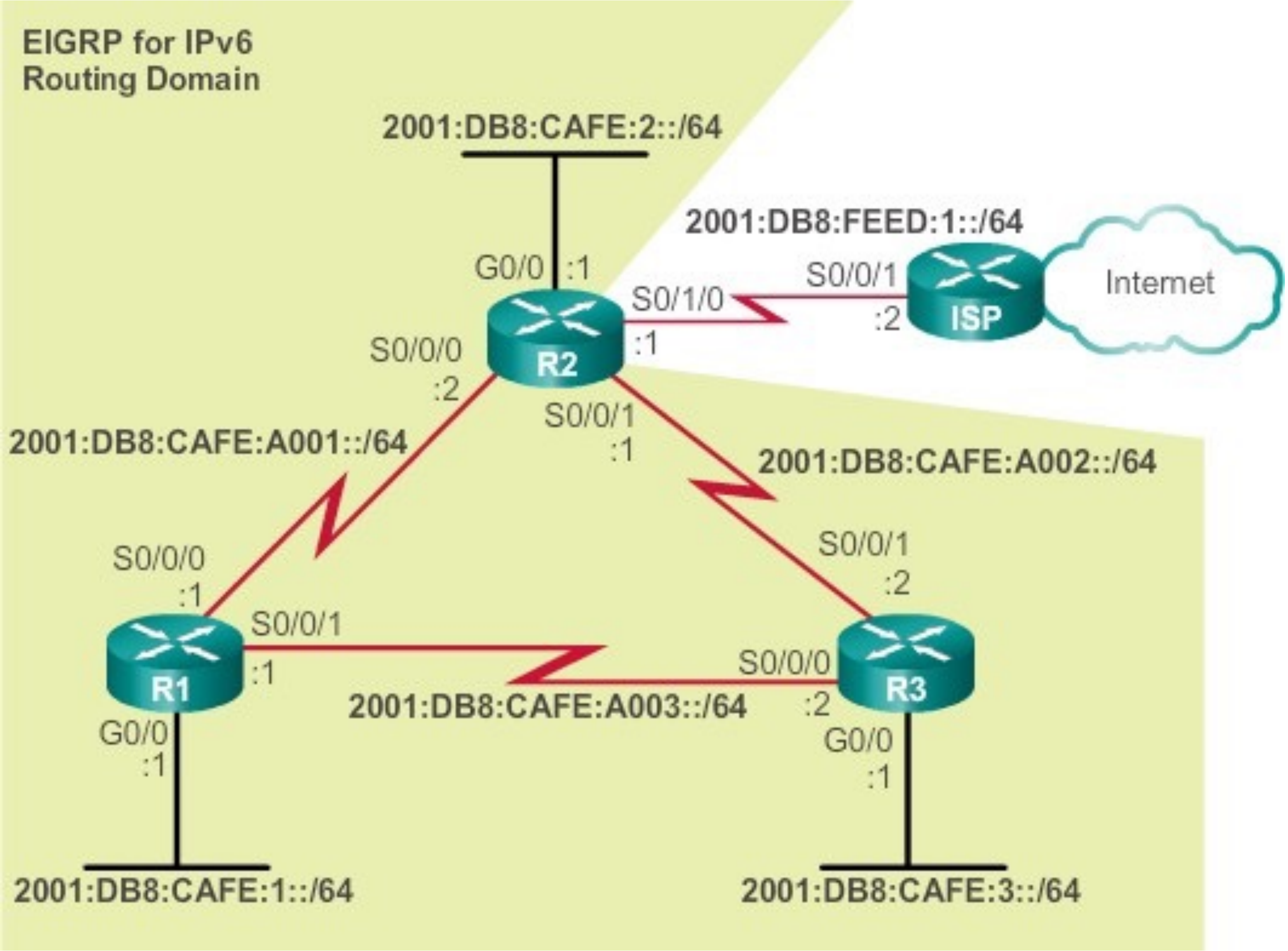
•IPv6 link-local addresses are in the FE80::/10 range.

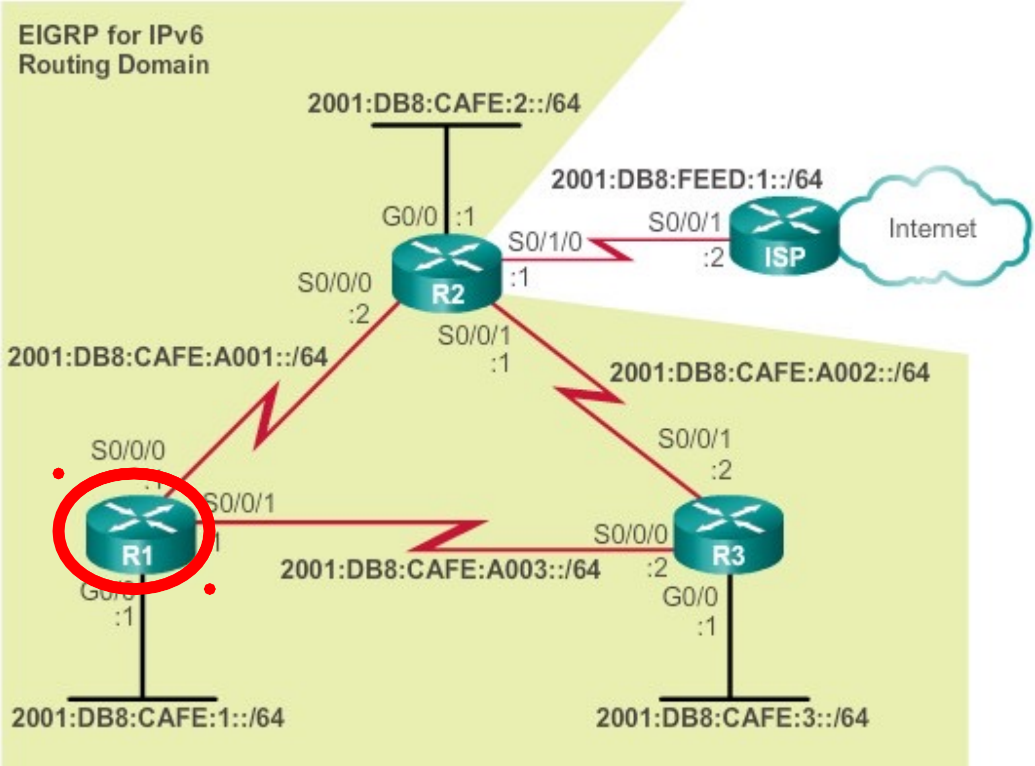
•The /10 indicates that the first 10 bits are 1111 1110 10xx xxxx, which results in the first hextet having a range of:

• 1111 1110 1000 0000 (FE80) to 1111 1110 1011 1111 (FEBF).

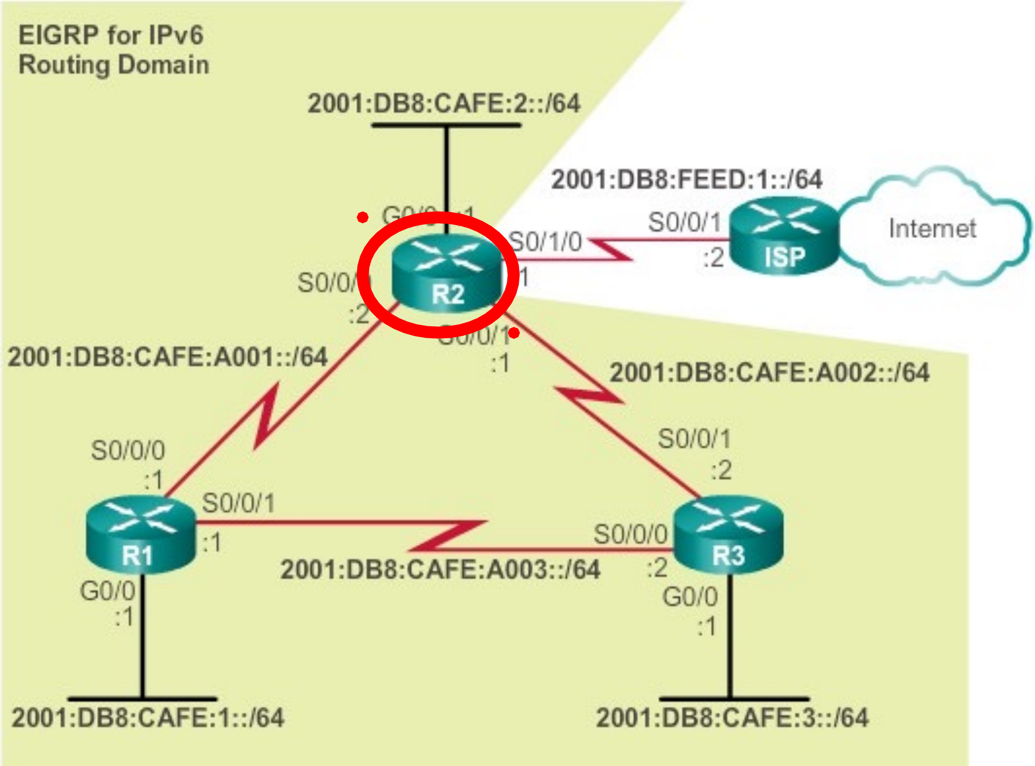
F E 8 0 F E B F

EIGRP for IPv6 Topology

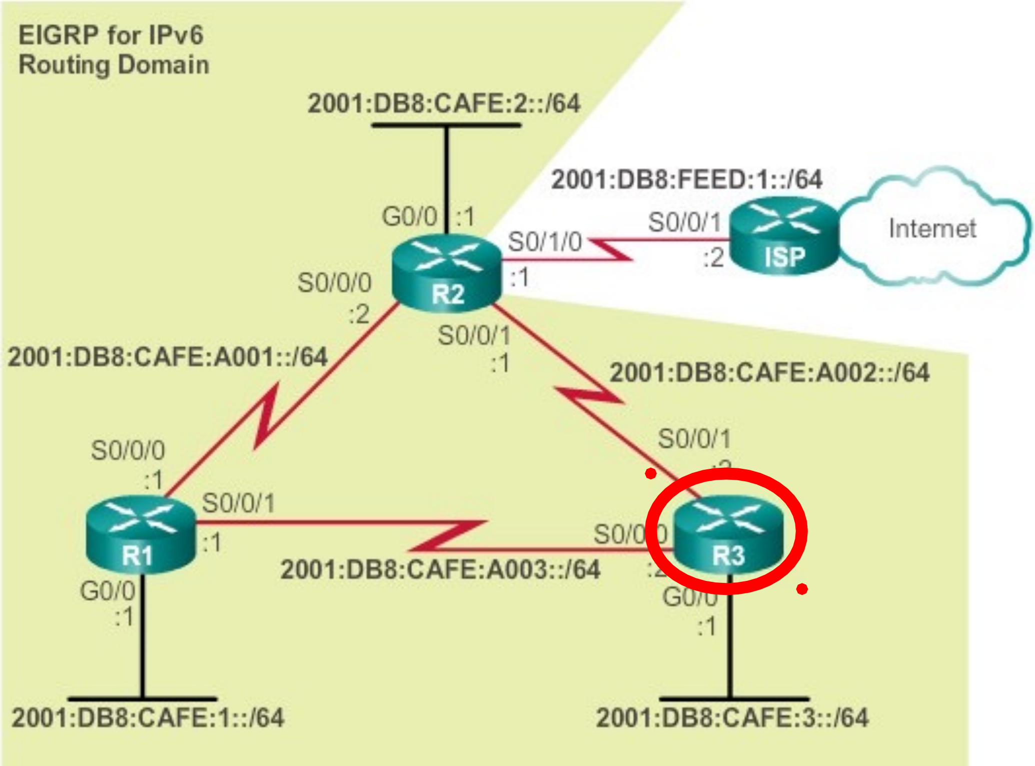




```
R1#show running-config
!
interface GigabitEthernet0/0
 ipv6 address 2001:DB8:CAFE:1::1/64
!
interface Serial0/0/0
 ipv6 address 2001:DB8:CAFE:A001::1/64
 clock rate 64000
!
interface Serial0/0/1
 ipv6 address 2001:DB8:CAFE:A003::1/64
```

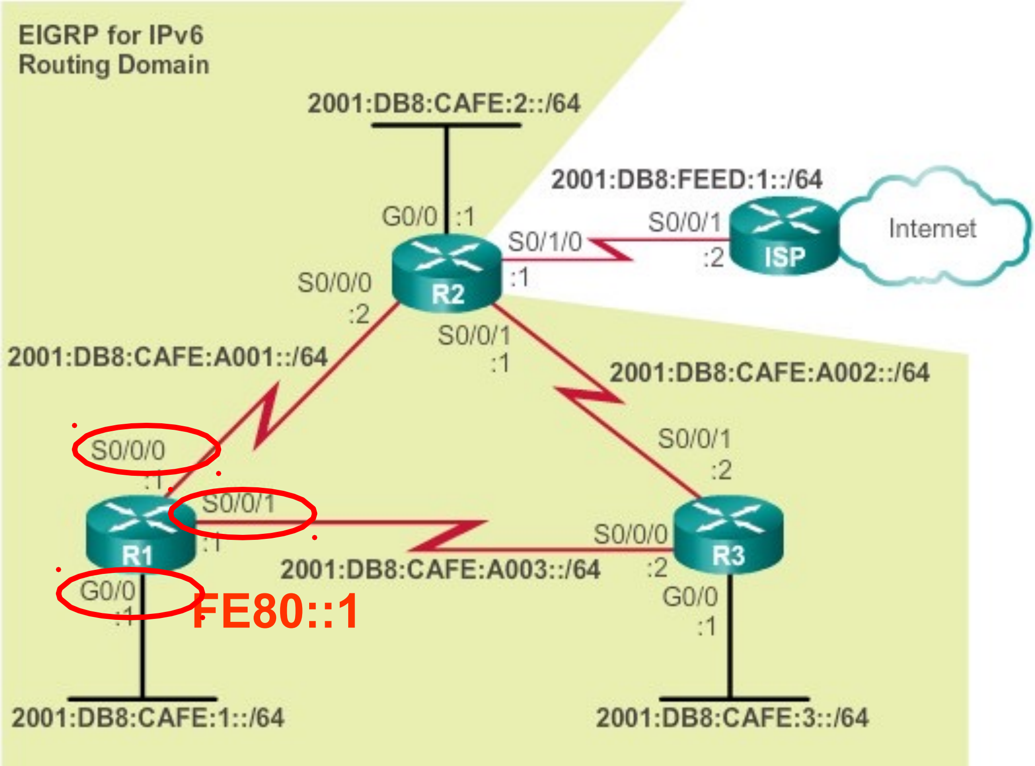


```
R2#show running-config
!
interface GigabitEthernet0/0
 ipv6 address 2001:DB8:CAFÉ:2::1/64
!
interface Serial0/0/0
 ipv6 address 2001:DB8:CAFE:A001::2/64
!
interface Serial0/0/1
 ipv6 address 2001:DB8:CAFE:A002::1/64
 clock rate 64000
!
interface Serial0/1/0
 ipv6 address 2001:DB8:FEED:1::1/64
```

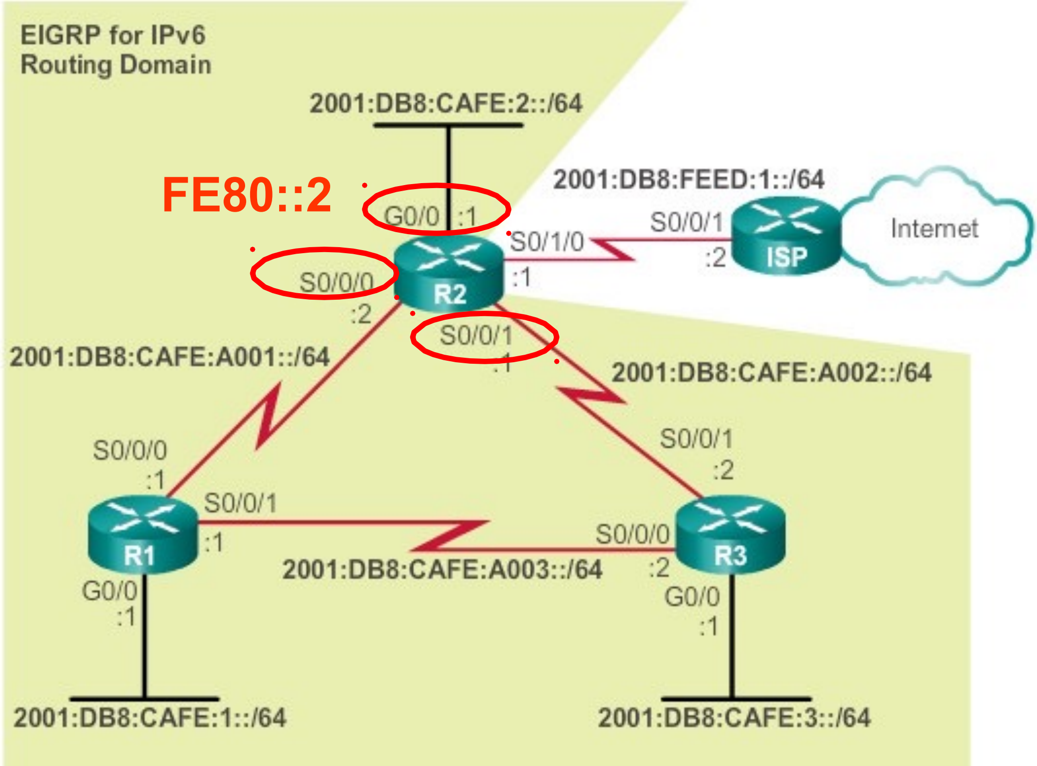


```
R3#show running-config
!
interface GigabitEthernet0/0
 ipv6 address 2001:DB8:CAFE:3::1/64
!
interface Serial0/0/0
 ipv6 address 2001:DB8:CAFE:A003::2/64
 clock rate 64000
!
interface Serial0/0/1
 ipv6 address 2001:DB8:CAFE:A002::2/64
```

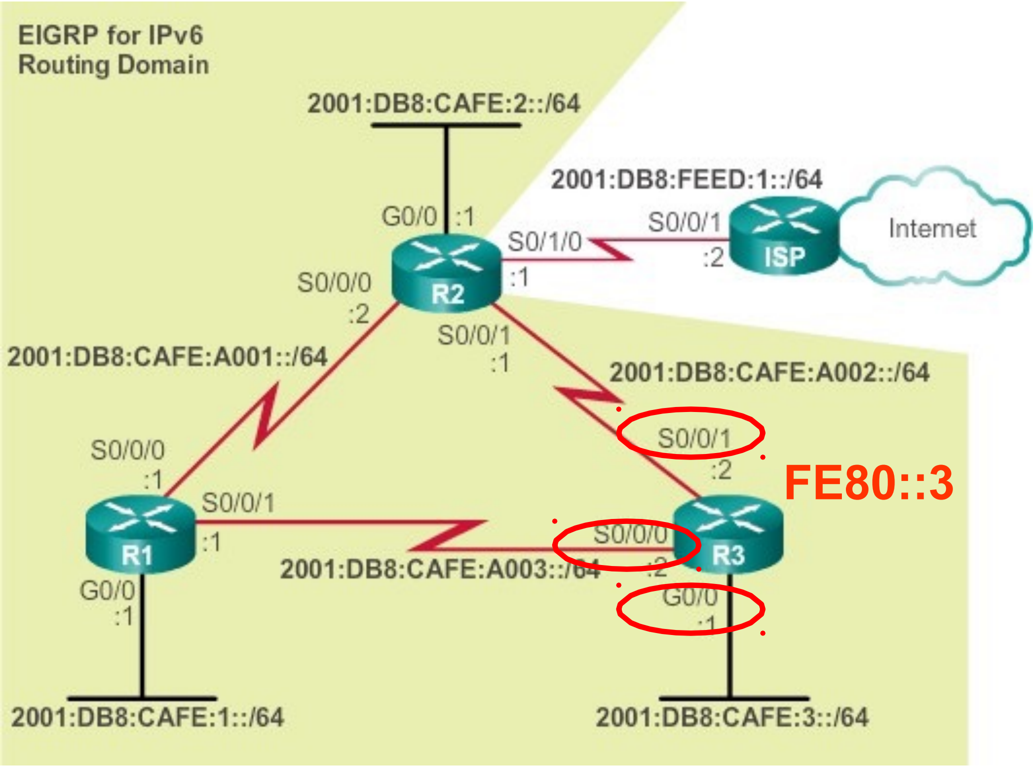
- IPv6 routing protocols use link-local addresses to exchange routing messages.
- By default, Cisco routers use EUI-64 to automatically create a link-local address.
- Static link-local addresses make it easier to remember and identify the router.
- Link-local addresses only need to be unique on the link.



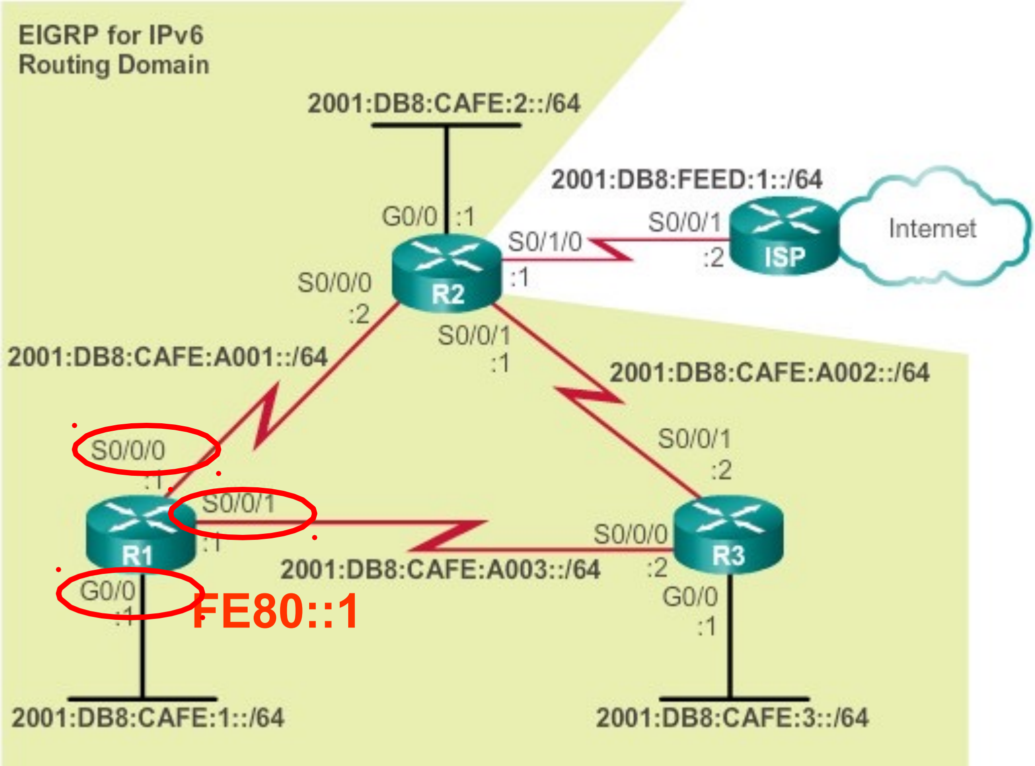
```
R1(config)# interface s 0/0/0
R1(config-if)# ipv6 address fe80::1 ?
  link-local  Use link-local address

R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if)# exit
R1(config)# interface s 0/0/1
R1(config-if)# ipv6 address fe80::1 link-local
R1(config-if) #exit
R1(config)# interface g 0/0
R1(config-if)# ipv6 address fe80::1 link-local
```



```
R2(config)# interface s 0/0/0
R2(config-if)# ipv6 address fe80::2 link-local
R2(config-if)# exit
R2(config)# interface s 0/0/1
R2(config-if)# ipv6 address fe80::2 link-local
R2(config-if)# exit
R2(config)# interface s 0/1/0
R2(config-if)# ipv6 address fe80::2 link-local
R2(config-if)# exit
R2(config)# interface g 0/0
R2(config-if)# ipv6 address fe80::2 link-local
```



```
R3(config)# interface serial 0/0/0
R3(config-if)# ipv6 address fe80::3 link-local
R3(config-if)# exit
R3(config)# interface serial 0/0/1
R3(config-if)# ipv6 address fe80::3 link-local
R3(config-if)# exit
R3(config)# interface gigabitethernet 0/0
R3(config-if)# ipv6 address fe80::3 link-local
R3(config-if)#
```

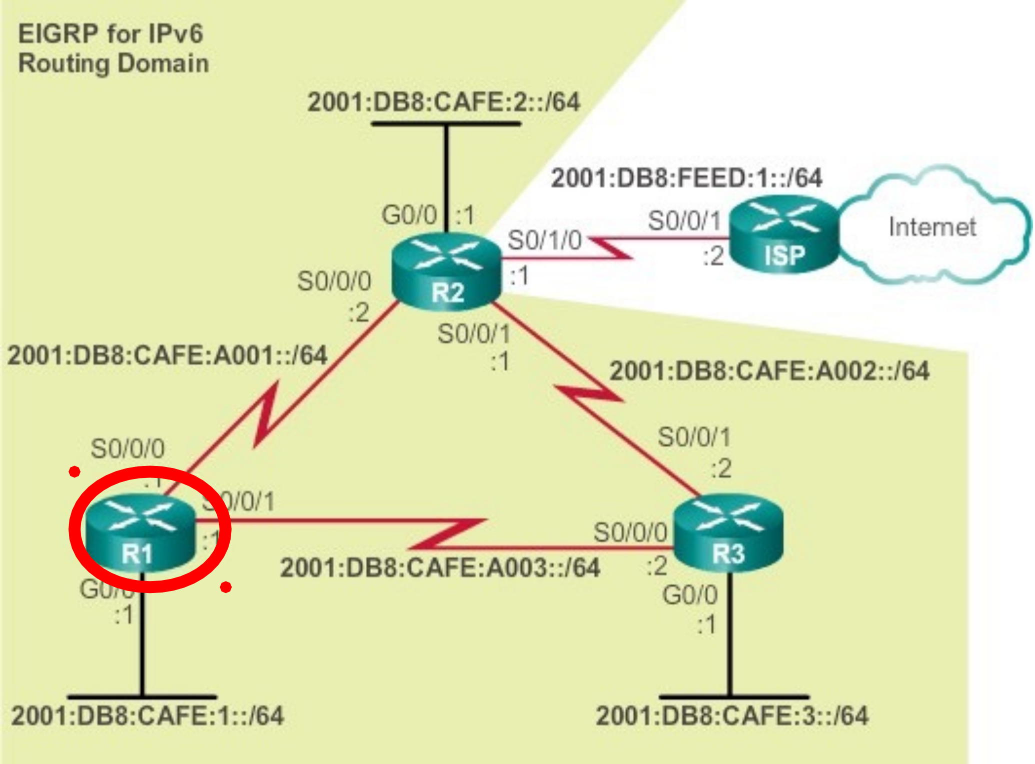


- Static link-local addresses make it easier to remember and identify the router.
- Link-local addresses only need to be unique on the link.

```
R1#show ipv6 interface brief
GigabitEthernet0/0          [up/up]
    FE80::1
    2001:DB8:CAFE:1::1
Serial0/0/0                 [up/up]
    FE80::1
    2001:DB8:CAFE:A001::1
Serial0/0/1                 [up/up]
    FE80::1
    2001:DB8:CAFE:A003::1
R1#
```

Same IPv6 link-local address is configured on all interfaces.

Enabling IPv6 Routing



```
R1(config)# ipv6 router eigrp 2
% IPv6 routing not enabled
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 router eigrp 2
R1(config-rtr)#
```

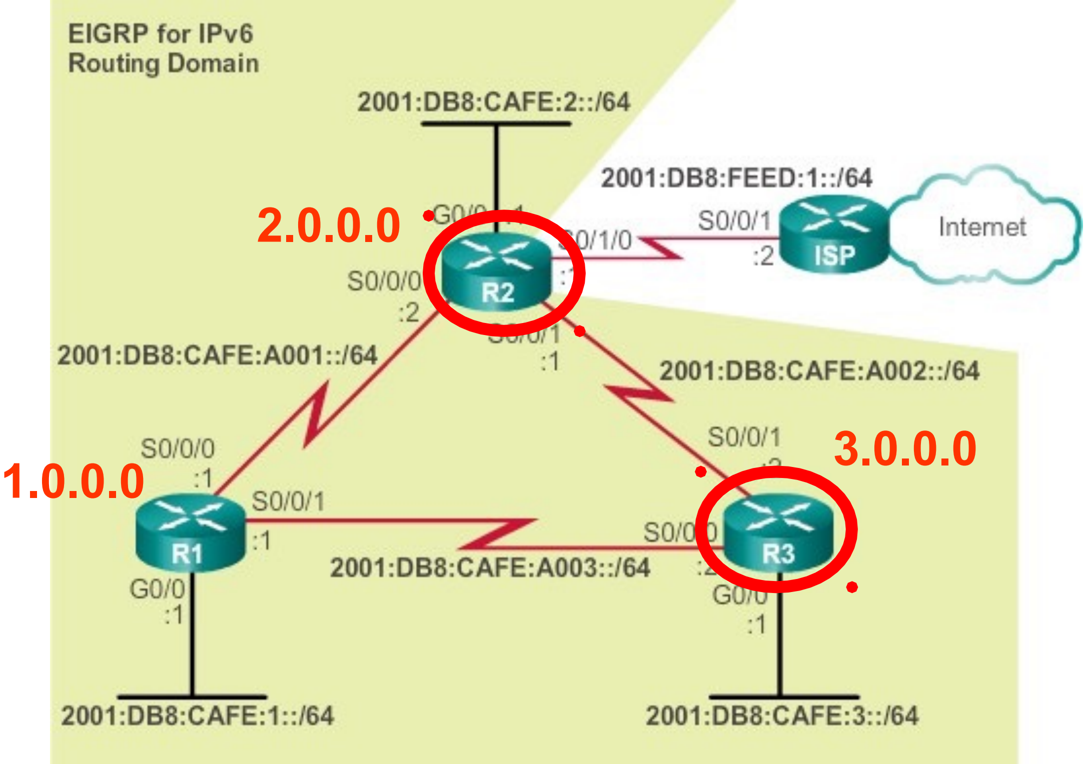
- **`ip unicast-routing`** command is required for forwarding IPv6 packets, static IPv6 routes and dynamic IPv6 routing protocols.
- The EIGRP AS “2” must be the same on all routers.

EIGRP Router ID and no shutdown

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 router eigrp 2
R1(config-rtr)# eigrp router-id 1.0.0.0
R1(config-rtr)# no shutdown
R1(config-rtr)#
```

- Criteria for deriving the router ID:
  1. **Configured router ID**:
     - Configured with **`eigrp router-id`** *`router-id`* command
     - Note: Some versions of IOS will accept **`router-id`**.
  2. **Highest Loopback IPv4 address**:
  3. **Highest active interface IPv4 address**:

- EIGRP (IPv4 and IPv6) requires a 32-bit router ID.
- If there is no 32-bit IPv4 address configured on the router, then a router-id command is required.
- This is used to uniquely identify the router in EIGRP messages.
- EIGRP for IPv6 also requires the **`no shutdown`** command

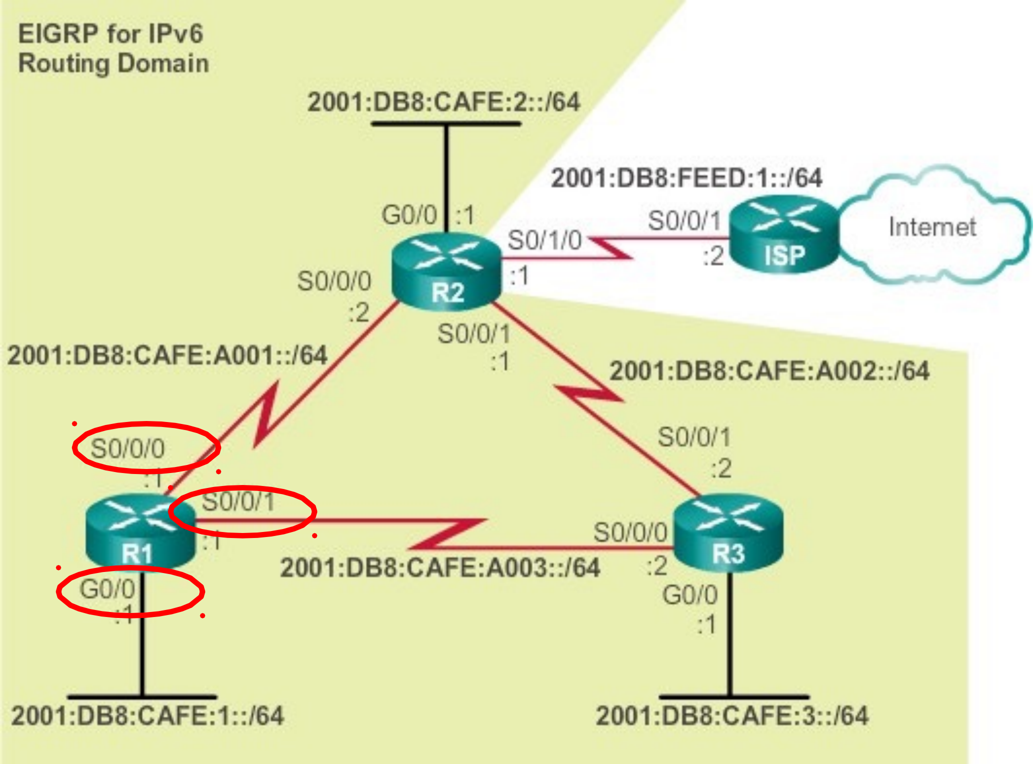


```
R2(config)# ipv6 unicast-routing
R2(config)# ipv6 router eigrp 2
R2(config-rtr)# eigrp router-id 2.0.0.0
R2(config-rtr)# no shutdown
R2(config-rtr)#
```

```
R3(config)# ipv6 unicast-routing
R3(config)# ipv6 router eigrp 2
R3(config-rtr)# eigrp router-id 3.0.0.0
R3(config-rtr)# no shutdown
R3(config-rtr)#
```

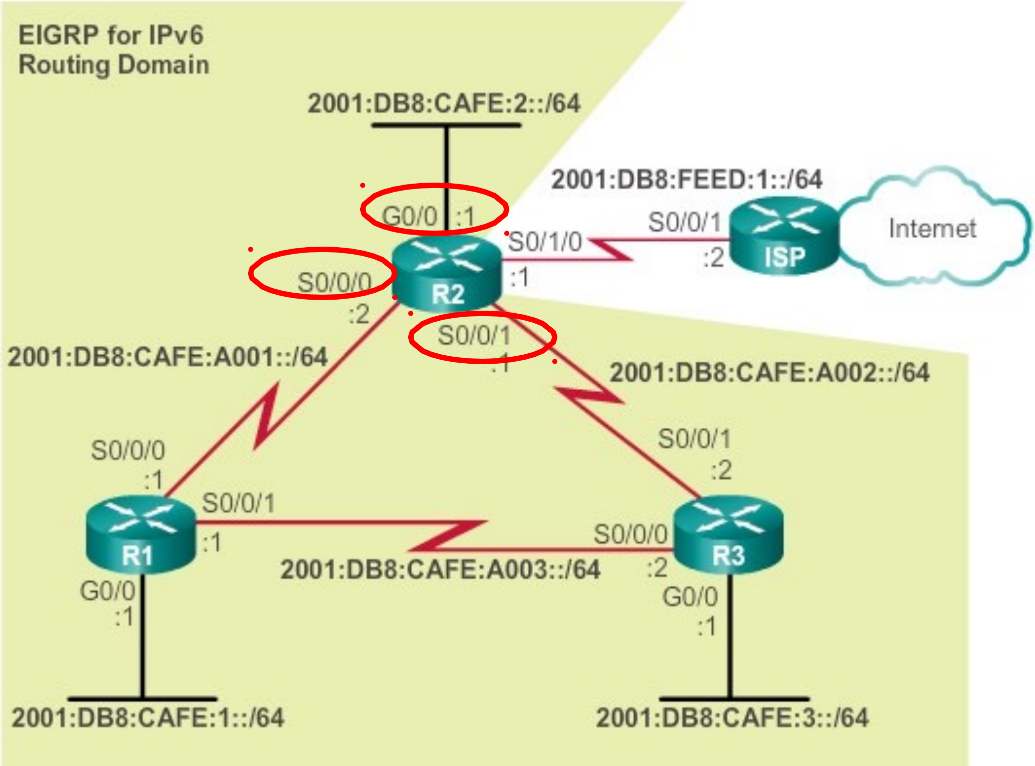
Enabling EIGRP for IPv6 on the Interface

- EIGRP for IPv6 is enabled on the interface with the **`ipv6 eigrp`** *AS* command.
- There is no **`network`** command



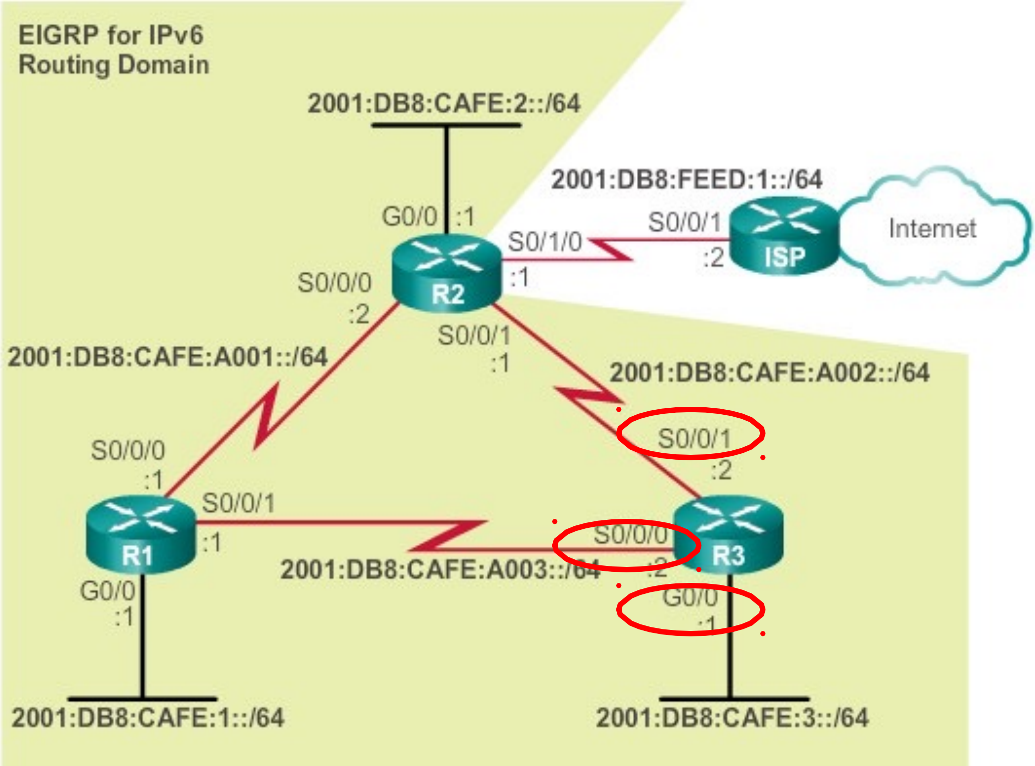
```
R1(config)# interface g0/0
R1(config-if)# ipv6 eigrp 2
R1(config-if)# exit
R1(config)# interface s 0/0/0
R1(config-if)#ipv6 eigrp 2
R1(config-if)# exit
R1(config)# interface s 0/0/1
R1(config-if)# ipv6 eigrp 2
R1(config-if)#
```

Enabling EIGRP for IPv6 on the Interface



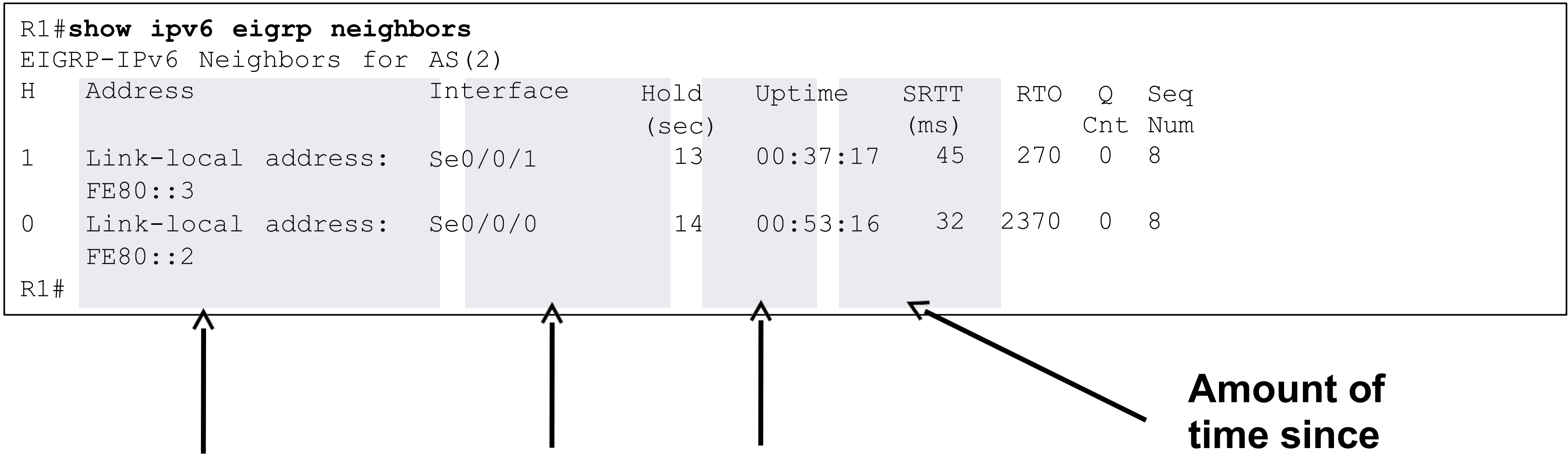
```
R2(config)# interface g 0/0
R2(config-if)# ipv6 eigrp 2
R2(config-if)# exit
R2(config)# interface s 0/0/0
R2(config-if)# ipv6 eigrp 2
R2(config-if)# exit
%DUAL-5-NBRCHANGE: EIGRP-IPv6 2: Neighbor FE80::1
(Serial0/0/0) is up: new adjacency
R2(config)# interface s 0/0/1
R2(config-if)# ipv6 eigrp 2
R2(config-if)#
```

Enabling EIGRP for IPv6 on the Interface



```
R3(config)# interface g 0/0
R3(config-if)# ipv6 eigrp 2
R3(config-if) #exit
R3(config)# interface s 0/0/0
R3(config-if)# ipv6 eigrp 2
R3(config-if)#
*Mar  4 03:02:00.696: %DUAL-5-NBRCHANGE: EIGRP-IPv6 2:
Neighbor FE80::1 (Serial0/0/0) is up: new adjacency
R3(config-if)# exit
R3(config)# interface s 0/0/1
R3(config-if)# ipv6 eigrp 2
R3(config-if)#
*Mar  4 03:02:17.264: %DUAL-5-NBRCHANGE: EIGRP-IPv6 2:
Neighbor FE80::2 (Serial0/0/1) is up: new adjacency
```

Verifying EIGRP for IPv6: Examining Neighbors



```
R1#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(2)
H   Address                   Interface       Hold   Uptime     SRTT   RTO   Q   Seq
                                              (sec)             (ms)        Cnt  Num
1   Link-local address:       Se0/0/1          13    00:37:17    45    270   0   8
    FE80::3
0   Link-local address:       Se0/0/0          14    00:53:16    32   2370   0   8
    FE80::2
R1#
```

Neighbor's IPv6 Link-local Address

Local Interface receiving EIGRP for IPv6 Hello packets

Seconds remaining before declaring neighbor down.

The current hold time and is reset to the maximum hold time whenever a Hello packet is received.

Amount of time since this neighbor was added to the neighbor table.

Verifying EIGRP for IPv6: show ipv6 protocols

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 2"
EIGRP-IPv6 Protocol for AS(2)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 1.0.0.0
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 16
    Maximum hopcount 100
    Maximum metric variance 1

  Interfaces:
    GigabitEthernet0/0
    Serial0/0/0
    Serial0/0/1
  Redistribution:
    None
R1#
```

1 **Routing protocol and Process ID (AS Number)**

2 **K values used in composite metric**

3 **EIGRP Router ID**

4 **EIGRP Administrative Distances**

5 **Interfaces enabled for this EIGRP for IPv6.**

Verifying EIGRP for IPv6: Examining the IPv6 Routing Table

```
R1#show ipv6 route

C   2001:DB8:CAFE:1::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:DB8:CAFE:1::1/128 [0/0]
     via GigabitEthernet0/0, receive
D   2001:DB8:CAFE:2::/64 [90/3524096]
     via FE80::3, Serial0/0/1
D   2001:DB8:CAFE:3::/64 [90/2170112]
     via FE80::3, Serial0/0/1
C   2001:DB8:CAFE:A001::/64 [0/0]
     via Serial0/0/0, directly connected
L   2001:DB8:CAFE:A001::1/128 [0/0]
     via Serial0/0/0, receive
D   2001:DB8:CAFE:A002::/64 [90/3523840]
     via FE80::3, Serial0/0/1
C   2001:DB8:CAFE:A003::/64 [0/0]
     via Serial0/0/1, directly connected
L   2001:DB8:CAFE:A003::1/128 [0/0]
     via Serial0/0/1, receive
L   FF00::/8 [0/0]
     via Null0, receive
R1#
```

Notice link-local addresses used as next-hop addresses.

Routing messages come from the link-local address of the neighboring routers