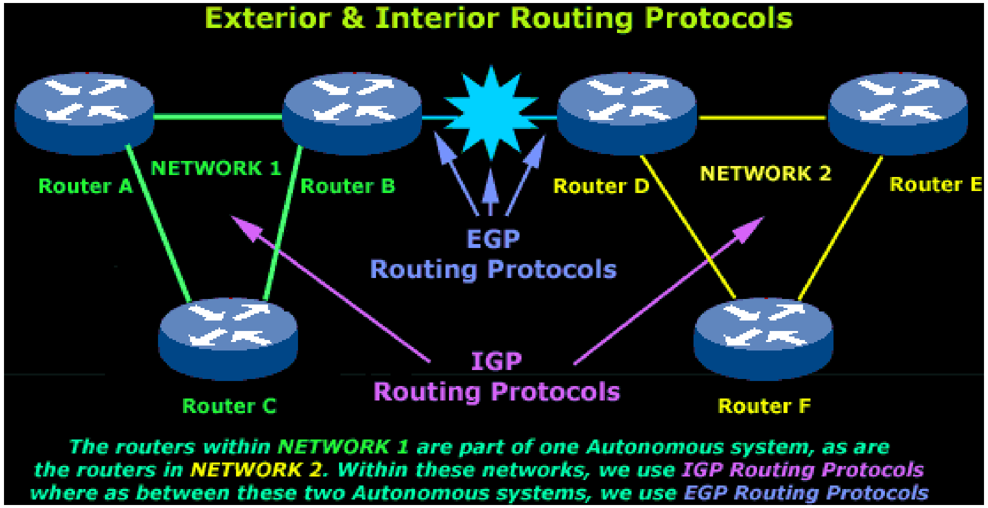
# CIS 3210
# **Routing Concepts**



# Dynamic Routing Protocols

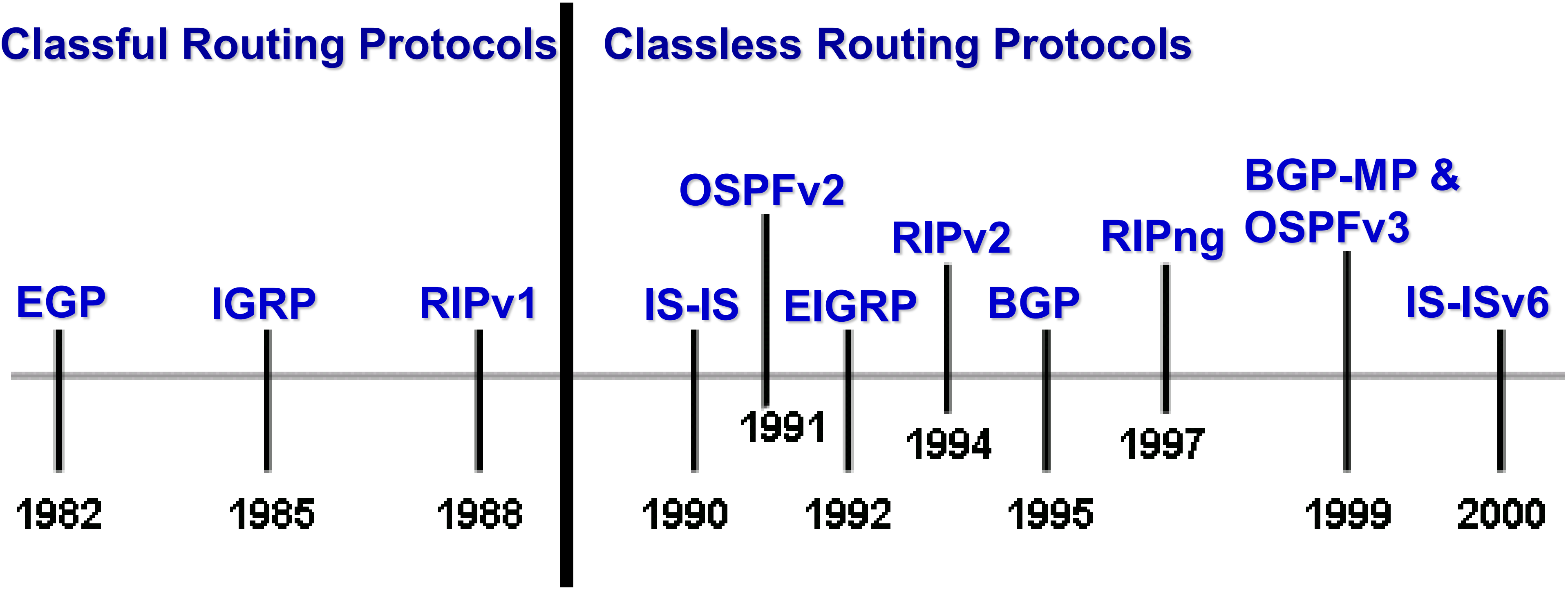
# Evolution of Dynamic Routing Protocols

## Routing Protocols Classification

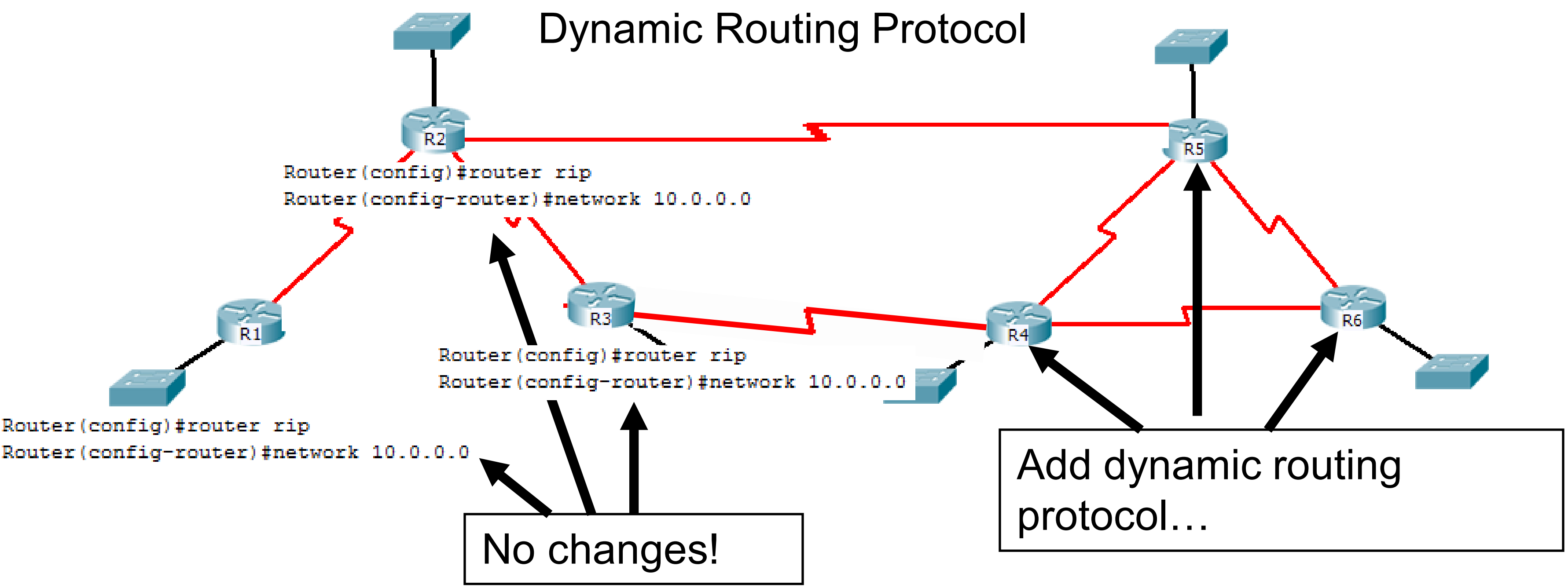
| | Interior Gateway Protocols | | | | Exterior Gateway Protocols |
|---|---|---|---|---|---|
| | Distance Vector | | Link-State | | Path Vector |
| IPv4 | RIPv2 | EIGRP | OSPFv2 | IS-IS | BGP-4 |
| IPv6 | RIPng | EIGRP for IPv6 | OSPFv3 | IS-IS for IPv6 | BGP-MP |

- Dynamic routing protocols used in networks since the late 1980s.
- Newer versions support the communication based on IPv6.

# Evolution of Dynamic Routing Protocols



# Purpose of Dynamic Routing Protocols




- Routing protocols are used to facilitate the exchange of routing information between routers.
- The purpose of dynamic routing protocols includes:
  - Discovery of remote networks
  - Maintaining up-to-date routing information
  - Choosing the best path to destination networks
  - Ability to find a new best path if the current path is no longer available

# Main Components of Routing Protocols

**Data Structures**

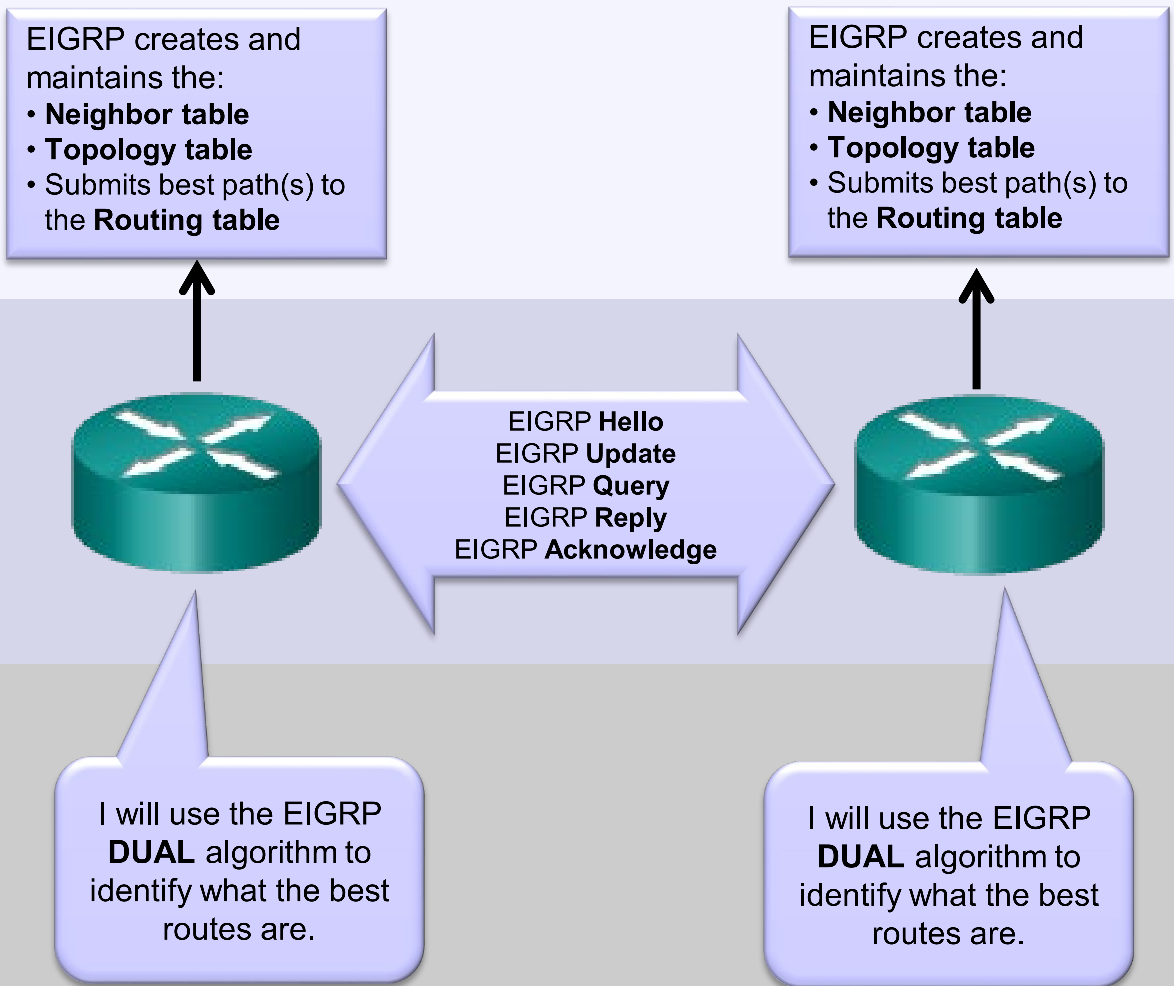
- Routing protocols create and maintain tables (databases) in RAM for its operations.

**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**

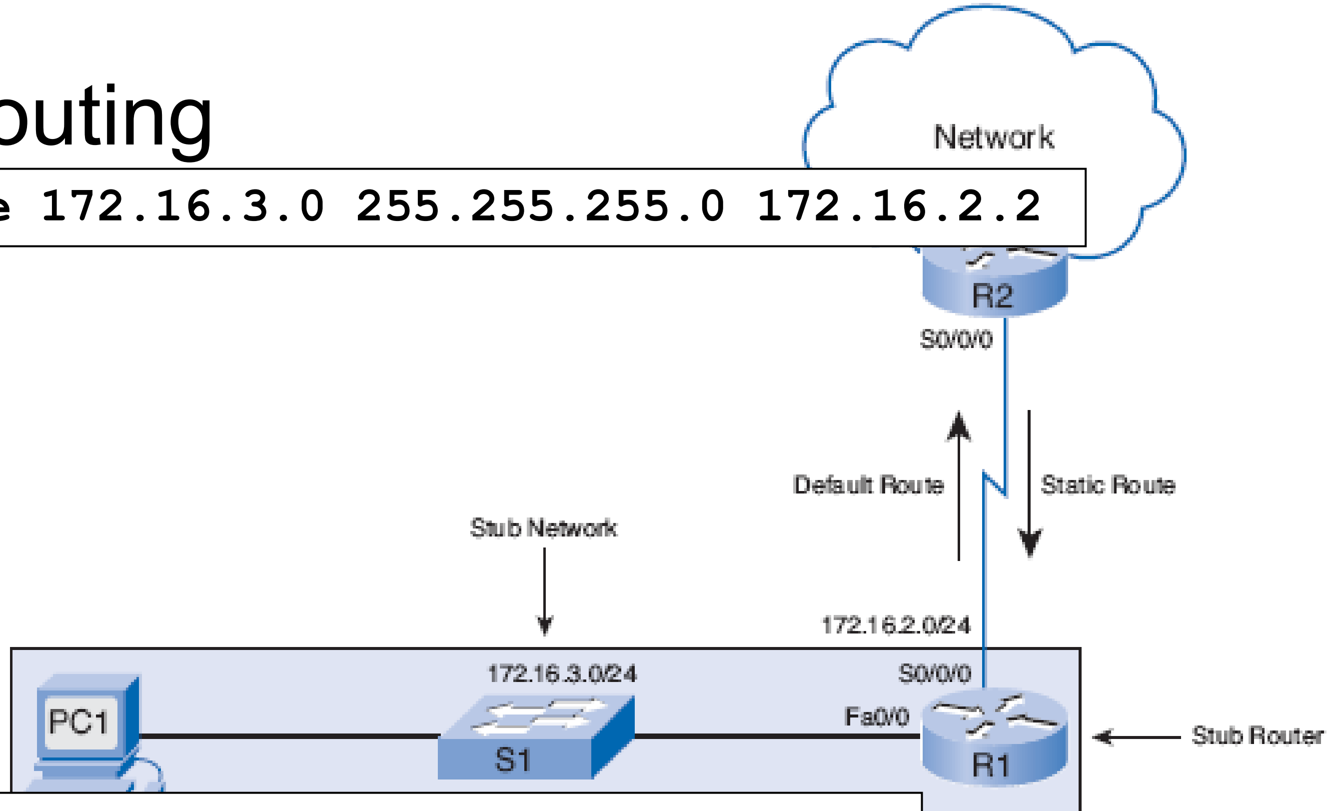
- Routing protocols use algorithms to determine the best path to various destinations.



# Dynamic versus Static Routing

# Using Static Routing

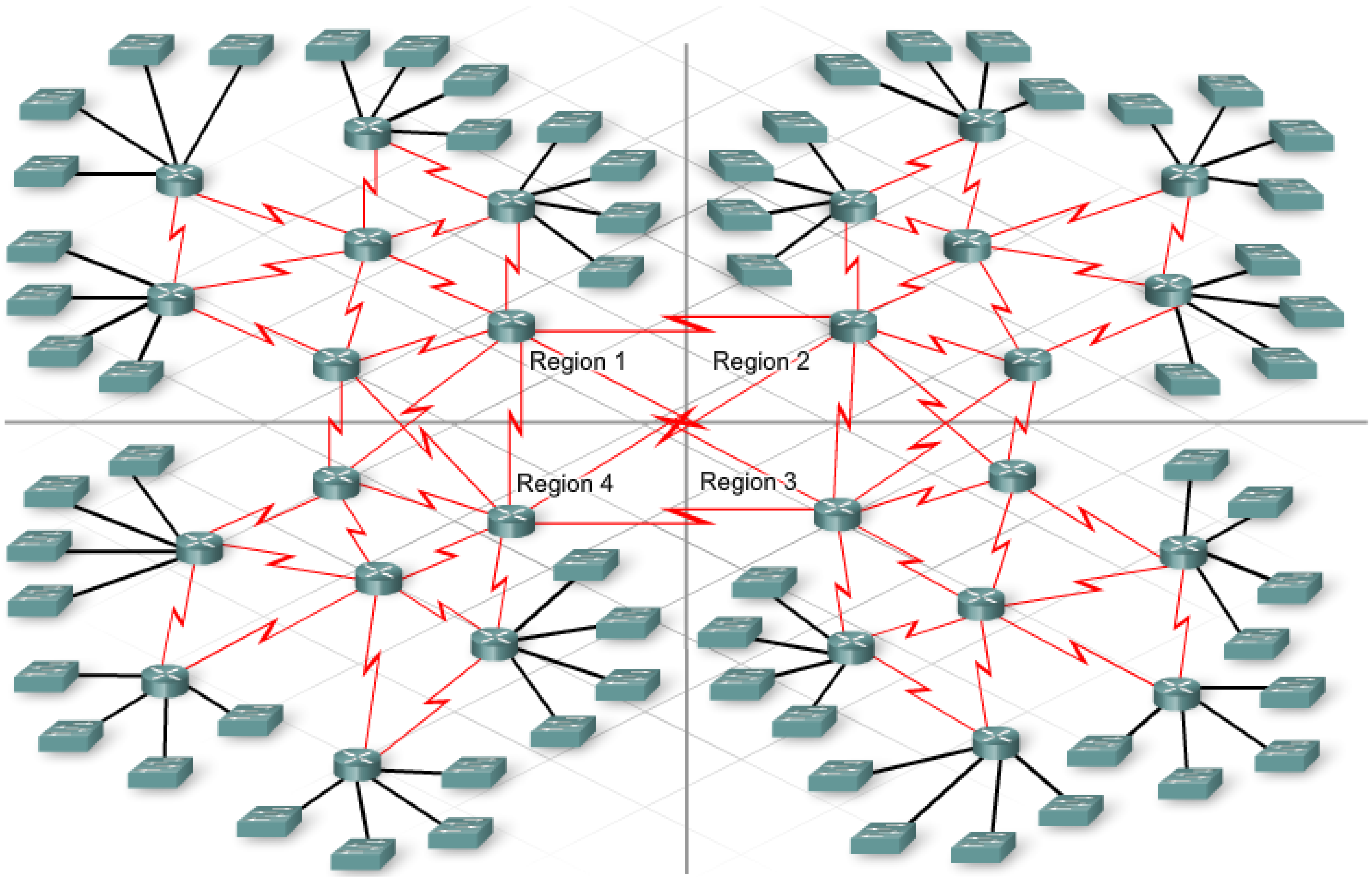
```
R2(config)# ip route 172.16.3.0 255.255.255.0 172.16.2.2
```




```
R1(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.1
```

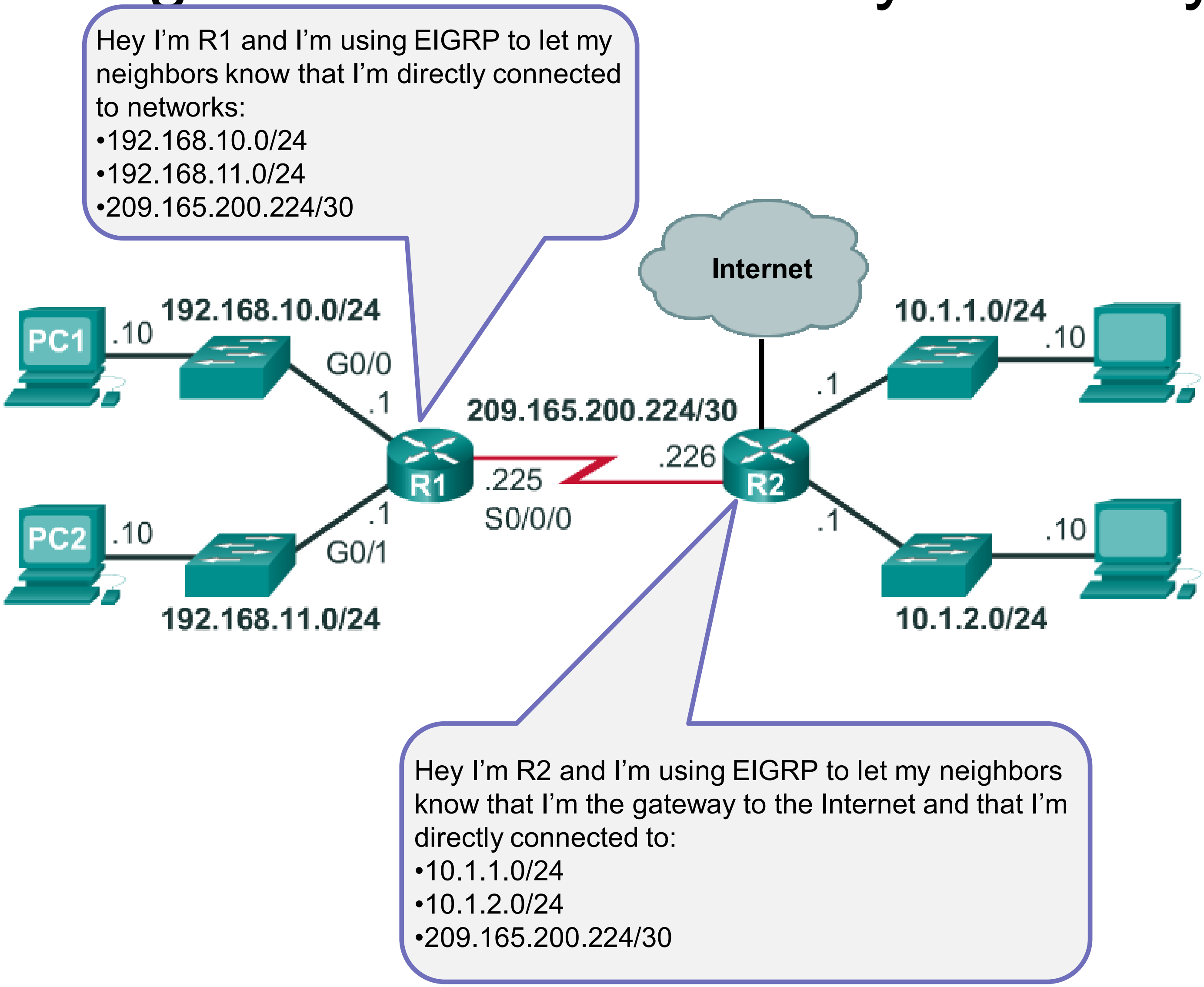
- Static routing has several primary uses:
  - Best for smaller networks that are not expected to grow significantly.
  - Routing to and from a stub network.
  - A default route.
  - Networks typically use a combination of both static and dynamic routing.

# Dynamic Routing Scenario



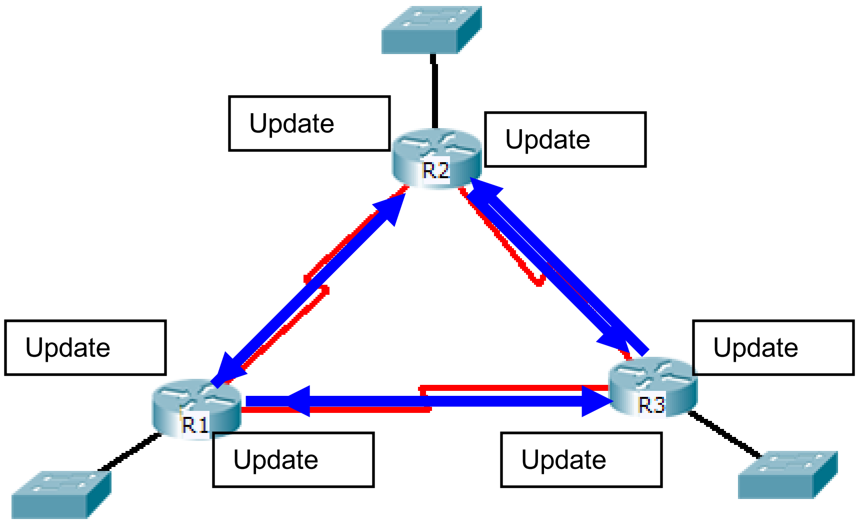

Dynamic routing is the best choice for large networks.

# Reaching Remote Networks Dynamically



# Routing Protocol Operation

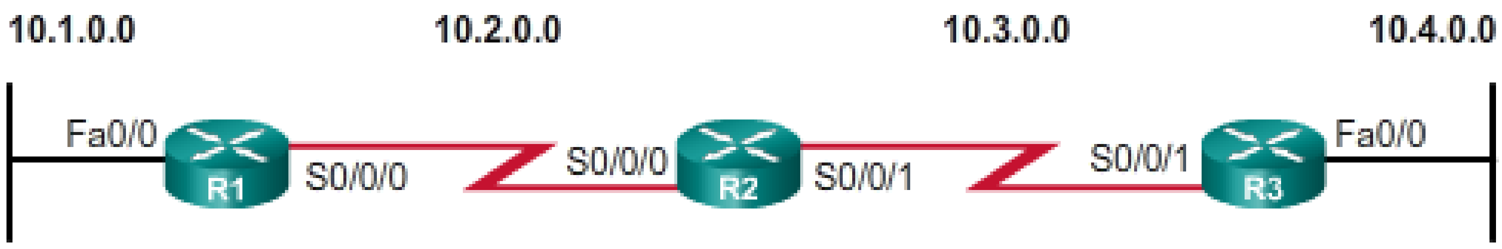
# Dynamic Routing Protocol Operation




- In general, the operations of a dynamic routing protocol can be described as follows:
  - The router sends and receives ***routing messages on its interfaces***.
  - The router ***shares routing messages and routing information*** with other routers that are using the *same routing prot*ocol.
  - Routers ***exchange routing information*** to learn about ***remote networks.***
  - When a router detects a ***topology change*** the routing protocol can ***advertise this change to other routers***.

# Cold Start

- When a router powers up, it knows nothing about the network topology.
- After it boots successfully, it applies the saved start-up configuration from NVRAM (if there is one).
- The router initially discovers its own directly connected networks.

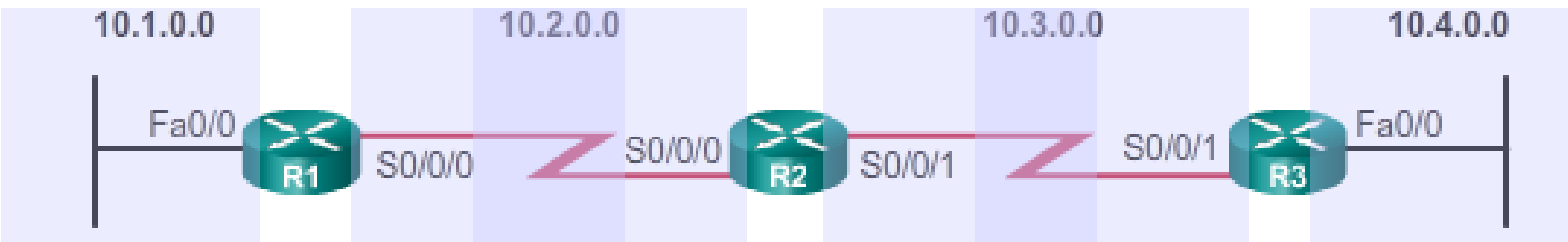





Note: Routers running RIPv2

# Discover Directly Attached Networks

- R1 adds the 10.1.0.0 network available through interface FastEthernet 0/0 and 10.2.0.0 is available through interface Serial 0/0/0.
- R2 adds the 10.2.0.0 network available through interface Serial 0/0/0 and 10.3.0.0 is available through interface Serial 0/0/1.
- R3 adds the 10.3.0.0 network available through interface Serial 0/0/1 and 10.4.0.0 is available through interface FastEthernet 0/0.




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| | | | |
| | | | |

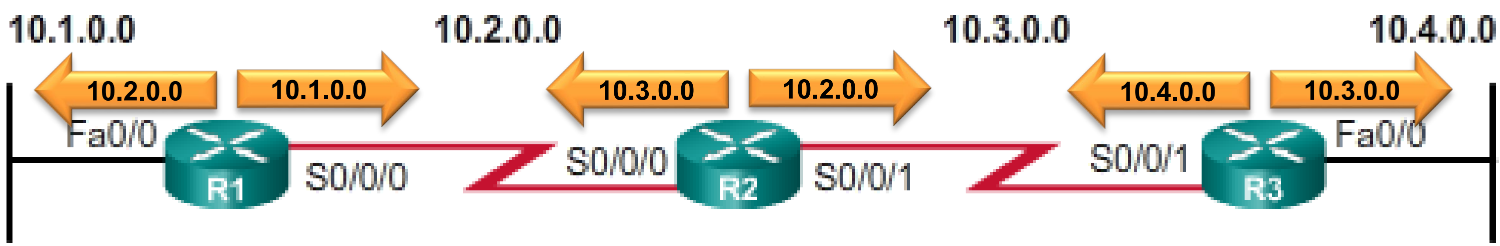
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| | | | |
| | | | |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| | | | |
| | | | |

This example is only for ***distance vector*** routing protocols (not link state routing protocols).
More later!

# Network Discovery – Initial Exchange

- R1 sends an update about network 10.1.0.0 out the Serial0/0/0 interface and sends an update about network 10.2.0.0 out of Fa0/0.
- R2 sends an update about network 10.3.0.0 out of Serial 0/0/0 and sends an update about network 10.2.0.0 out of Serial 0/0/1.
- R3 sends an update about network 10.4.0.0 out of Serial 0/0/1 and sends an update about network 10.3.0.0 out of FastEthernet0/0.




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| | | | |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| | | | |

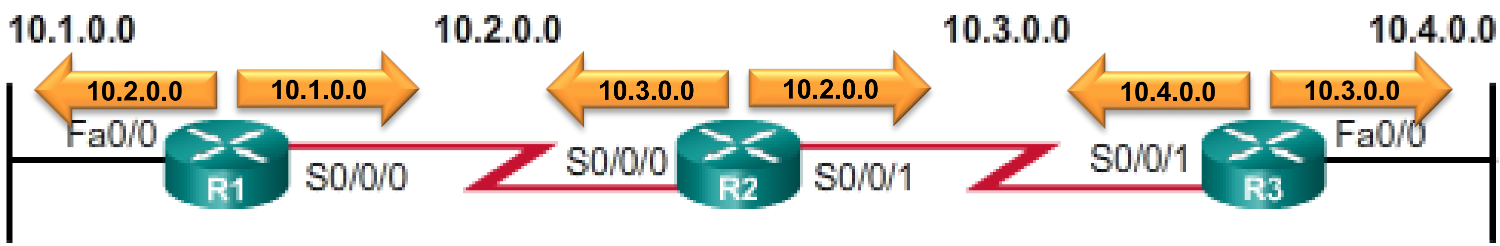
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| | | | |

Distance vector routing protocols implement a routing loop prevention technique known as **split horizon**.
Split horizon prevents information from being sent out the same interface from which it was received.
For example, R2 does not send an update containing the network 10.1.0.0 out of Serial 0/0/0, because R2 learned about network 10.1.0.0 through Serial 0/0/0.

This example is only for ***distance vector*** routing protocols (not link state routing protocols). More later!

# Update Routing Table

- R1 receives the update from R2 about network 10.3.0.0, increments the hop count by 1, and stores the network in the routing table (metric of 1).
- R2 receives the update from R1 about network 10.1.0.0 and R3 about network 10.4.0.0, increments and stores both networks in the routing table (metric of 1).
- R3 receives the update from R2 about network 10.2.0.0, increments the hop count by 1, and stores the network in the routing table (metric of 1).




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| | | | |
| | | | |

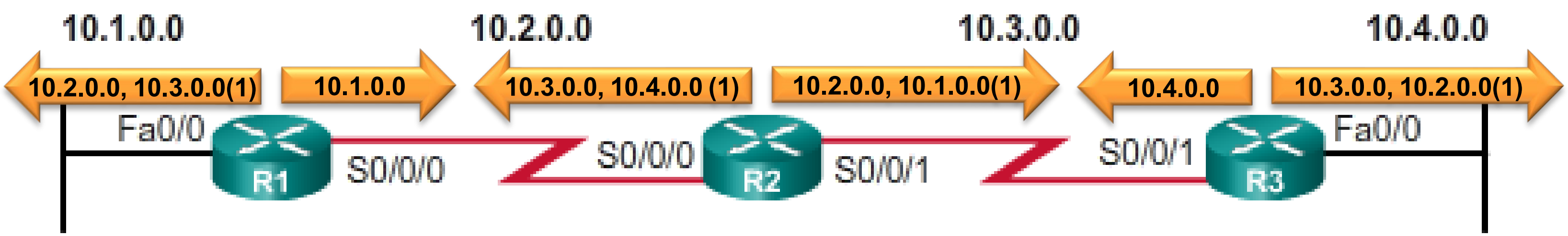
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| | | | |
| | | | |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| | | | |
| | | | |

This example is only for ***distance vector*** routing protocols (not link state routing protocols). More later!

# Network Discovery – Next Exchange

- R1 sends an update about 10.1.0.0 out Serial0/0/0 and networks 10.2.0.0 and 10.3.0.0 (metric 1) out of Fa0/0.
- R2 sends an update about 10.3.0.0 and 10.4.0.0 (metric 1) out Serial 0/0/0 and network 10.2.0.0 and 10.1.0.0 (metric 1) out of Serial 0/0/1.
- R3 sends an update about 10.4.0.0 out Serial 0/0/1 and networks 10.2.0.0 (metric 1) and 10.3.0.0 out FastEthernet0/0.




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| | | | |

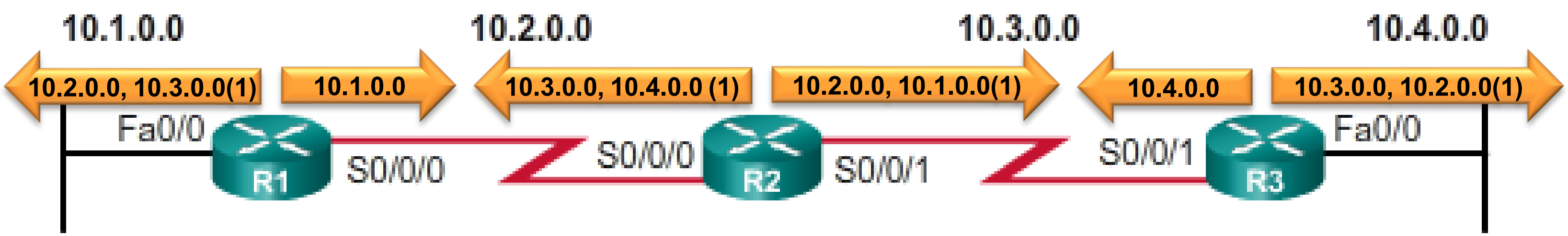
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/1 | 1 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| | | | |

This example is only for ***distance vector*** routing protocols (not link state routing protocols). More later!

# Update Routing Table

- R1 receives the update from R2 about network 10.3.0.0 and 10.4.0.0 (1).
  - Refreshes the information for 10.3.0.0.
  - It increments the 10.4.0.0 hop count by 1 and stores it in the routing table (metric 2).
  - ***10.4.0.0 is 1 hop for R2, so 2 hops for R1***




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| | | | |

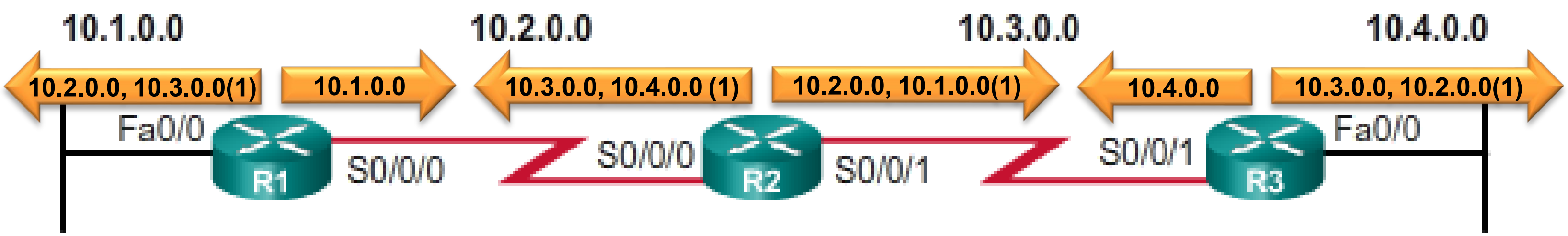
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| | | | |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| | | | |

This example is only for ***distance vector*** routing protocols (not link state routing protocols). More later!

# Update Routing Table

- R2 receives the update from R1 about network 10.1.0.0 and from R3 about network 10.4.0.0 and refreshes the routing table.
- ***No changes to routing table because it didn't learn any new routes or of a better route to any networks.***




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| **R** | **10.4.0.0** | **S0/0/0** | **2** |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/1 | 1 |

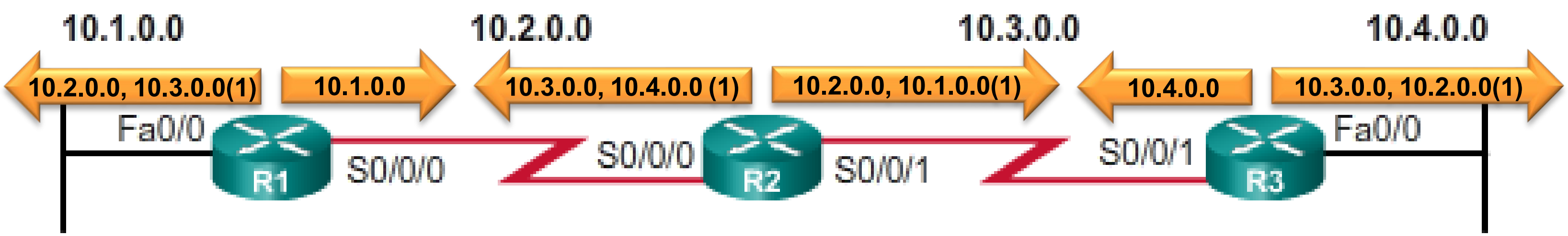
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| | | | |

This example is only for ***distance vector*** routing protocols (not link state routing protocols). More later!

# Update Routing Table

- R3 receives the update from R2 about network 10.1.0.0 (1) and 10.2.0.0.
  - Refreshes the information for 10.2.0.0.
  - It increments the 10.1.0.0 hop count by 1 and stores it in the routing table (metric 2).
  - ***10.1.0.0 is 1 hop for R2 so 2 hops or R3***

**Network Has Converged**




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| **R** | **10.4.0.0** | **S0/0/0** | **2** |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/1 | 1 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| | | | |

# Split Horizon

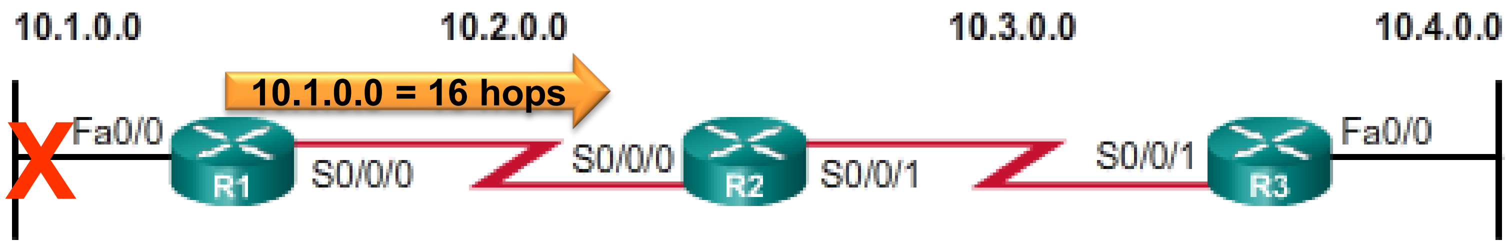

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/0 | 2 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/1 | 1 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| R | 10.1.0.0 | S0/0/1 | 2 |

- **Split horizon** rule says that a *router should not advertise a network through the interface from which the update came*.
  - Helps prevent routing loops caused by slow convergence.

# Split Horizon – Route Poisoning



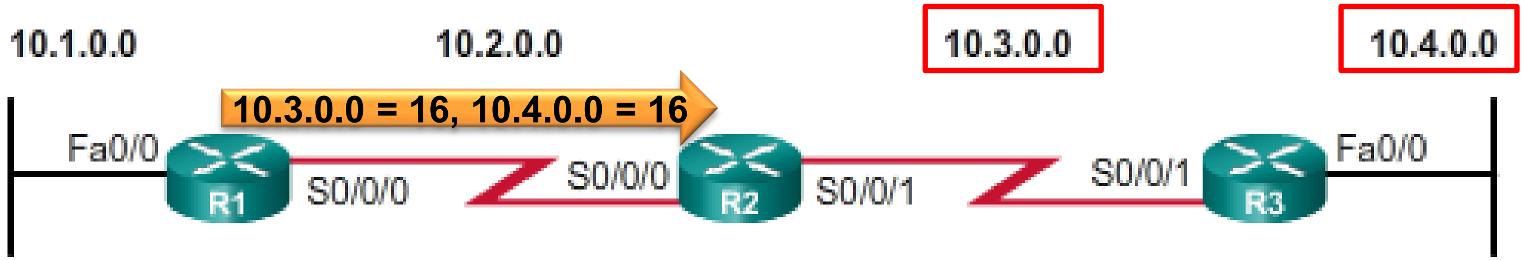

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/0 | 2 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/1 | 1 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| R | 10.1.0.0 | S0/0/1 | 2 |

- **Route poisoning** is used to mark the route as unreachable in a routing update that is sent to other routers.
  - **Unreachable** is interpreted as a metric that is set to the maximum.
  - For **RIP**, a poisoned route has a metric of 16.
- Route poisoning speeds the convergence process.

# Split Horizon with Poison Reverse




| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.1.0.0 | Fa0/0 | 0 |
| C | 10.2.0.0 | S0/0/0 | 0 |
| R | 10.3.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/0 | 2 |

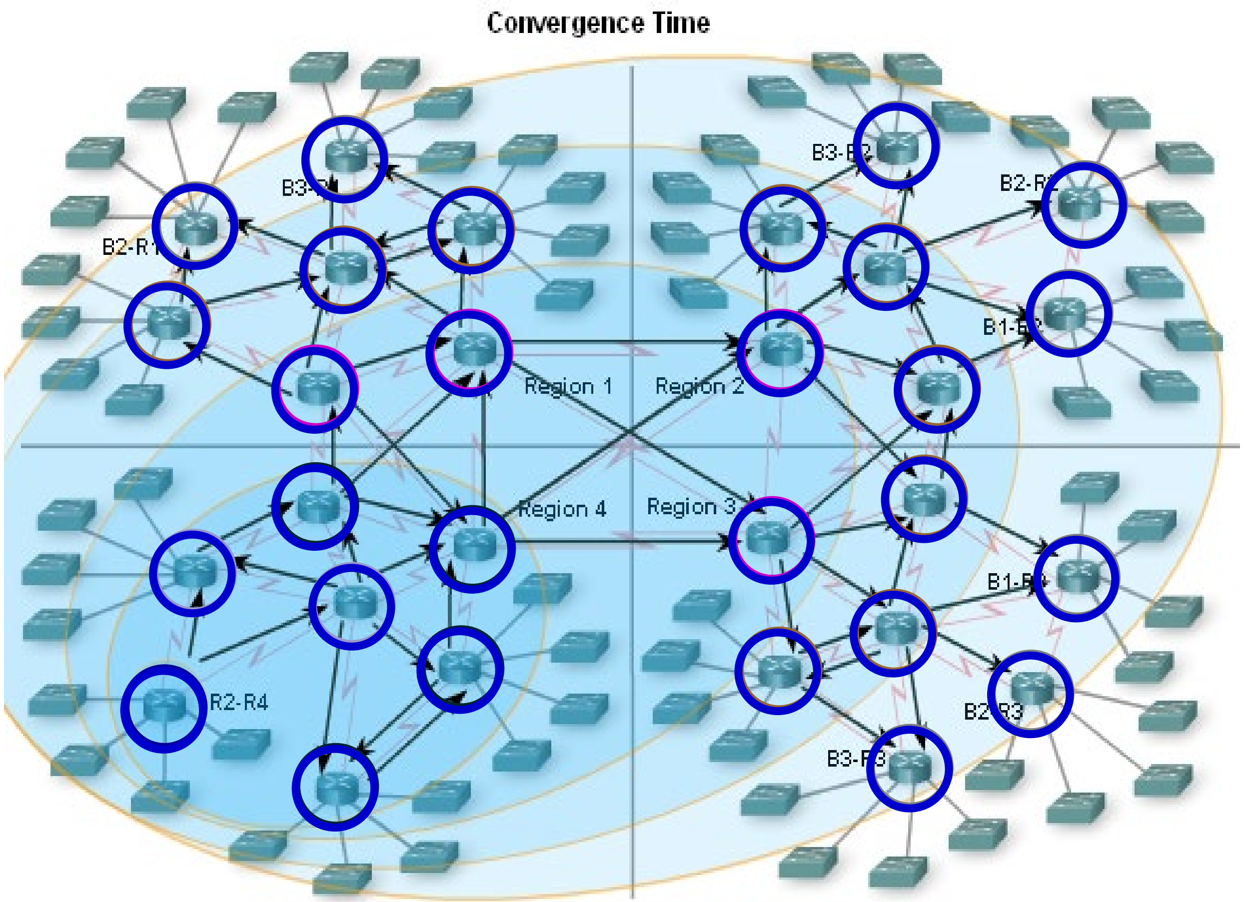
| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.2.0.0 | S0/0/0 | 0 |
| C | 10.3.0.0 | S0/0/1 | 0 |
| R | 10.1.0.0 | S0/0/0 | 1 |
| R | 10.4.0.0 | S0/0/1 | 1 |

| | Network | Interface | Hop |
|---|---|---|---|
| C | 10.3.0.0 | S0/0/1 | 0 |
| C | 10.4.0.0 | Fa0/0 | 0 |
| R | 10.2.0.0 | S0/0/1 | 1 |
| R | 10.1.0.0 | S0/0/1 | 2 |

- **Split horizon with poison reverse**
- The concept of split horizon with poison reverse is that explicitly telling a router to ignore a route is better than not telling it about the route in the first place.

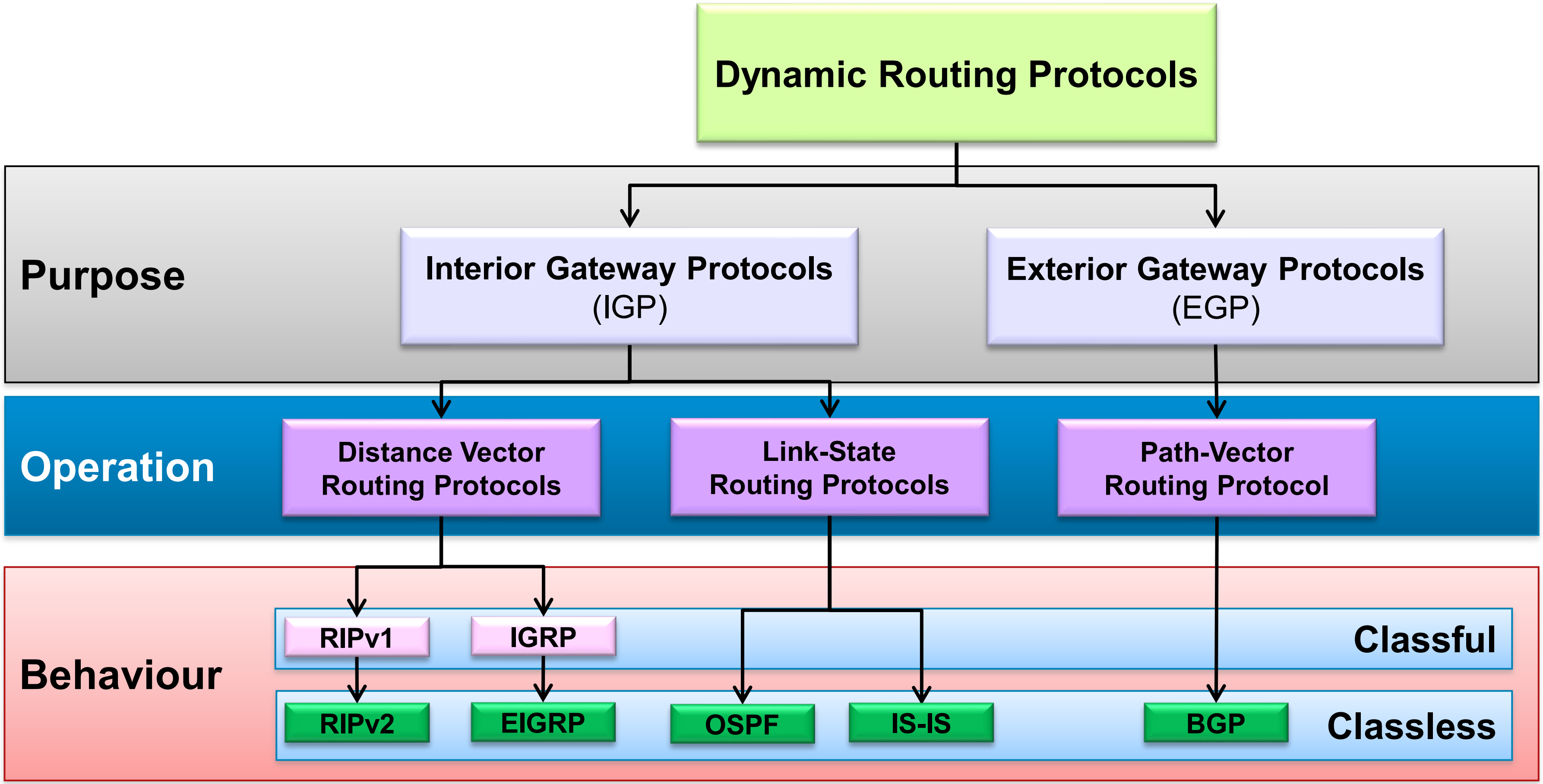
# Network Discovery - Convergence

- **Convergence** is when a network has complete and accurate information about the entire network
- **Convergence time** is the time it takes routers to share information, calculate best paths, and update their routing tables.
- **NOTE**:
  - A network is not completely operable until the network has converged.



# Routing Protocol Classification

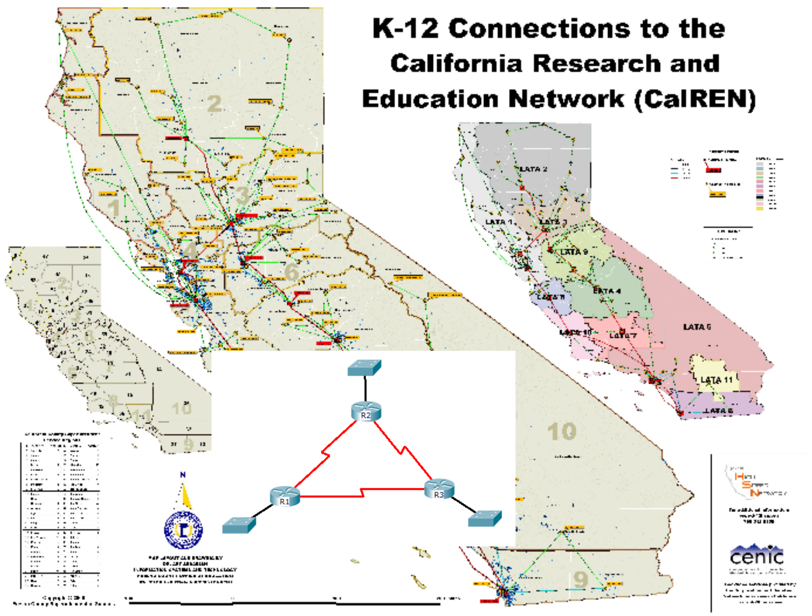
# Classifying Routing Protocols

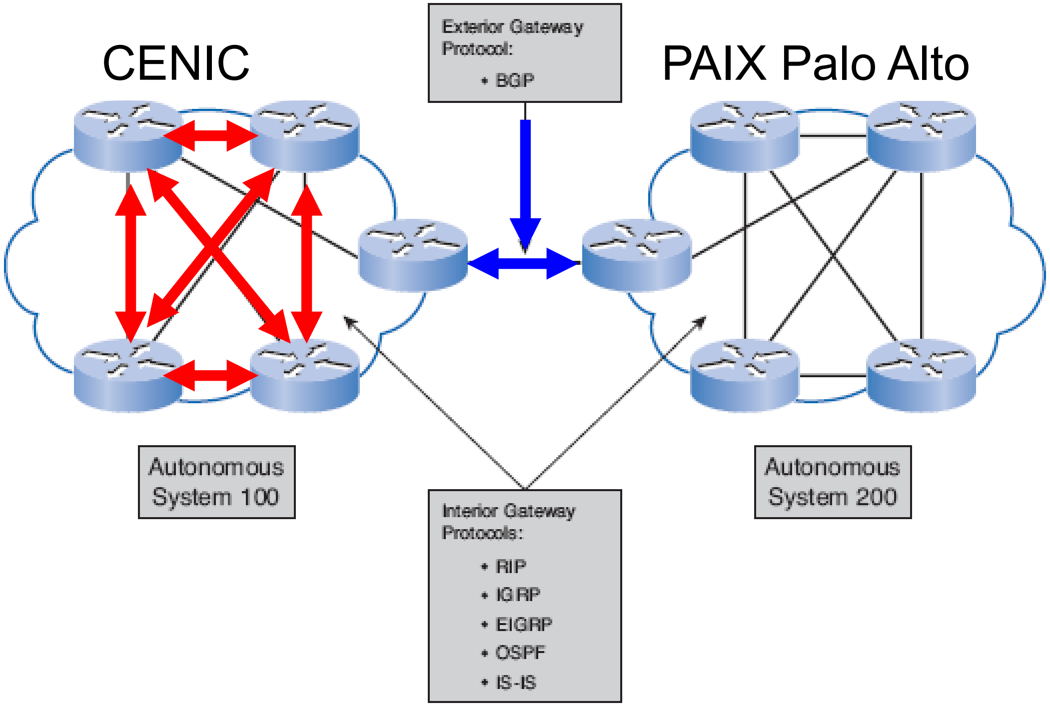



RIPv1 and IGRP are legacy protocols that have evolved into the classless routing protocols, RIPv2 and EIGRP.

Link-state routing protocols are classless by nature.

# IGP and EGP

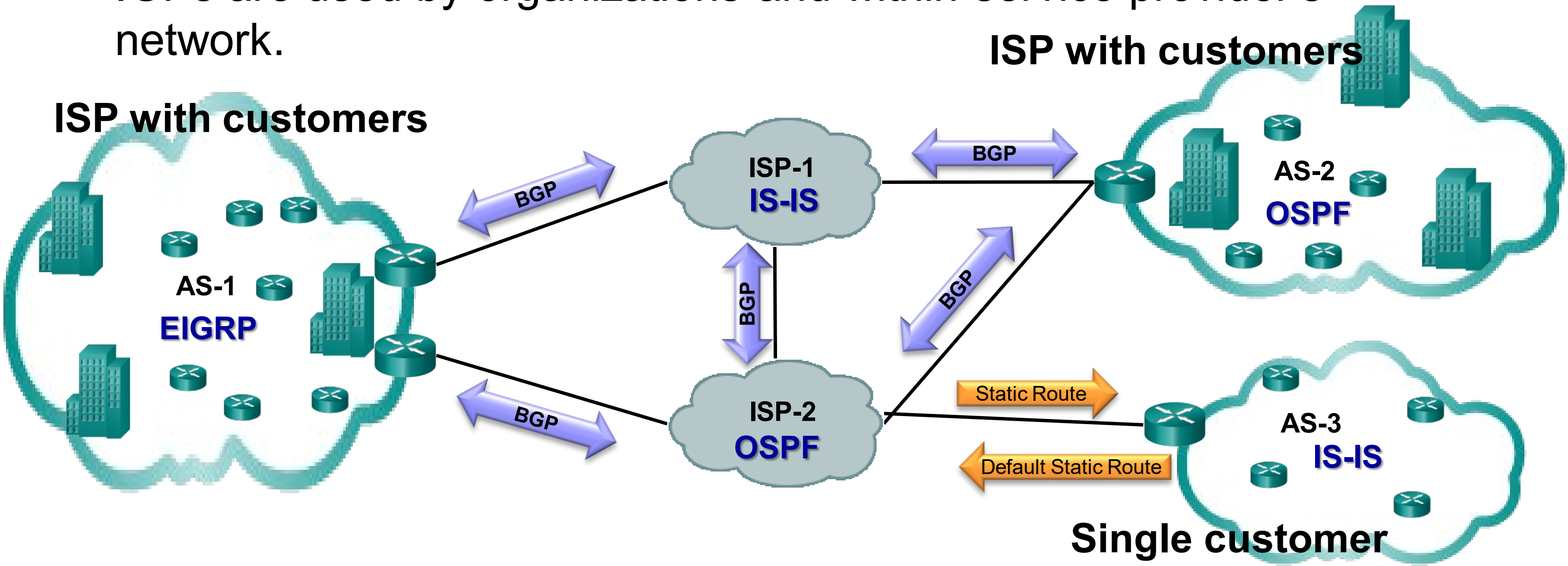



- An ***autonomous system*** (AS)— is a collection of routers under a <u>common administration</u>.
  - Company's internal network
  - An ISP's network.
- Because the <u>Internet is based on the autonomous system concept</u>, two types of routing protocols are required:
  - **Interior Gateway Protocols IGP (RIP, EIGRP, OSPF, IS-IS):**
    - Routing inside an autonomous system
  - **Exterior Gateway Protocols (BGP):** Between ISPs, CENIC and PAIX, and some customers (usually just say BGP).
    - Routing between autonomous systems

# IGP versus EGP Routing Protocols

- IGPs are used by organizations and within service provider's network.




- BGP could be used to interconnect large organizations to service providers and in between various service providers.
- Smaller organizations would typically connect using static routes but could also use BGP.

# Routing Protocol Operation

Distance Vector and Link-state

# Distance Vector Routing Protocol Operation

- **What does a street sign like this tell you?**
  - **How far (distance)**
  - **Which way (direction)**
- ***Distance vector***
  - Routes are advertised as ***vectors*** of distance and direction.
- **Distance** is defined in terms of a metric
  - Such as hop count
- **Direction** is simply the:
  - Nexthop router or
  - Exit interface
- Typically use the **Bellman-Ford algorithm** for the best-path (shortest) route determination

# Distance Vector Routing Protocol Operation


- Routing protocol
  - Does not know the topology of an internetwork.
  - Only knows the routing information received from its neighbors.
  - Does not know if another path would actually be faster.

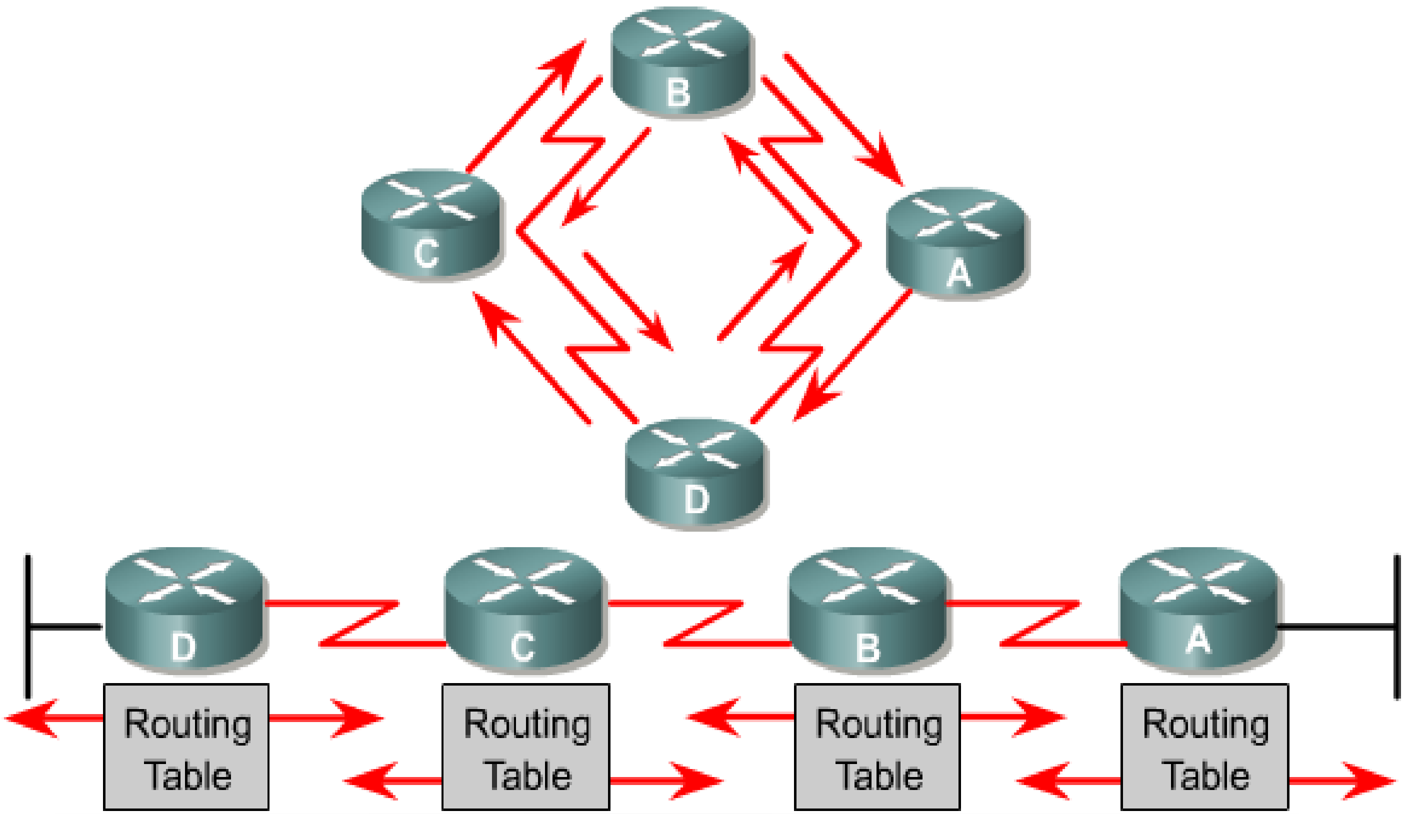
Would another path that is longer actually be faster? (speed limit)

I don’t have a map of the network.

All I know is how far and which direction (to next hop router)

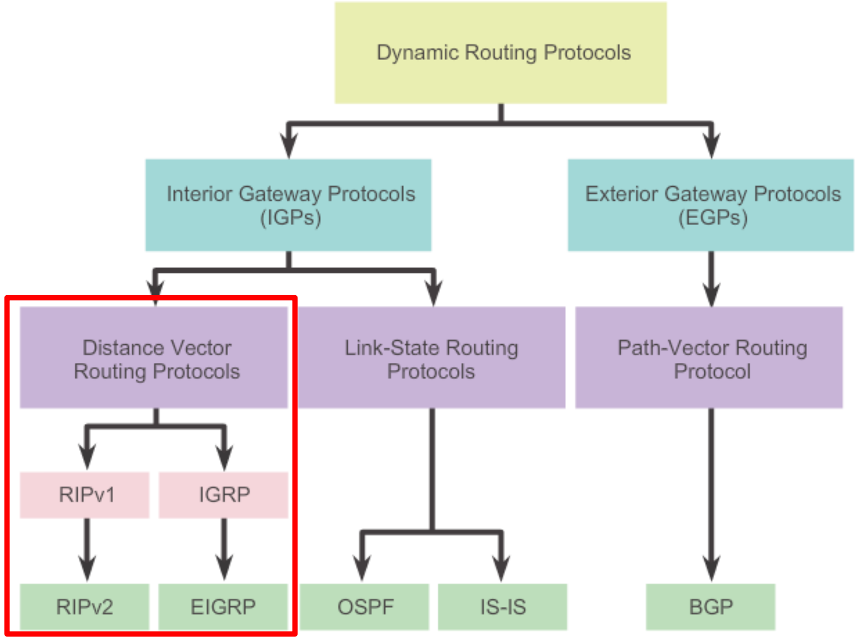
Distance Vector routing protocols are like signposts along the path to the final destination.

# Distance Vector Concepts




Pass periodic copies of a routing table to neighbor routers and accumulate distance vectors.

# Distance Vector Routing Protocols

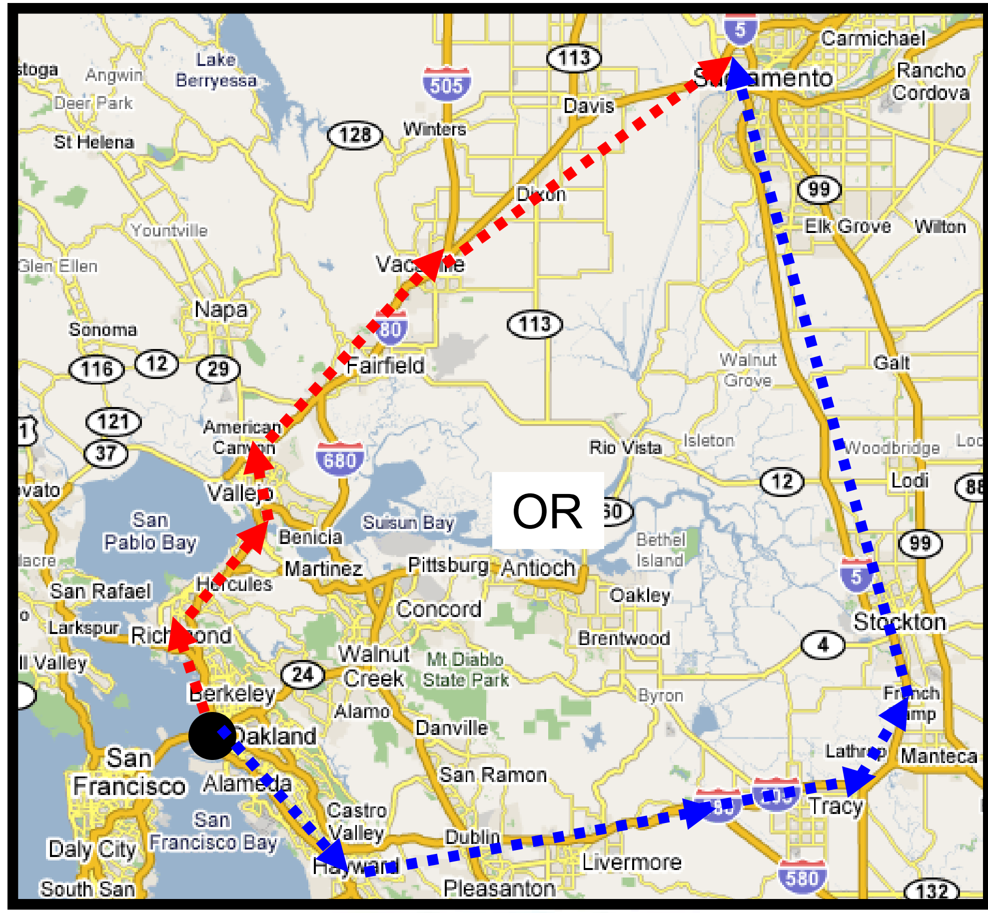



- **Routing Information Protocol (RIP)**
  - Three versions: IPv4 RIPv1 and RIPv2. RIPng for IPv6.
  - Uses hop counts as its metric.
- **Interior Gateway Routing Protocol (IGRP)**
  - Legacy Cisco Proprietary protocol.
  - Uses bandwidth and delay as its metric.
- **Enhanced IGRP (EIGRP)**
  - Cisco Proprietary protocol. Now RFC-7868.
  - Uses bandwidth and delay as its metric.
  - Only event driven distance-vector routing protocol.

**NOTE:**
•**EIGRP** is an "advanced" DV protocol.
•When we discuss DV protocols in general terms, we are referring more to **RIP** than EIGRP.

# Link-State Protocol Operation

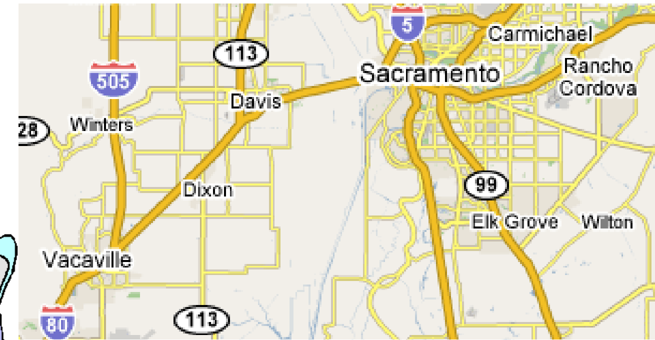
- ***Link-state*** routing protocol can create a "complete view," or topology, of the network.
- Link-state protocols are associated with Shortest Path First (SPF) calculations.
- A ***link-state router*** uses the link-state information to:
  - Create a topology map
  - Select the best path to all destination networks in the topology.
  - Each router makes the decision!

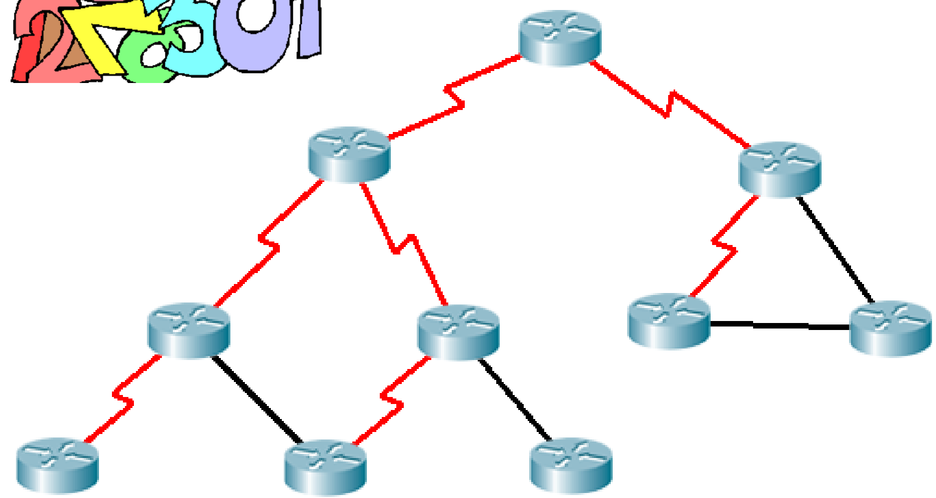



Link State routing protocols is like having a complete map of the network topology

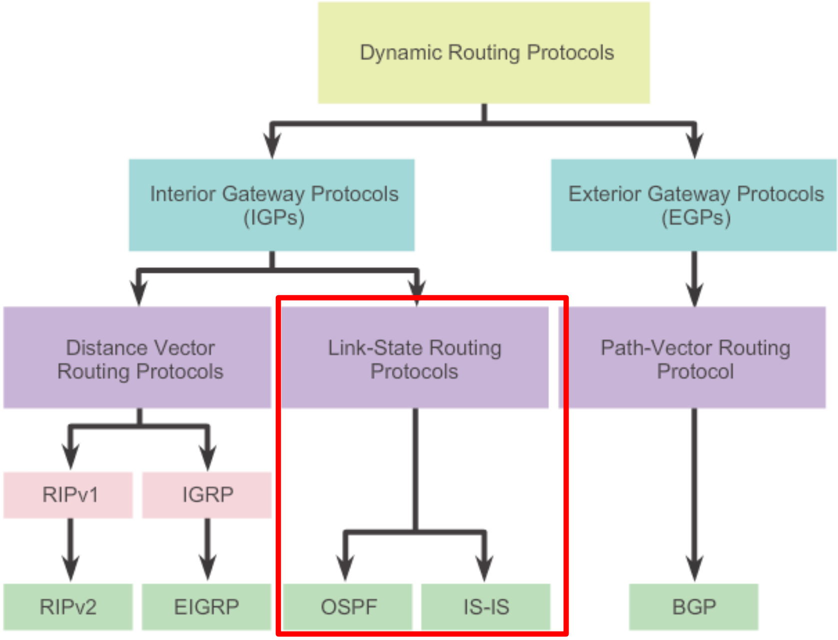
# Link-State Protocol Operation

- **Link-state protocols work best** in situations where
  - The <u>network design is hierarchical</u>, usually occurring in large networks.
  - The administrators have a <u>good knowledge</u> of the implemented <u>link-state routing protocol.</u>
  - <u>Fast convergence</u> of the network is crucial.



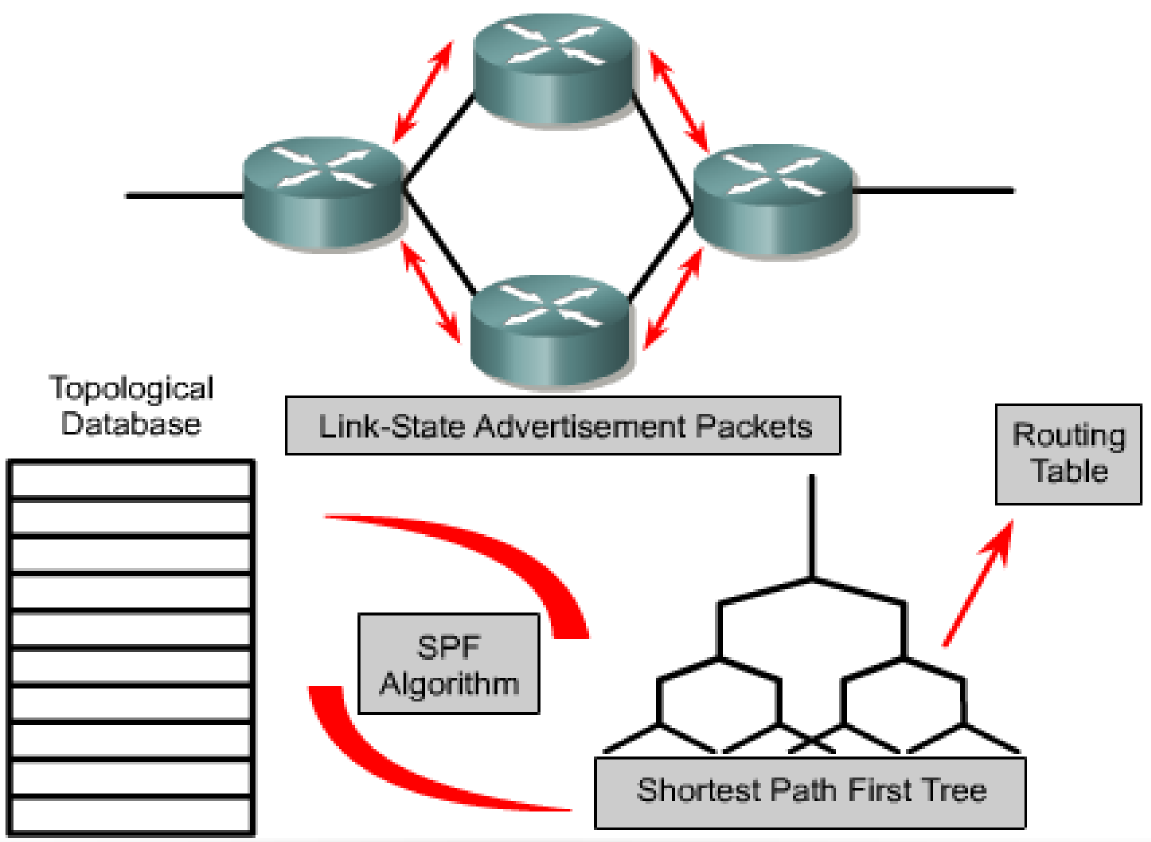



# Link-State Routing Protocols




- **Open Shortest Path First (OSPF)**
  - Popular standards based routing protocol
- **Intermediate System-to-Intermediate System (IS-IS)**
  - Popular in provider networks

# Link-State Concepts

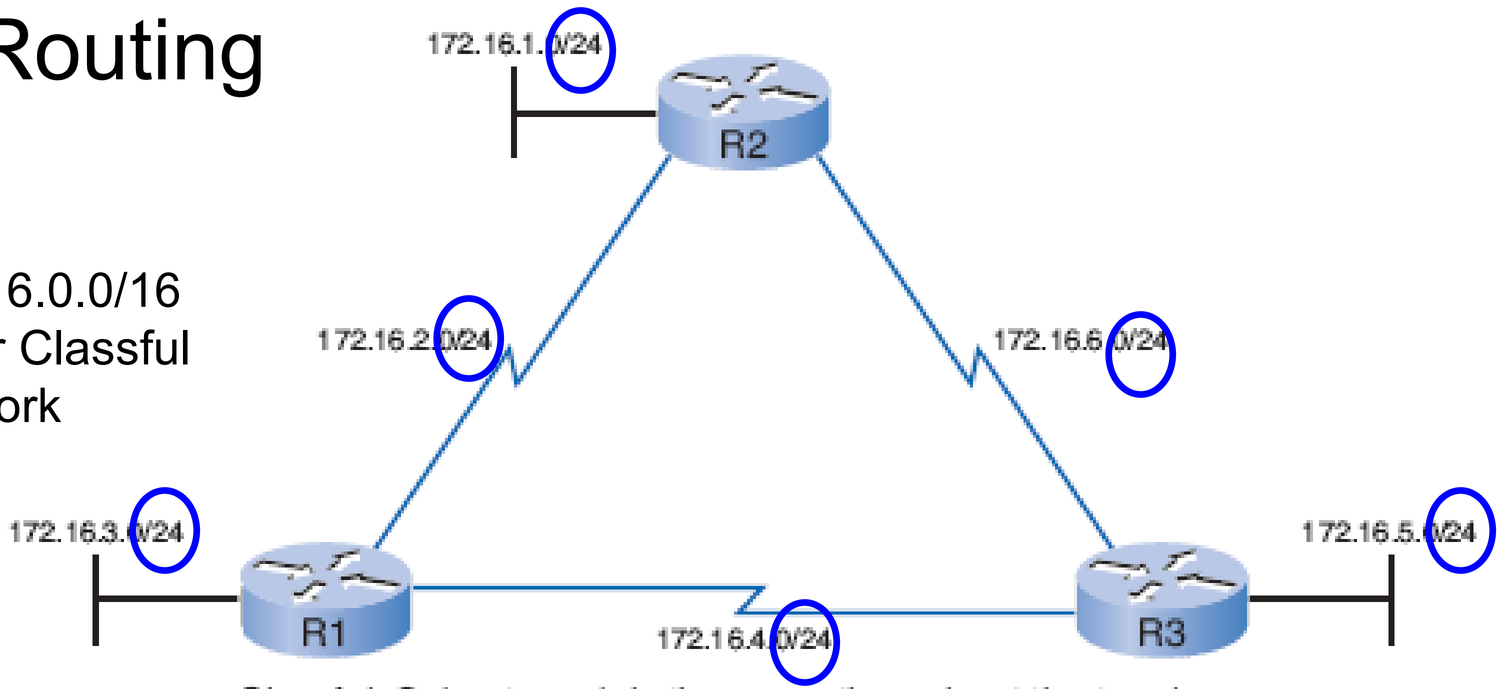



Routers send LSAs to their neighbors. The LSAs are used to build a topological database. The SPF algorithm is used to calculate the shortest path first tree in which the root is the individual router and then a routing table is created.

# Routing Protocol Behaviour

# Classful Routing Protocols

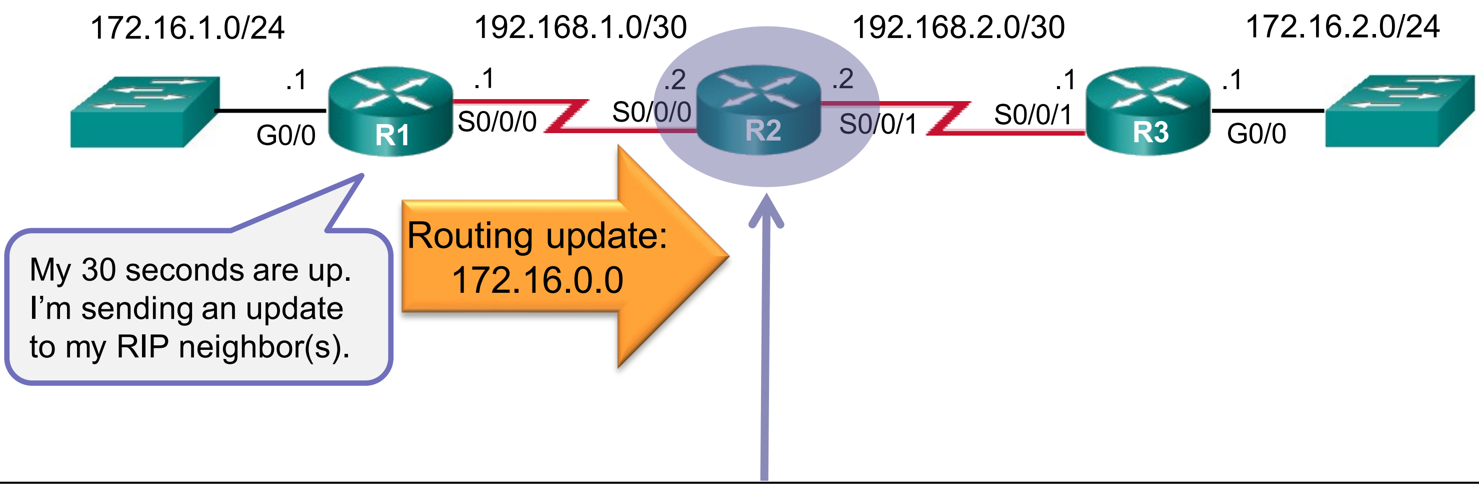
172.16.0.0/16
Major Classful
Network




Classful: Subnet mask is the same throughout the topology.

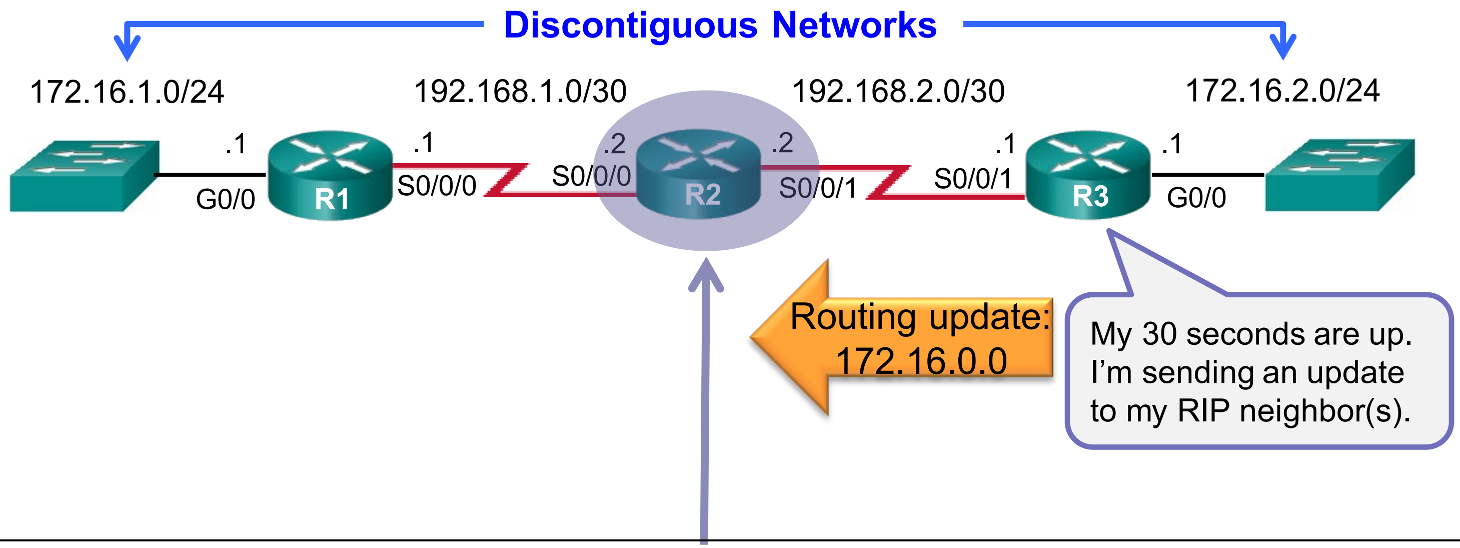
- Classful routing protocols do not send subnet mask information in their routing updates:
  - Only RIPv1 and IGRP are classful.
  - Created when network addresses were allocated based on classes (class A, B, or C).
  - Cannot provide variable length subnet masks (VLSMs) and classless interdomain routing (CIDR).
  - Create problems in discontiguous networks.

# R1 Forwards a Classful Update to R2



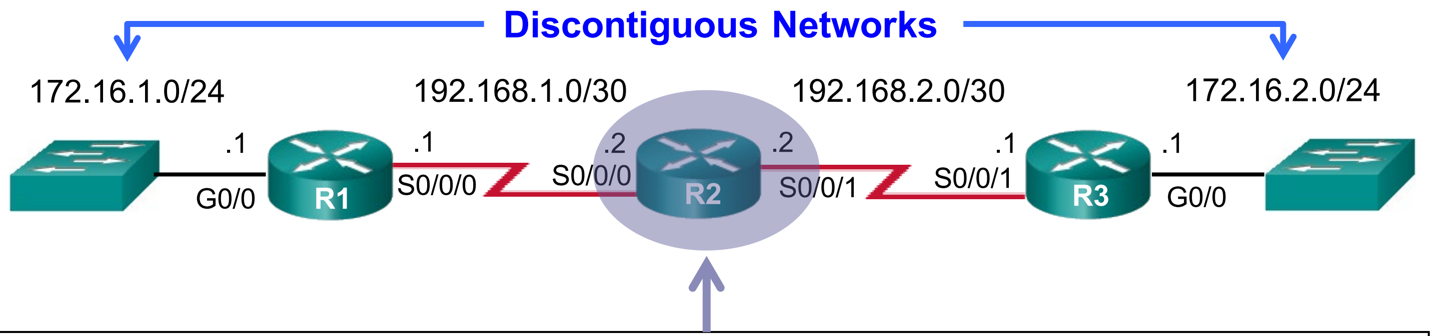

```
R2# show ip route | begin Gateway
Gateway of last resort is not set

R      172.16.0.0/16 [120/1] via 192.168.1.1, 00:00:11, Serial0/0/0
       192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.1.0/30 is directly connected, Serial0/0/0
L         192.168.1.2/32 is directly connected, Serial0/0/0
       192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.2.0/30 is directly connected, Serial0/0/1
L         192.168.2.2/32 is directly connected, Serial0/0/1
R2#
```

# R3 Forwards a Classful Update to R2



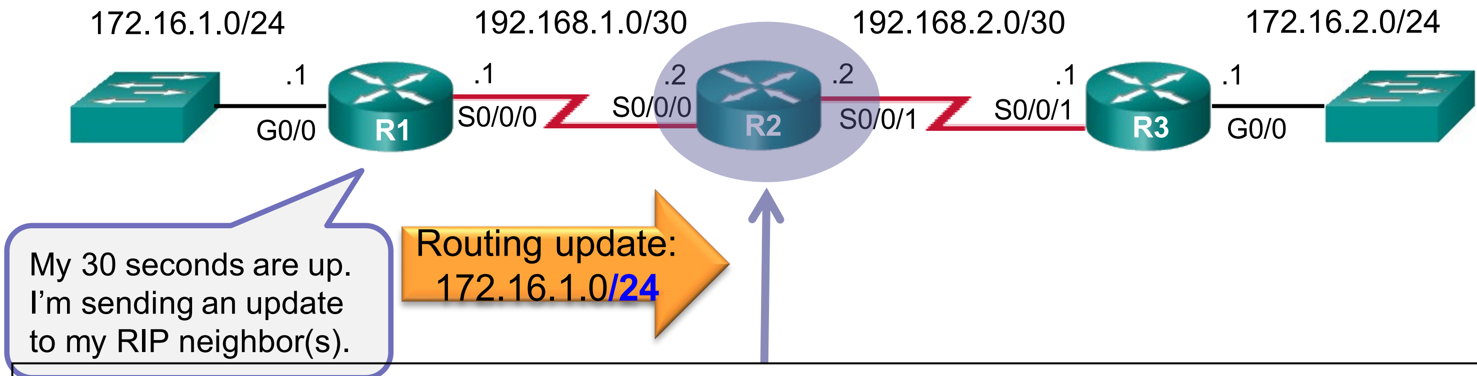

```
R2# show ip route | begin Gateway
Gateway of last resort is not set

R      172.16.0.0/16 [120/1] via 192.168.2.1, 00:00:14, Serial0/0/1
                     [120/1] via 192.168.1.1, 00:00:16, Serial0/0/0
       192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.1.0/30 is directly connected, Serial0/0/0
L         192.168.1.2/32 is directly connected, Serial0/0/0
       192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.2.0/30 is directly connected, Serial0/0/1
L         192.168.2.2/32 is directly connected, Serial0/0/1
R2#
```

# Connectivity Fails or Inconsistent at Best




```
R2# ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2
seconds:
U.U.U
Success rate is 0 percent (0/5)
R2#
R2# traceroute 172.16.1.1
Type escape sequence to abort.
Tracing the route to 172.16.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.1.1 4 msec
    192.168.2.1 4 msec
    192.168.1.1 4 msec
R2#
```

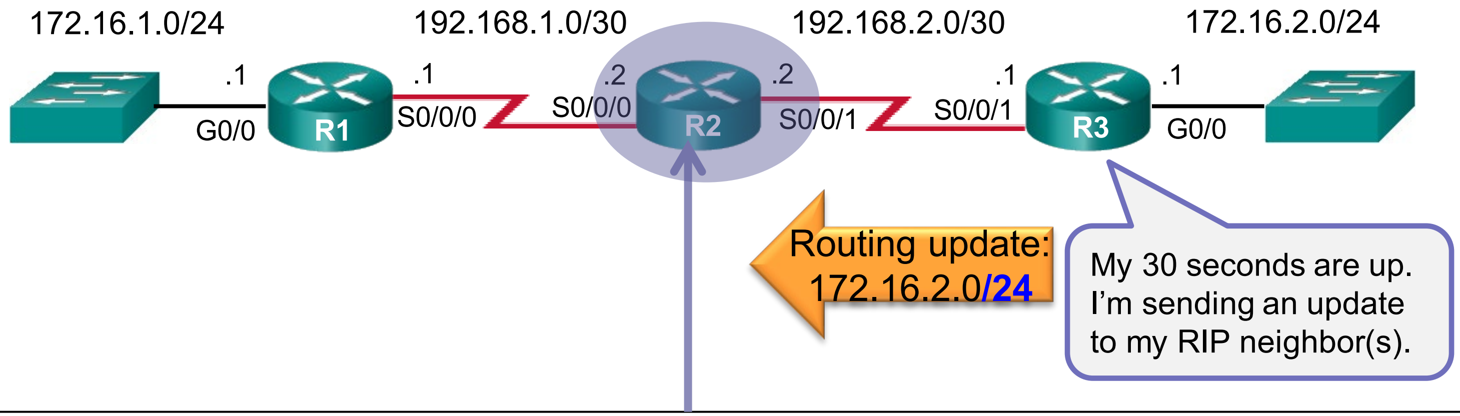
# Classless Routing Protocols




```
R2# show ip route | begin Gateway
Gateway of last resort is not set

      172.16.0.0/24 is subnetted, 1 subnets
R        172.16.1.0 [120/1] via 192.168.1.1, 00:00:06, Serial0/0/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
R2#
```

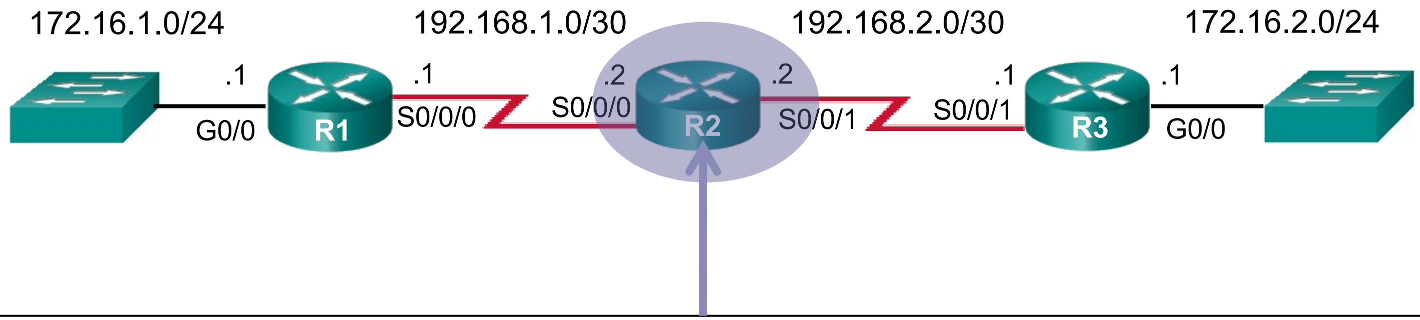
- Classless routing protocols include subnet mask information in the routing updates:
  - RIPv2, EIGRP, OSPF, and IS_IS and IPv6 routing protocols
  - Support VLSM and CIDR
  - Also support discontiguous networks

# R3 Forwards a Classless Update to R2




```
R2# show ip route | begin Gateway
Gateway of last resort is not set

      172.16.0.0/24 is subnetted, 2 subnets
R        172.16.1.0 [120/1] via 192.168.1.1, 00:00:03, Serial0/0/0
R        172.16.2.0 [120/1] via 192.168.2.1, 00:00:03, Serial0/0/1
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, Serial0/0/0
L        192.168.1.2/32 is directly connected, Serial0/0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/30 is directly connected, Serial0/0/1
L        192.168.2.2/32 is directly connected, Serial0/0/1
R2#
```

# Connectivity Success



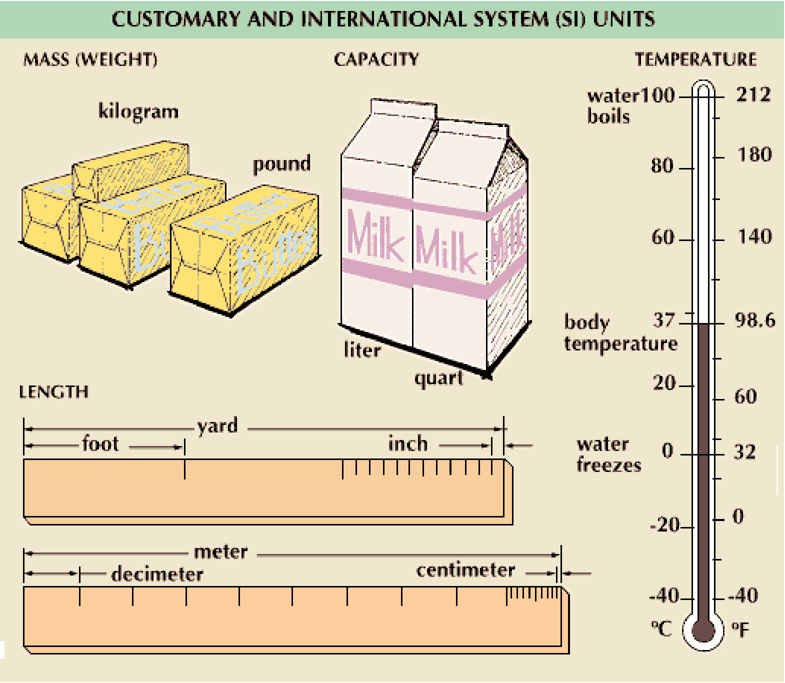

```
R2# ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
R2#
R2# traceroute 172.16.1.1
Type escape sequence to abort.
Tracing the route to 172.16.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 192.168.1.1 4 msec 4 msec *
R2#
```

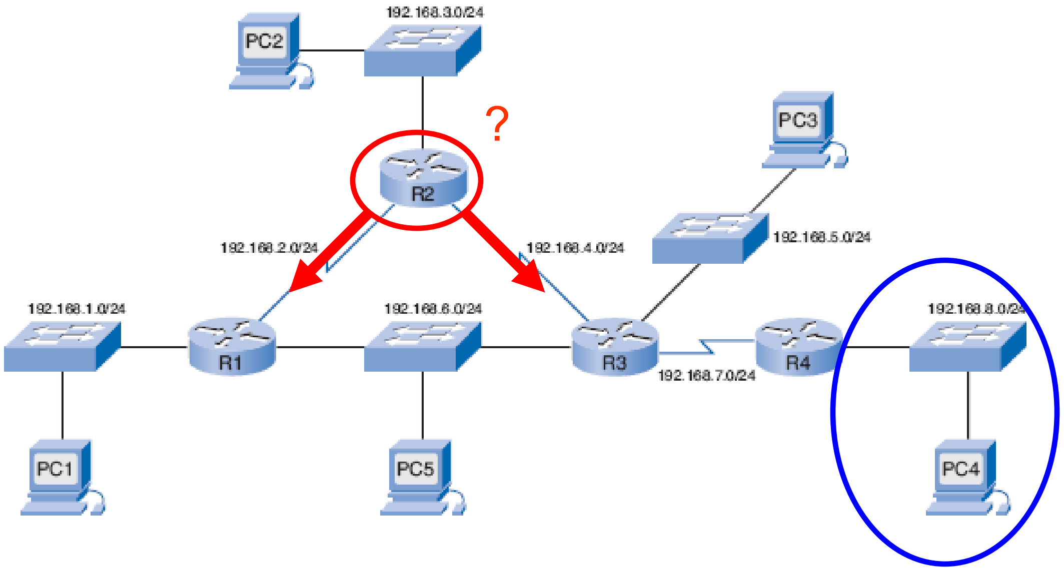
# Routing Protocol Characteristics

- **Speed of Convergence:**
  - Routing loops can occur when inconsistent routing tables are not updated due to slow convergence in a changing network.
- **Scalability:**
  - Large networks require a scalable the routing protocol.
- **Classful or Classless (Use of VLSM and summarization):**
  - Classful routing protocols do not include the subnet mask and cannot support VLSM while classless routing protocols do.
- **Resource Usage:**
  - Defines how much memory space (RAM), CPU utilization, and link bandwidth utilization is required by the routing protocol.
- **Implementation and Maintenance:**
  - This describes the level of knowledge that is required for a network administrator to implement and maintain the network based on the routing protocol deployed.
  - My opinion…. Link state is easier to understand, maintain and troubleshoot the distance vector.

# Metrics

# Purpose of a Metric

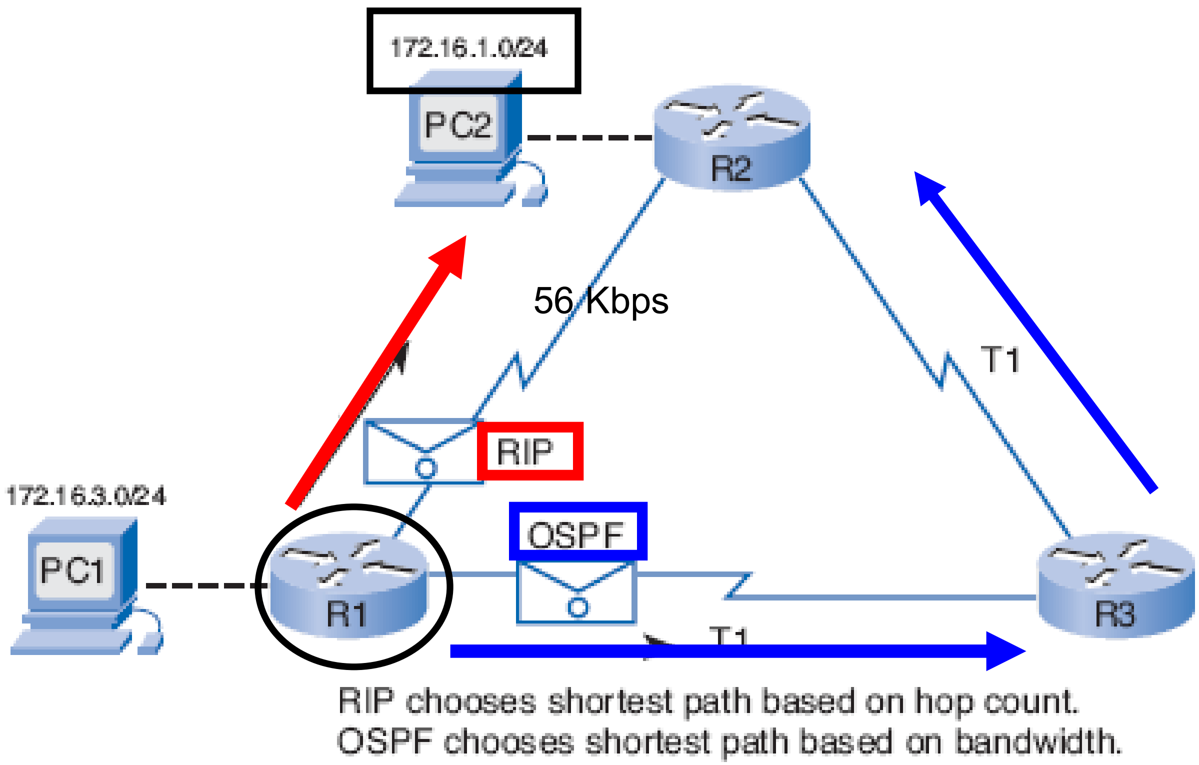



- **Metrics** are a way to measure or compare.
  - Determine the best path.
- **Routing protocol** learns multiple routes to the same destination.
  - Metric is used to determine which path is most preferable
  - Lower the metric, the better

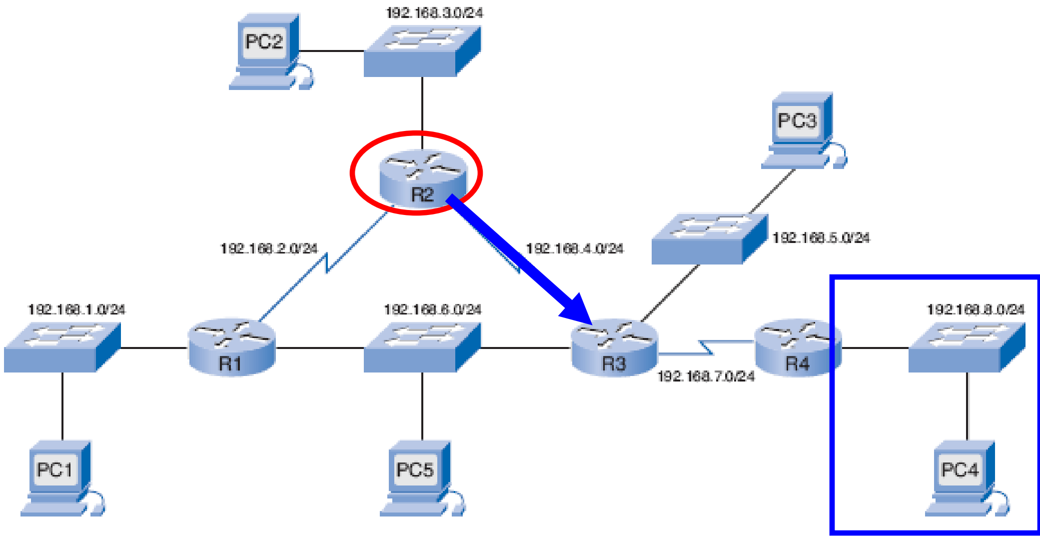
# Routing Protocol Metrics

- Different routing protocols use different metrics.
  - Routing metrics are not interchangeable between routing protocols.
  - Two different routing protocols might choose different paths to the same destination.
- For example:

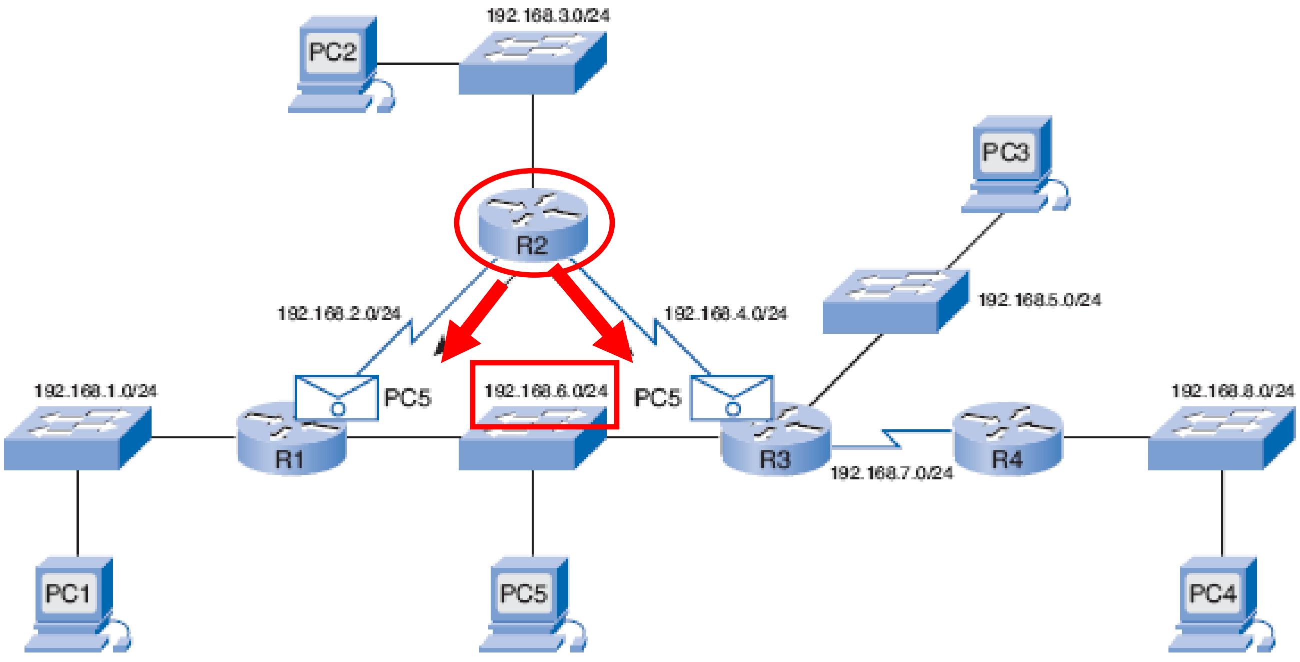


```
R2# show ip route
<output omitted>
Gateway of last resort is not set
R     192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
C     192.168.2.0/24 is directly connected, Serial0/0/0
C     192.168.3.0/24 is directly connected, FastEthernet0/0
C     192.168.4.0/24 is directly connected, Serial0/0/1
R     192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R     192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                     [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R     192.168.7.0/24 [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
R     192.168.8.0/24 [120/2] via 192.168.4.1, 00:00:26, Serial0/0/1
```

- All routers running RIP
- What is the metric (distance) for R2 to reach the 192.168.8.0 network?
  - **2 (hops away)**
- What is the direction (vector) for R2 to reach the 192.168.8.0 network?
  - **Serial 0/0/1 (via R3)**



# Load Balancing




R2 load balances traffic destined for the 192.168.6.0/24 network.

```
R2# show ip route
<output omitted>
R     192.168.6.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0
                     [120/1] via 192.168.4.1, 00:00:26, Serial0/0/1
```

- What happens when two or more routes to the same destination have identical metric values?
  - The router **load balances** between these equal-cost paths.
  - All routing protocols do equal cost load balancing.
  - EIGRP also does unequal cost load balancing.

# Purpose of Administrative Distance

- What if a router learns about a remote network from two different routing sources.
- What if RIP advertises the network as 10 hops away but OSPF advertises it as a cumulative bandwidth of 100,000.
- Which is better RIP or OSPF?
  - Can't tell
  - Can't compare apples and oranges.
  - Note: This is not common.
- ***Administrative distance (AD)*** is:
  - Used to determine which routing source takes precedence.
  - Used when there are multiple routing sources for the same destination network address.
- **Lower the AD** the more preferred the routing source.

| Route Source | AD |
|---|---|
| Connected | 0 |
| Static | 1 |
| EIGRP summary route | 5 |
| E[illegible] | 20 |
| I[illegible] | 90 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| [illegible] | 170 |
| [illegible] | 200 |

So, which one would be preferred RIP or OSPF? OSPF

Which route would be preferred, OSPF or a Static Route to the same network?
Static Route

# Purpose of Administrative Distance

- Cisco uses Administrative distance (AD) to define the preference of a routing source.
- Routing sources:
  - Directly connected networks
  - Static routes
  - Specific routing protocols
- AD for static and dynamic can be modifed (in CCNP)

**Note**

- The term ***trustworthiness*** is commonly used when defining administrative distance.
- *The lower the administrative distance value, the more trustworthy the route.*

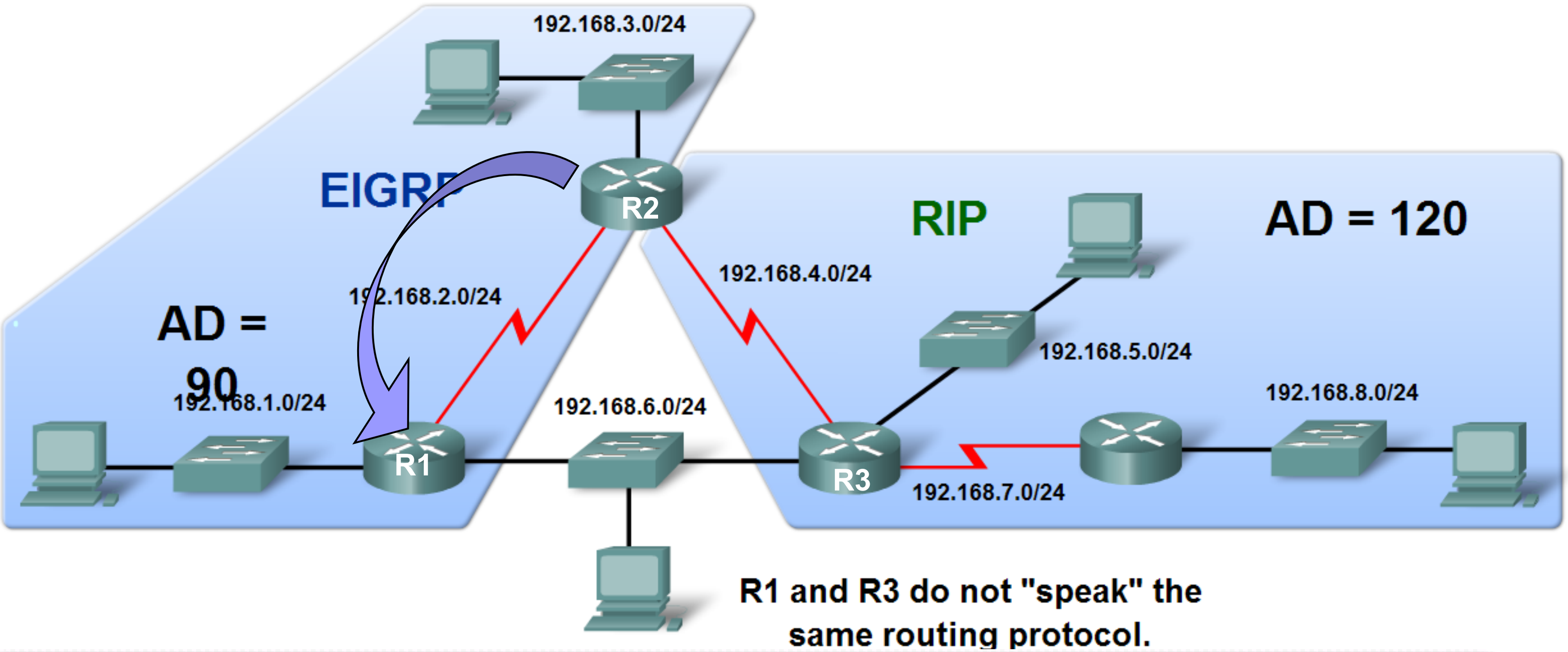
| Route Source | AD |
|---|---|
| Connected | 0 |
| Static | 1 |
| EIGRP summary route | 5 |
| External BGP | 20 |
| Internal EIGRP | 90 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| External EIGRP | 170 |
| Internal BGP | 200 |

# Purpose of Administrative Distance

- AD: 0 to 255.
- The lower the value, the more preferred the route source.
- **AD of 0** is the most preferred.
  - *Only a directly connected network has an administrative distance of 0, which cannot be changed.*
  - No better route to a network than being directly connected to that network.
- **AD of 255** means the router will not believe the source of that route
  - Route will not be installed in the routing table.

| Route Source | AD |
|---|---|
| Connected | 0 |
| Static | 1 |
| EIGRP summary route | 5 |
| External BGP | 20 |
| Internal EIGRP | 90 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| External EIGRP | 170 |
| Internal BGP | 200 |

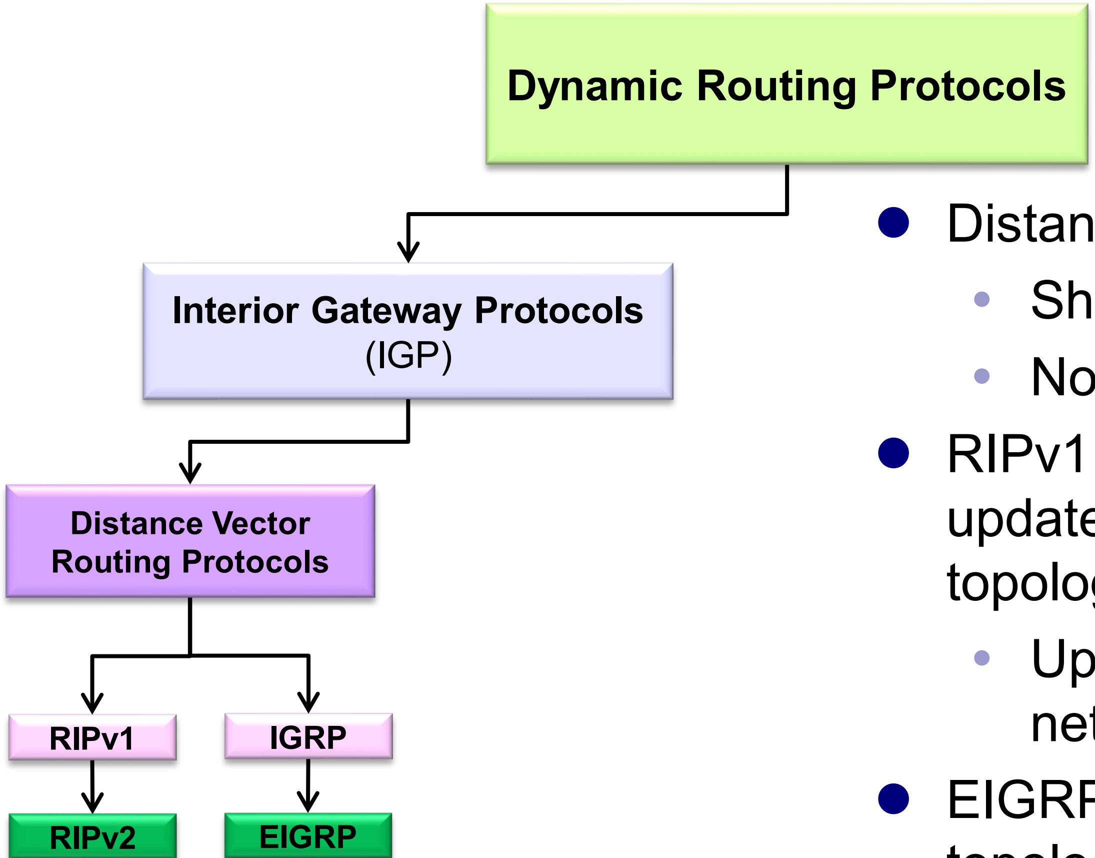
# Administrative Distance of a Route




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

Gateway of last resort is not set

D    192.168.1.0/24 [90/2172416] via 192.168.2.1, 00:00:24, Serial0/0/0
C    192.168.2.0/24 is directly connected, Serial0/0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/0
C    192.168.4.0/24 is directly connected, Serial0/0/1
R
  D    192.168.6.0/24 [90/2172416] via 192.168.2.1, 00:00:24, Serial0/0/0
D    192.168.6.0/24 [90/2172416] via 192.168.2.1, 00:00:24, Serial0/0/0
R    192.168.7.0/24 [120/1] via 192.168.4.1, 00:00:08, Serial0/0/1
R    192.168.8.0/24 [120/2] via 192.168.4.1, 00:00:08, Serial0/0/1
```

# Distance Vector Routing Protocol

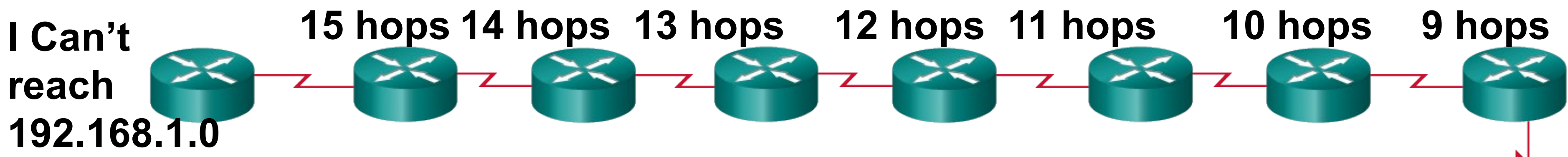
# Distance Vector Routing Protocols

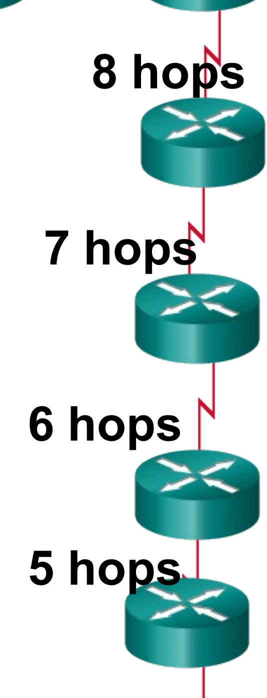



- Distance vector routing protocols:
  - Share updates between neighbors
  - Not aware of the network topology
- RIPv1 sends periodically broadcasts updates to IP 255.255.255.255 even if topology has not changed
  - Updates consume bandwidth and network device CPU resources
- EIGRP will only send an update when topology has changed
- RIPv2 and EIGRP use multicast addresses

# Routing Information Protocol ver. 1 (RIPv1)

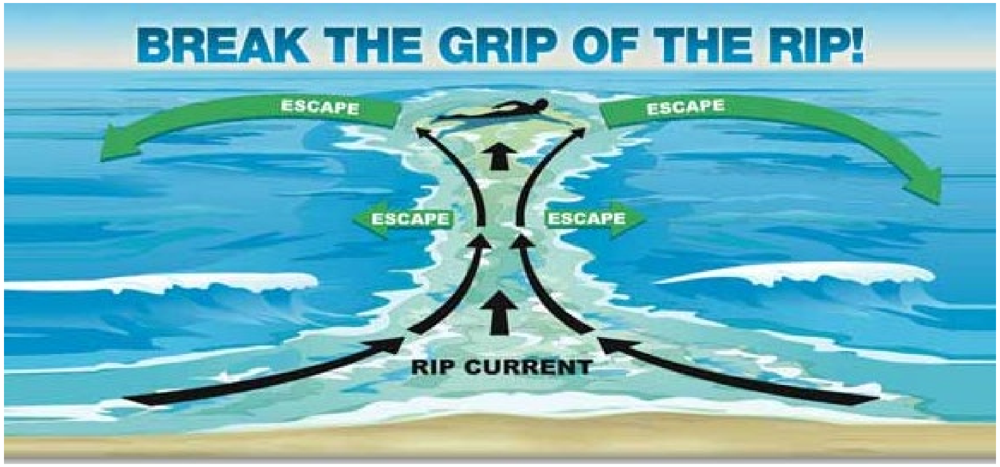
| | | | |
|---|---|---|---|
| Data Link Frame Header | IP Packet Header | UDP Segment Header | RIP Message (512 Bytes: Up to 25 Routes) |



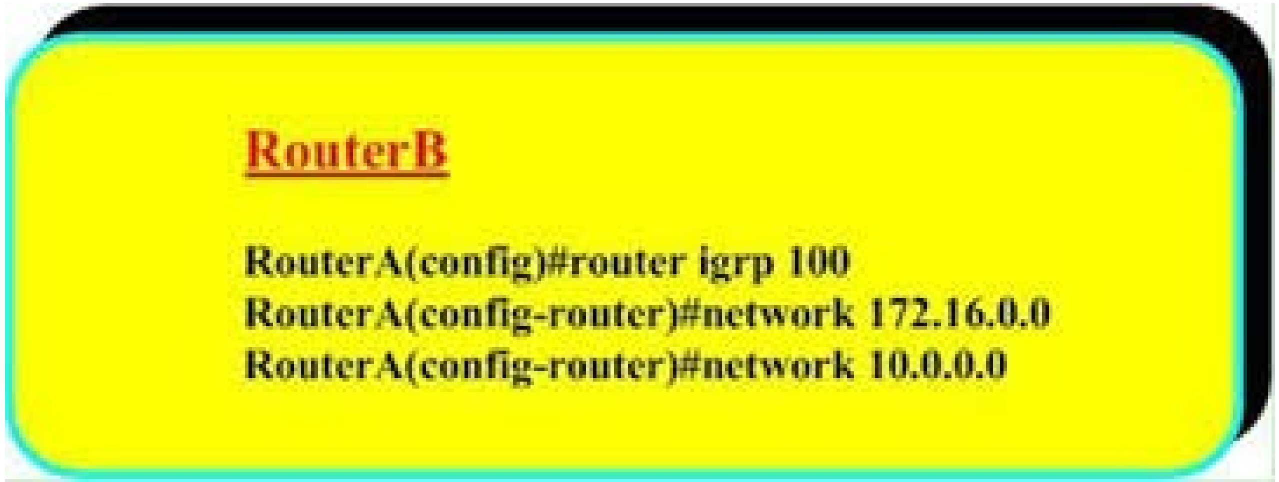

- RIPv1 (RFC 1058) is a legacy 1st generation IPv4 protocol.
  - Easy to configure, making it a good choice for small networks.
- RIPv1 has the following key characteristics:
  - **Metric** = **hop count** (lower is better).
  - **Updates** broadcasted **every 30 seconds** to 255.255.255.255.
  - **If hop count > 15 hops = too f**ar and the update is not propagated.
  - RIP updates are encapsulated into a **UDP segment**, with both source and destination port numbers set to UDP port 520.

# Routing Information Protocol ver. 2 (RIPv2)


- RIPv2 (RFC 1058) replaced RIPv1and included the following improvements:
  - **Classless routing protocol:** Supports VLSM and CIDR, because it includes the subnet mask in the routing updates.
  - **Increased efficiency:** Forwards updates to multicast address 224.0.0.9, instead of the broadcast address 255.255.255.255.
  - **Reduced routing entries:** Supports manual route summarization.
  - **Secure:** Supports an authentication mechanism to secure routing table updates between neighbors.

# IGRP



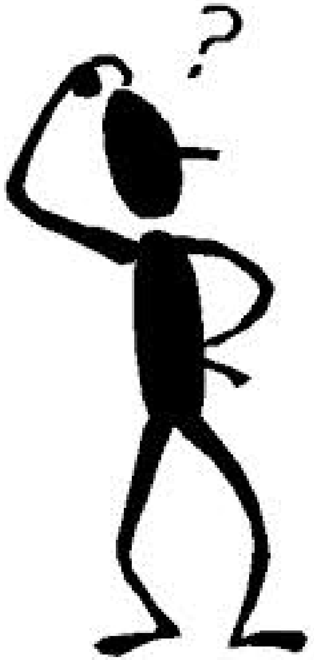

- The **Interior Gateway Routing Protocol (IGRP)** was the first proprietary IPv4 routing protocol developed by **Cisco in 1984**.
- Gave customers a choice between two standard protocols RIPv1 and OSPF.
  - No hop limit
  - Easy to understand
- Bandwidth and delay are used to create a composite metric.
  - Optionally, load and reliability can also be included in the calculation.
- Routing **updates are broadcast every 90 seconds**, by default.

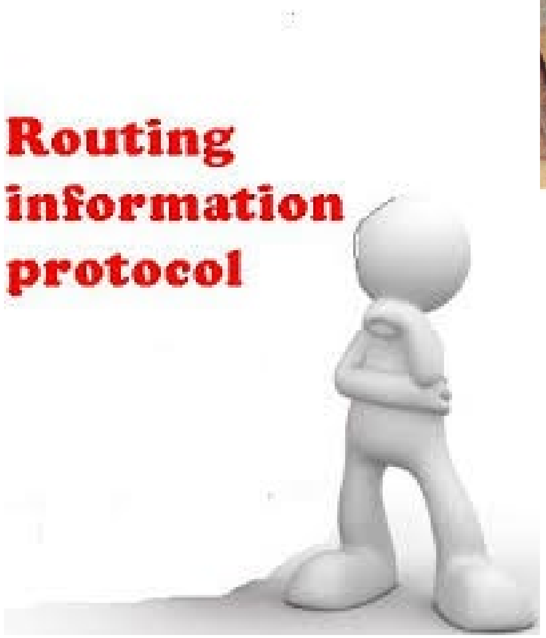
# EIGRP


- 1n 1992, **Enhanced IGRP** (EIGRP) replaced IGRP.
  - EIGRP also supports VLSM and CIDR, increases efficiency, reduces routing updates, and supports secure message exchange.
- EIGRP also introduced: (more later!)
  - Bounded triggered updates
  - Hello keepalive mechanism
  - Maintains a topology table (DUAL – backup routes)
  - Rapid convergence
  - Multiple network layer protocol support (IPv4, IPv6, IPX, Appletalk)

# Routing Information Protocol

# Why RIP?







- RIP is rarely used in modern networks!
- However, it's useful as a foundation for understanding basic network routing.
- This section provides a brief overview of how to configure basic RIP settings and to verify RIPv2.

# Overview

| | RIPv1 | RIPv2 |
|---|---|---|
| **Distance Vector or Link State?** | Distance Vector | Distance Vector |
| **Is it a Classless routing protocol?** | ✗ | ✓ |
| **Does it use triggered updates?** | ✓ | ✓ |
| **Does it use either split horizon or split horizon w/poison reverse?** | ✓ | ✓ |
| **Does it uses hold-down timers?** | ✓ | ✓ |
| **Is the maximum hop count = 15?** | ✓ | ✓ |
| **Does it do auto summary?** | ✓ | ✓ |
| **Does it support CIDR?** | ✗ | ✓ |
| **Does it support VLSM?** | ✗ | ✓ |
| **Does it support authentication?** | ✗ | ✓ |

# RIP Cheat Sheet

## RIP

packetlife.net

### RIP Implementations

**RIPv1**
Original RIP implementation, limited to classful routing (obsolete)

**RIPv2**
Introduced support for classless routing, authentication, triggered updates, and multicast announcements (RFC 2453)

**RIPng (RIP Next Generation)**
Extends RIPv2 to support IPv6 routing (RFC 2080); functions very similarly to RIPv2 and is subsequently as limited

### Protocols Comparison

| | RIPv1 | RIPv2 | RIPng |
|---|---|---|---|
| **IP** | IPv4 | IPv4 | IPv6 |
| **Admin Distance** | 120 | 120 | 120 |
| **UDP Port** | 520 | 520 | 521 |
| **Classless** | No | Yes | Yes |
| **Adv. Address** | Broadcast | 224.0.0.9 | FF02::9 |
| **Authentication** | None | Plain, MD5 | None |

### RIPv2 Configuration

```
! Enable RIPv2 IPv4 routing
router rip
 version 2

 ! Disable RIPv2 automatic summarization
 no auto-summary

 ! Designate RIPv2 interfaces by network
 network network

 ! Identify unicast-only neighbors
 neighbor IP-address

 ! Originate a default route
 default-information originate

 ! Designate passive interfaces
 passive-interface {interface | default}

 ! Modify timers
 timers basic update invalid hold flush
```

### RIPng Configuration

```
! Enable IPv6 routing
ipv6 unicast-routing

! Enable RIPng IPv6 routing
ipv6 router rip name

 ! Toggle split-horizon and poison-reverse
 [no] split-horizon
 [no] poison-reverse

 ! Modify timers
 timers basic update invalid hold flush
```

### Attributes

| | |
|---|---|
| **Type** | Distance Vector |
| **Algorithm** | Bellman-Ford |
| **Admin Distance** | 120 |
| **Metric** | Hop count (max 15) |
| **Standard** | RFCs 2080, 2453 |
| **Protocols** | IPv4, IPv6 |
| **Transport** | UDP |
| **Authentication** | Plaintext, MD5 |
| **Multicast IP** | 224.0.0.9/FF02::9 |

### Terminology

**Split Horizon**
A rule that states a router may not advertise a route back to the neighbor from which it was learned

**Route Poisoning**
When a network becomes unreachable, an update with an infinite metric is generated to explicitly advertise the route as unreachable

**Poison Reverse**
A router advertises a network as unreachable through the interface on which it was learned

### Timer Defaults

| | | | |
|---|---|---|---|
| **Update** | 30 sec | **Flush** | 240 sec |
| **Invalid** | 180 sec | **Hold-down** | 180 sec |

### RIPv2 Interface Configuration

```
! Configure manual route summarization
ip summary-address rip network mask

! Enable MD5 authentication (RIPv2 only)
ip rip authentication mode md5
ip rip authentication key-chain key-chain
```

### RIPng Interface Configuration

```
! Enable RIPng on the interface
ipv6 rip name enable

! Configure manual route summarization
ipv6 rip name summary-address prefix
```

### Troubleshooting

```
show ip[v6] protocols
show ip[v6] rip database
show ip[v6] route rip
debug ip rip { database | events }
debug ipv6 rip [interface]
```

by Jeremy Stretch

v1.1

# RIP

packetlife.net

## RIP Implementations

**RIPv1**

**RIPv2**

**RIPng (RIP Next Generation)**

## Protocols Comparison

| | RIPv1 | RIPv2 | RIPng |
|---|---|---|---|
| IP | | | |
| Admin Distance | | | |
| UDP Port | | | |
| Classless | | | |
| Adv. Address | | | |
| Authentication | | | |

## Attributes

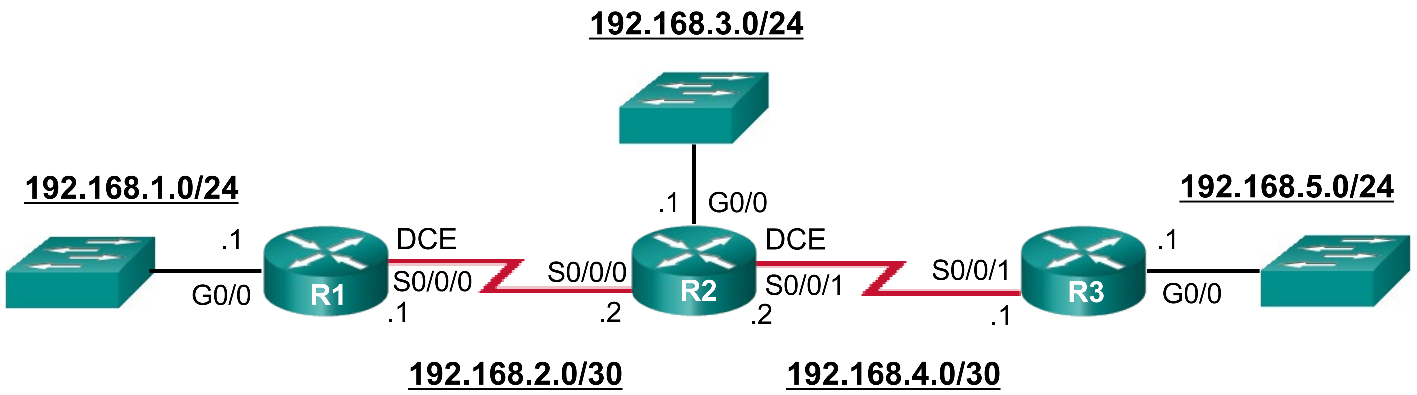
| | |
|---|---|
| Type | |
| Algorithm | |
| Admin Distance | |
| Metric | |
| Standard | |
| Protocols | |
| Transport | |
| Authentication | |
| Multicast IP | |

## Terminology

## Timer Defaults

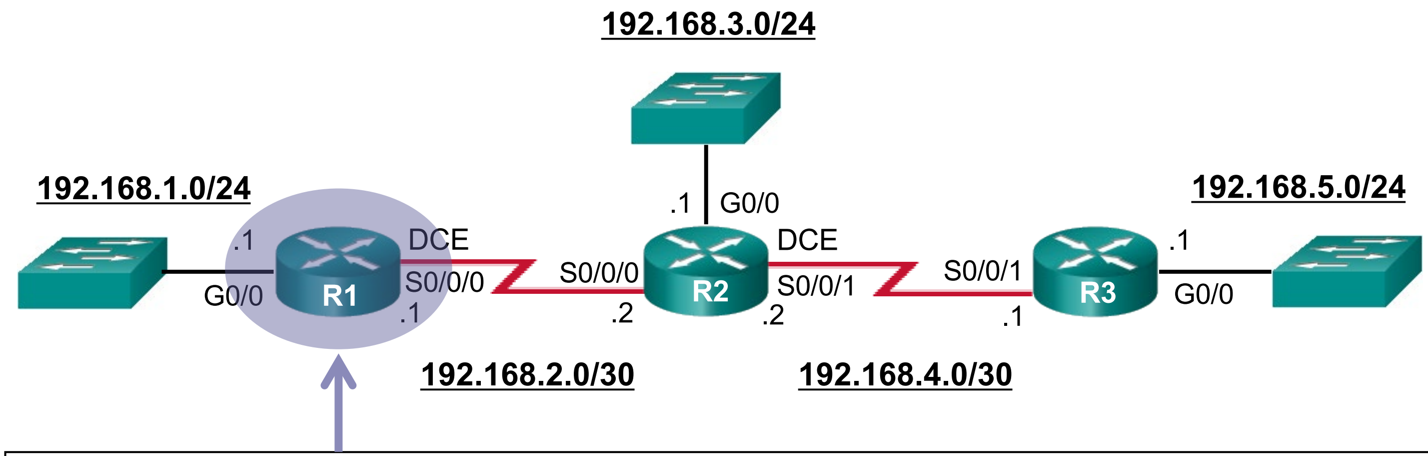
| | | | |
|---|---|---|---|
| Update | | Flush | |
| Invalid | | Hold-down | |

# RIP Reference Topology




| Device | Interface | IP Address | Subnet Mask |
|---|---|---|---|
| R1 | G0/0 | 192.168.1.1 | 255.255.255.0 |
| | S0/0/0 | 192.168.2.1 | 255.255.255.0 |
| R2 | G0/0 | 192.168.3.1 | 255.255.255.0 |
| | S0/0/0 | 192.168.2.2 | 255.255.255.0 |
| | S0/0/1 | 192.168.4.2 | 255.255.255.0 |
| R3 | G0/0 | 192.168.5.1 | 255.255.255.0 |
| | S0/0/1 | 192.168.4.1 | 255.255.255.0 |

# Entering Routing Configuration Mode

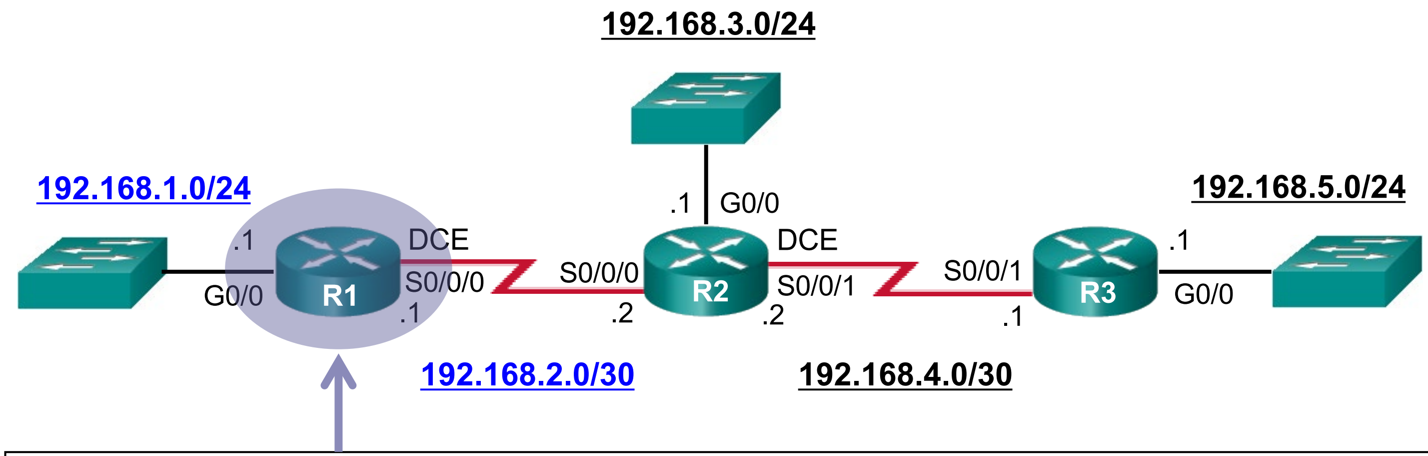



```
R1# conf t
Enter configuration commands, one per line.  End with
CNTL/Z.
R1(config)# router rip
R1(config-router)#
```

# RIP Configuration Options

```
R1(config-router)# ?
Router configuration commands:
  address-family            Enter Address Family command mode
  auto-summary              Enable automatic network number summarization
  default                   Set a command to its defaults
  default-information       Control distribution of default information
  default-metric            Set metric of redistributed routes
  distance                  Define an administrative distance
  distribute-list           Filter networks in routing updates
  exit                      Exit from routing protocol configuration mode
  flash-update-threshold    Specify flash update threshold in second
  help                      Description of the interactive help system
  input-queue               Specify input queue depth
  maximum-paths             Forward packets over multiple paths
  neighbor                  Specify a neighbor router
  network                   Enable routing on an IP network
  no                        Negate a command or set its defaults
  offset-list               Add or subtract offset from RIP metrics
  output-delay              Interpacket delay for RIP updates
  passive-interface         Suppress routing updates on an interface
  redistribute              Redistribute information from another routing
                            protocol
  timers                    Adjust routing timers
  traffic-share             How to compute traffic share over alternate paths
  validate-update-source    Perform sanity checks against source address of
                            routing updates
  version                   Set routing protocol version

R1(config-router)#
```

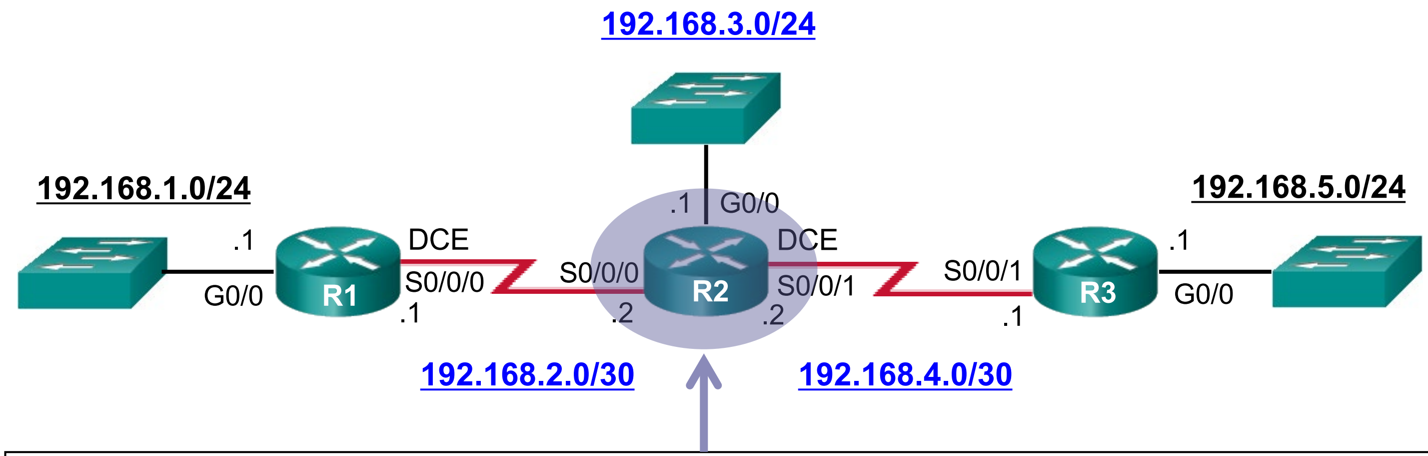
# Advertising the R1 Networks




```
R1(config)# router rip
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0
R1(config-router)#
```

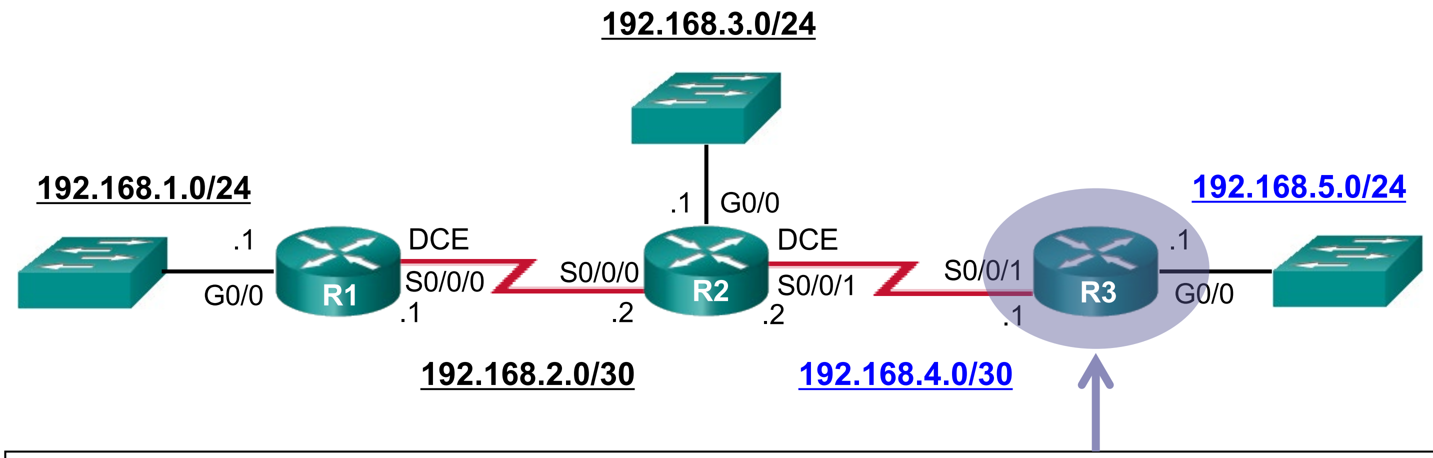
- To enable RIP and advertise a network, use the routing configuration command **`network`** *`network-address`*
  - Enter the network address for ***each directly connected network***.
- Entering the command automatically :
  - Enables RIP on all interfaces that belong to a specific network.
    - Interfaces now both send and receive RIP updates.
  - Advertises the specified network in RIP updates every 30 seconds.

# Advertising the R2 Networks




```
R2(config)# router rip
R2(config-router)# network 192.168.2.0
R2(config-router)# network 192.168.3.0
R2(config-router)# network 192.168.4.0
R2(config-router)#
```

# Advertising the R3 Networks

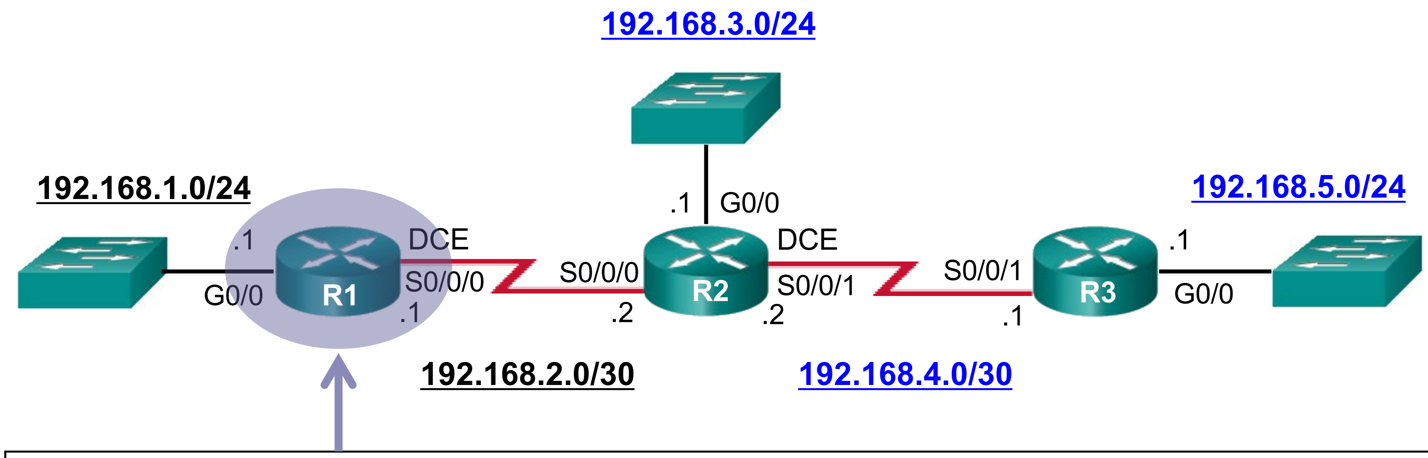



```
R3(config)# router rip
R3(config-router)# network 192.168.4.0
R3(config-router)# network 192.168.5.0
R3(config-router)#
```

# Verifying RIP Settings on R1

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

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 16 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface              Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0     1     1 2
    Serial0/0/0            1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.1.0
    192.168.2.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.2          120      00:00:15
  Distance: (default is 120)

R1#
```

# Verifying RIP Routes on R1

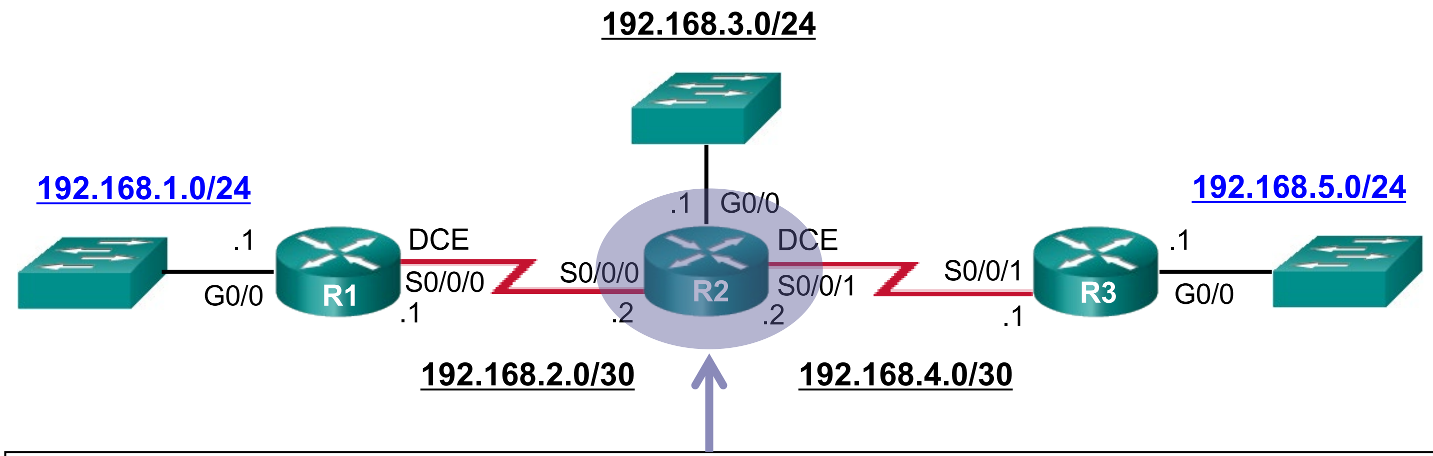



```
R1# show ip route | begin Gateway
Gateway of last resort is not set

      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
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Serial0/0/0
L        192.168.2.1/32 is directly connected, Serial0/0/0
R     192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
R     192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:24, Serial0/0/0
R     192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:24, Serial0/0/0
R1#
```

# Verifying RIP Settings on R2

```
R2# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 16 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    1     1 2
    Serial0/0/0           1     1 2
    Serial0/0/1           1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.2.0
    192.168.3.0
    192.168.4.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.1          120      00:00:13
    192.168.4.1          120      00:00:22
  Distance: (default is 120)
```

# Verifying RIP Routes on R2

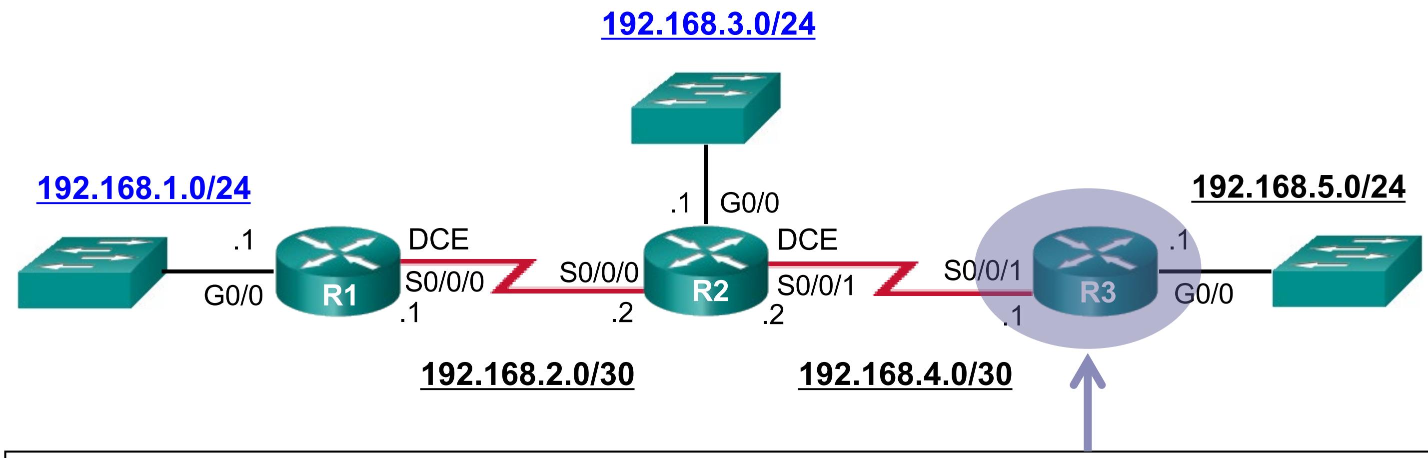



```
R2# show ip route | begin Gateway
Gateway of last resort is not set

R      192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:20, Serial0/0/0
       192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.2.0/24 is directly connected, Serial0/0/0
L         192.168.2.2/32 is directly connected, Serial0/0/0
       192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.3.0/24 is directly connected, GigabitEthernet0/0
L         192.168.3.1/32 is directly connected, GigabitEthernet0/0
       192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.4.0/24 is directly connected, Serial0/0/1
L         192.168.4.2/32 is directly connected, Serial0/0/1
R      192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:01, Serial0/0/1
R2#
```

# Verifying RIP Settings on R3

```
R3# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 7 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    1     1 2
    Serial0/0/1           1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.4.0
    192.168.5.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.4.2          120      00:00:27
  Distance: (default is 120)
```

# Verifying RIP Routes on R3



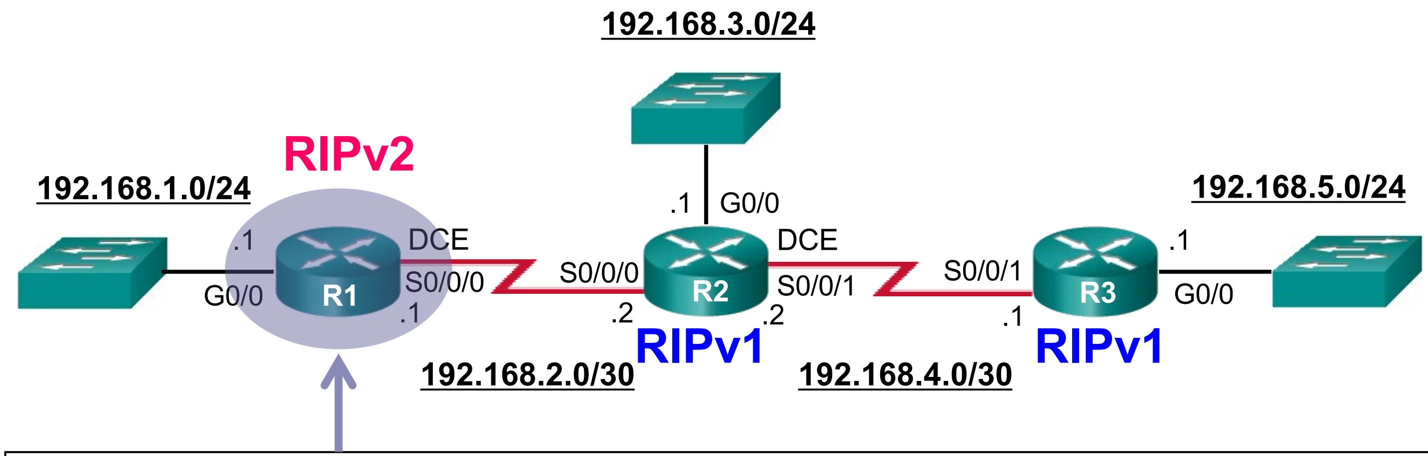

```
R3# show ip route | begin Gateway
Gateway of last resort is not set

R      192.168.1.0/24 [120/2] via 192.168.4.2, 00:00:02, Serial0/0/1
R      192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:02, Serial0/0/1
R      192.168.3.0/24 [120/1] via 192.168.4.2, 00:00:02, Serial0/0/1
       192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.4.0/24 is directly connected, Serial0/0/1
L         192.168.4.1/32 is directly connected, Serial0/0/1
       192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.5.0/24 is directly connected, GigabitEthernet0/0
L         192.168.5.1/32 is directly connected, GigabitEthernet0/0
R3#
```

# RIP Versions

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

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 16 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    1     1 2
    Serial0/0/0           1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    192.168.1.0
    192.168.2.0

<Output omitted>
```

- By default, RIPv1 is enabled when entering **router rip**
- Notice how R1 can send RIPv1 messages, but it can receive both RIPv1 and RIPv2 messages.
  - R1 would ignore the RIPv2 fields in the route entry.

# Enable and Verify RIPv2 on R1

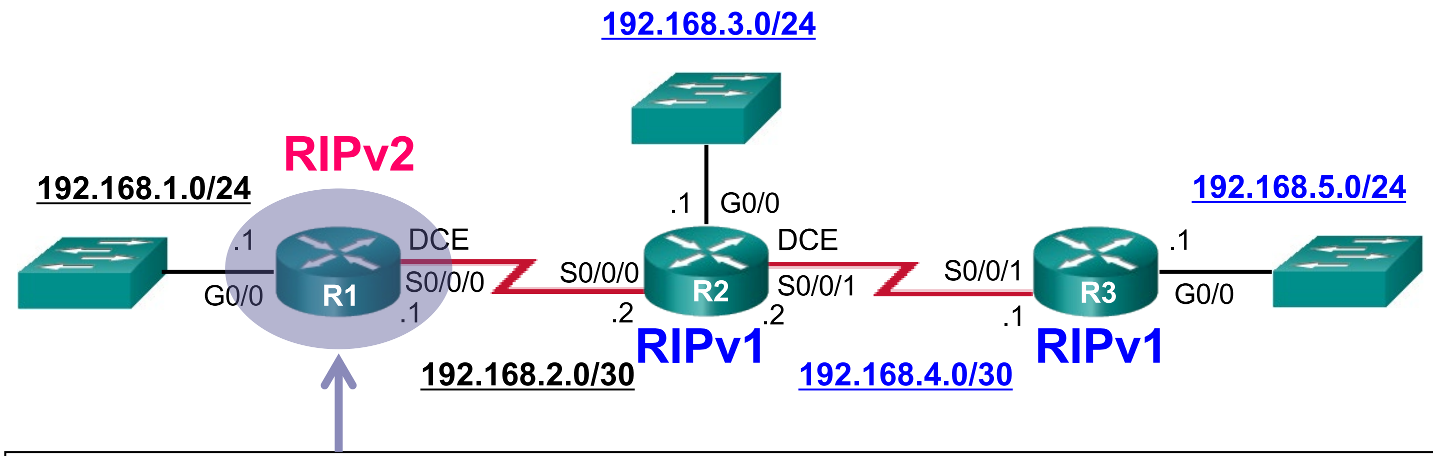



```
R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# ^Z
R1#
R1# show ip protocols | section Default
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    2     2
    Serial0/0/0           2     2
R1#
```

**Note**:

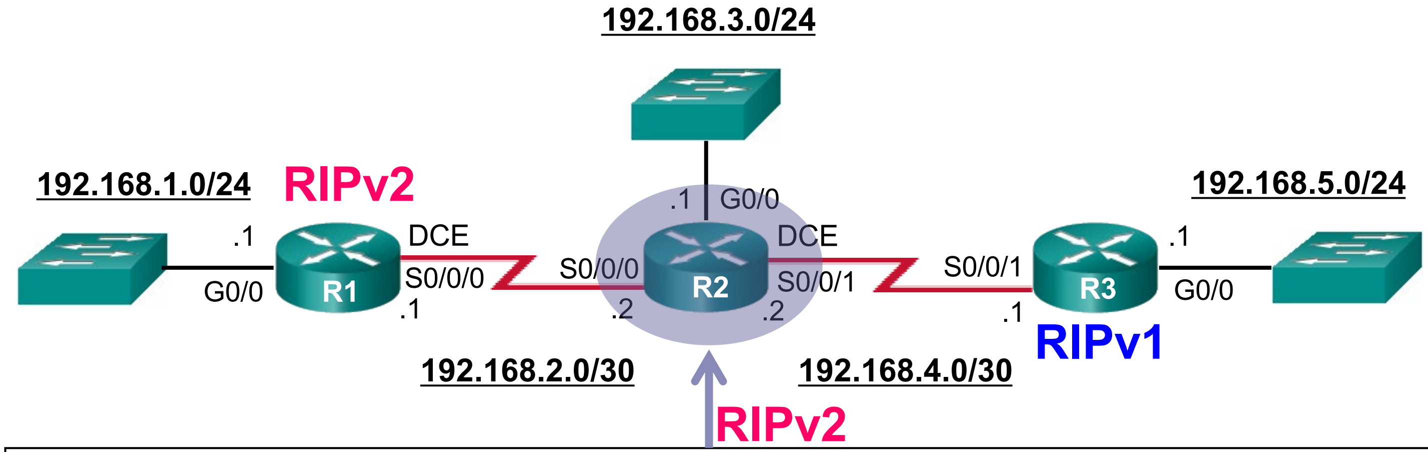
- Configuring **version 1** enables RIPv1 only
- Configuring **no version** returns the router to the default settings.

# Verifying RIP Routes on R1



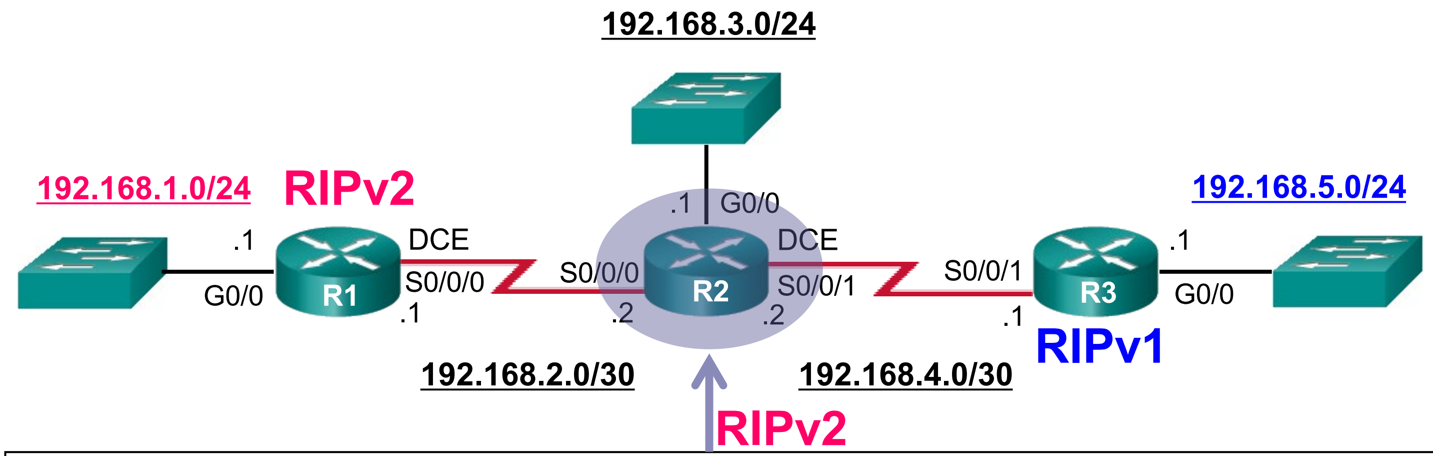

```
R1# show ip route | begin Gateway
Gateway of last resort is not set

      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
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Serial0/0/0
L        192.168.2.1/32 is directly connected, Serial0/0/0
R1#
```

# Enable and Verify RIPv2 on R2



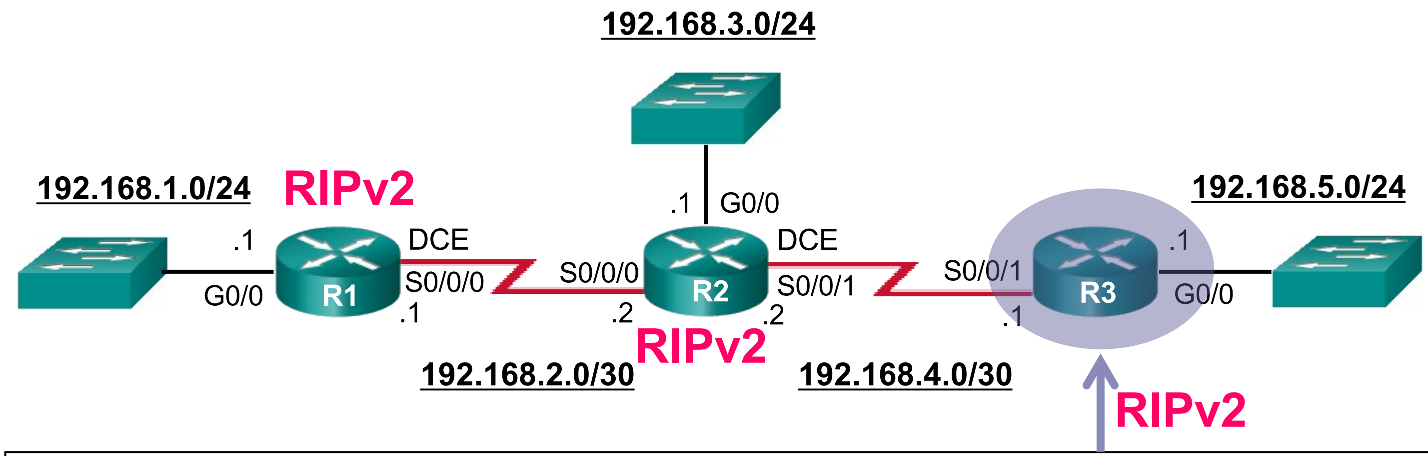

```
R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# end
R2#
R2# show ip protocols | section Default
Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    2     2
    Serial0/0/0           2     2
    Serial0/0/1           2     2
R2#
```

# Verifying RIP Routes on R2



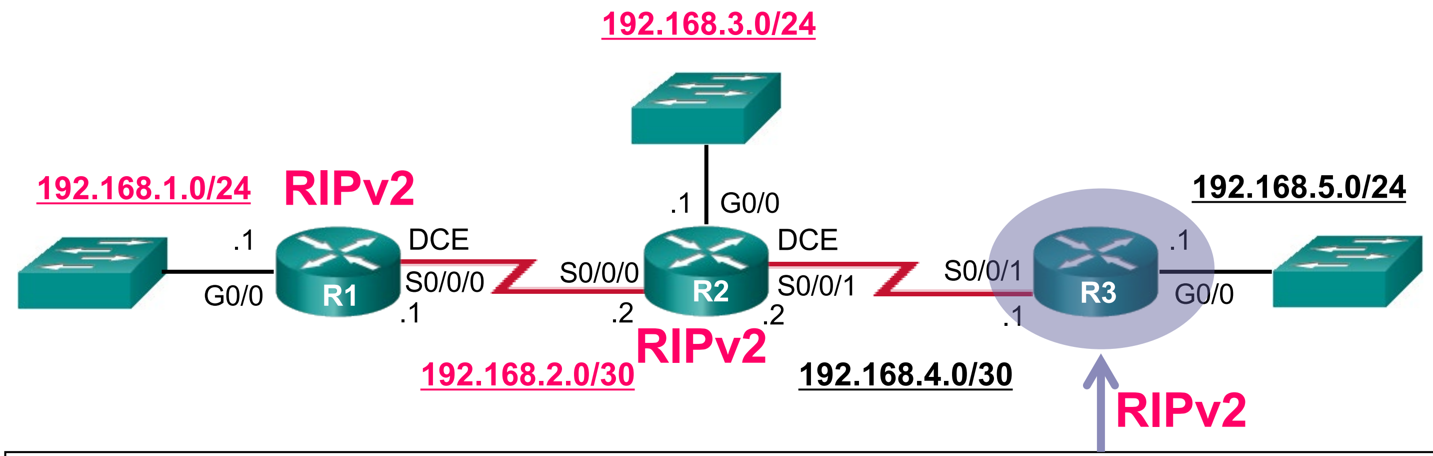

```
R2# show ip route | begin Gateway
Gateway of last resort is not set

R      192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:03, Serial0/0/0
       192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.2.0/24 is directly connected, Serial0/0/0
L         192.168.2.2/32 is directly connected, Serial0/0/0
       192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.3.0/24 is directly connected, GigabitEthernet0/0
L         192.168.3.1/32 is directly connected, GigabitEthernet0/0
       192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.4.0/24 is directly connected, Serial0/0/1
L         192.168.4.2/32 is directly connected, Serial0/0/1
R2#
```

# Enable and Verify RIPv2 on R3



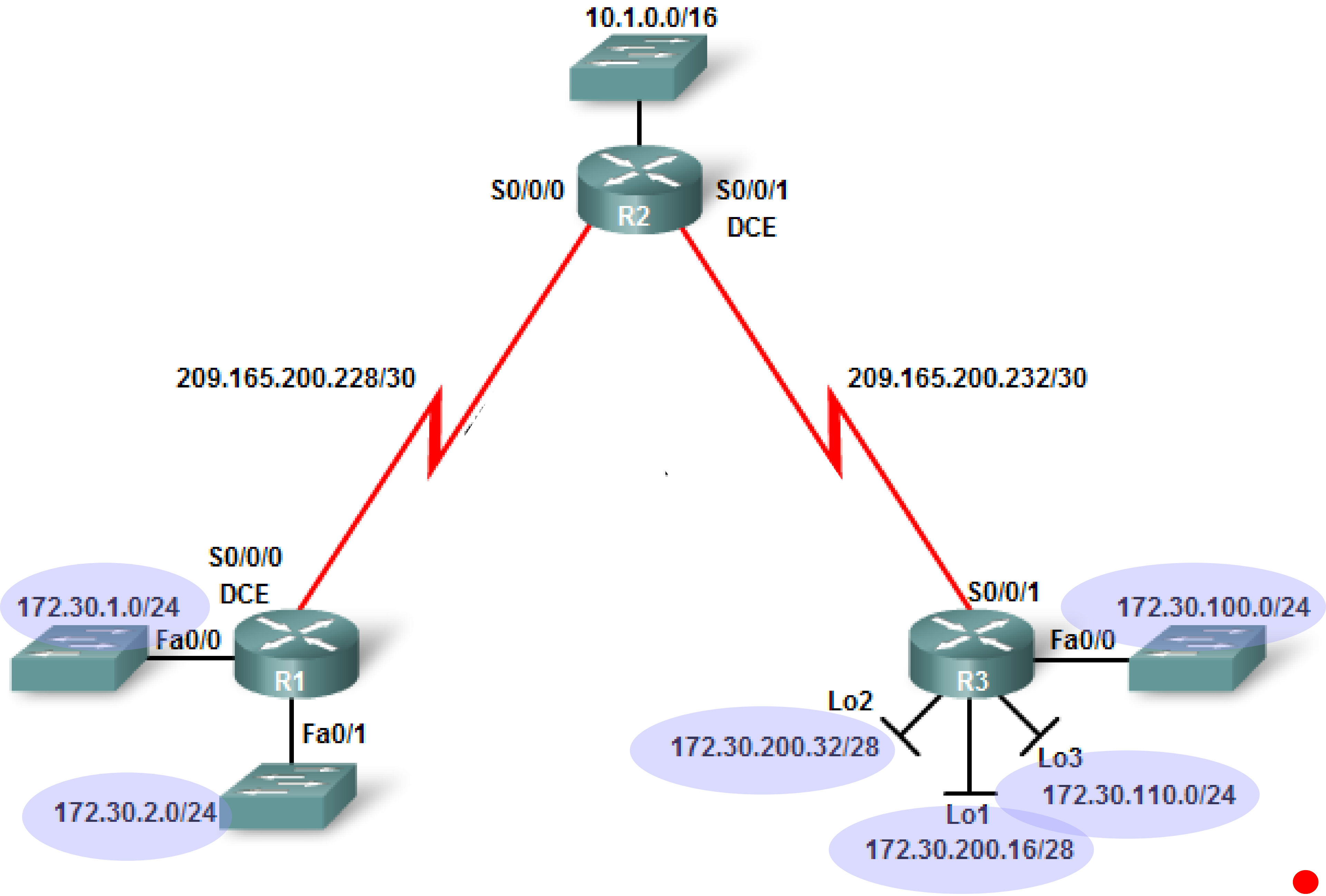

```
R3(config)# router rip
R3(config-router)# version 2
R3(config-router)# end
R3#
R3# show ip protocols | section Default
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    2     2
    Serial0/0/1           2     2
R3#
```

# Verifying RIP Routes on R3

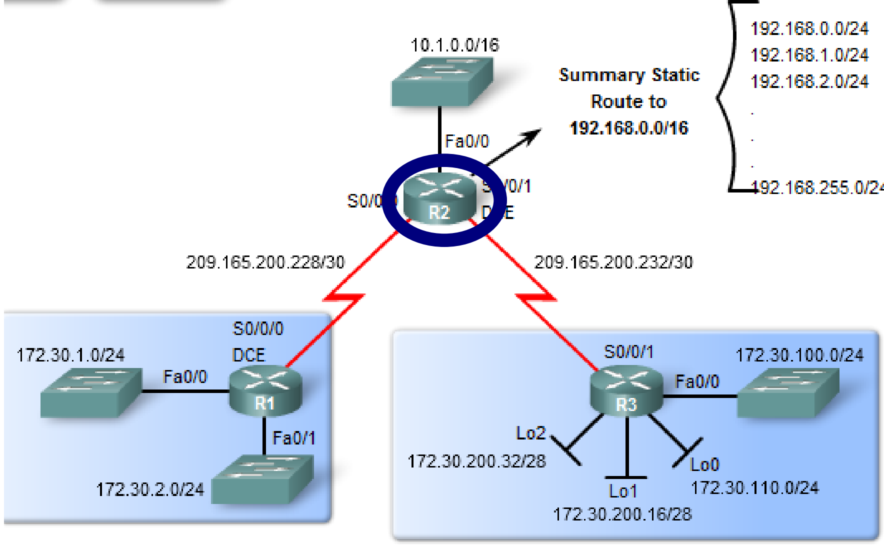



```
R3# show ip route | begin Gateway
Gateway of last resort is not set

R     192.168.1.0/24 [120/2] via 192.168.4.2, 00:00:02, Serial0/0/1
R     192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:02, Serial0/0/1
R     192.168.3.0/24 [120/1] via 192.168.4.2, 00:00:02, Serial0/0/1
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, Serial0/0/1
L        192.168.4.1/32 is directly connected, Serial0/0/1
      192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.5.0/24 is directly connected, GigabitEthernet0/0
L        192.168.5.1/32 is directly connected, GigabitEthernet0/0
R3#
```

# Automatic Summarization



# Auto-Summary and RIPv2

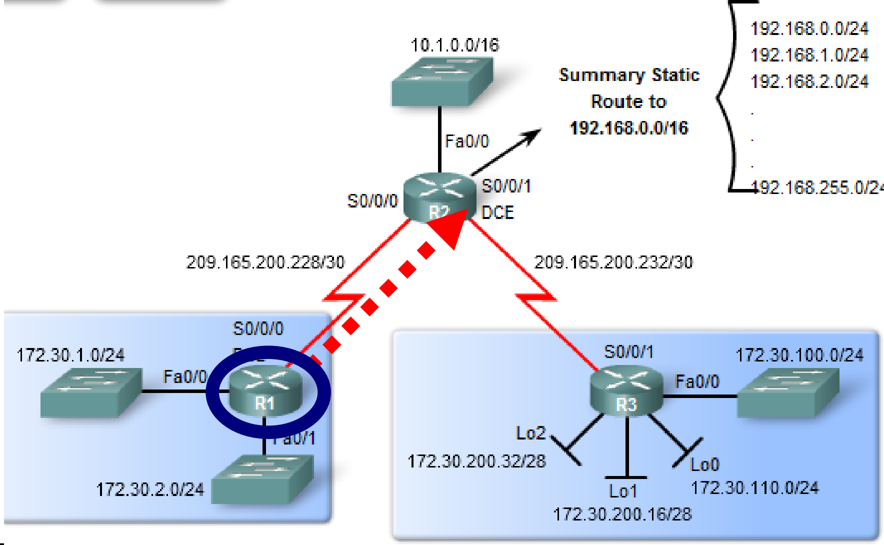



What do you expect to see?

```
R2# show ip route

R     172.30.0.0/16 [120/1] via 209.165.200.230, 00:00:28, Serial0/0/0
                    [120/1] via 209.165.200.234, 00:00:18, Serial0/0/1
      209.165.200.0/30 is subnetted, 2 subnets
C        209.165.200.232 is directly connected, Serial0/0/1
C        209.165.200.228 is directly connected, Serial0/0/0
      10.0.0.0/16 is subnetted, 1 subnets
C        10.1.0.0 is directly connected, FastEthernet0/0
S     192.168.0.0/16 is directly connected, Null0
```

- You still see the **summarized 172.30.0.0/16 route** with the same **two equal-cost paths**.

# Auto-Summary and RIPv2




```
R1# show ip protocols
Routing Protocol is "rip"
<output omitted>
  Default version control: send version 2, receive version 2
    Interface             Send     Recv     Triggered RIP Key-chain
    FastEthernet0/0       2        2
    FastEthernet0/1       2        2
    Serial0/1/0           2        2
  Automatic network summarization is in effect
```

- ***By default, RIPv2 automatically summarizes networks at major network boundaries, just like RIPv1.***
- Both R1 and R3 routers are still summarizing their 172.30.0.0 subnets

# Disabling Auto-summary in RIPv2

```
R2(config)# router rip
R2(config-router)# no auto-summary
```

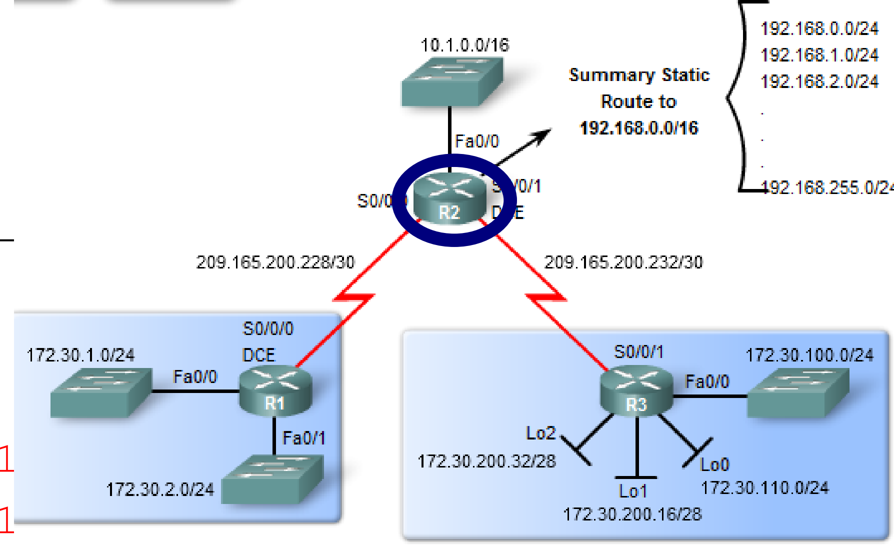
```
R3(config)# router rip
R3(config-router)# no auto-summary
```

```
R1(config)# router rip
R1(config-router)# no auto-summary
```

```
R1# show ip protocols
<output omitted>
  Automatic network summarization is not in effect
<output omitted>
```

- To modify the default RIPv2 behavior of automatic summarization, use the **no auto-summary** command
- RIPv2 no longer summarizes networks to their classful address at boundary routers but instead includes all subnets and their appropriate masks in its routing updates.

# Verifying RIPv2 Updates

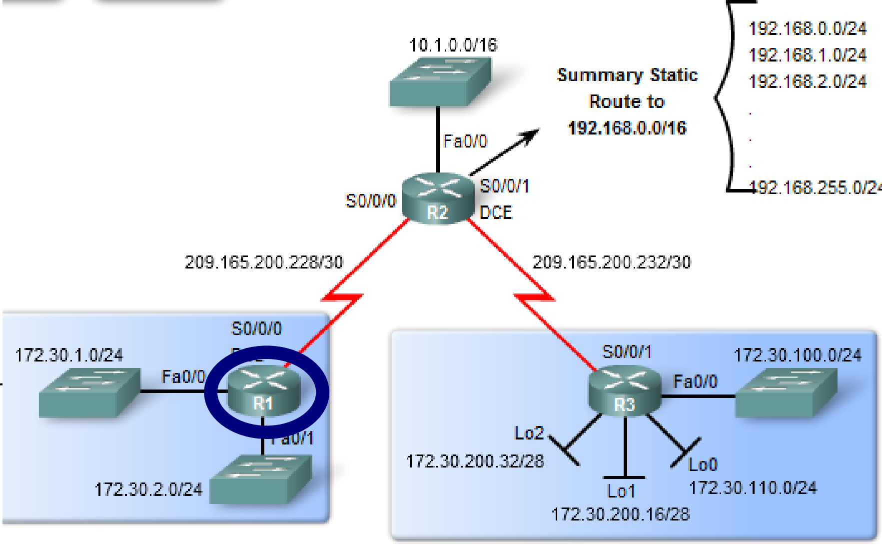



## What do you expect to see?

```
R2# show ip route

   172.30.0.0/16 is variably subnetted,
R     172.30.200.32/28 [120/1] via 209.1
R     172.30.200.16/28 [120/1] via 209.1
R     172.30.2.0/24 [120/1] via 209.165.200.230, 00:00:03, Serial0/0/0
R     172.30.1.0/24 [120/1] via 209.165.200.230, 00:00:03, Serial0/0/0
R     172.30.100.0/24 [120/1] via 209.165.200.234, 00:00:09, Serial0/0/1
R     172.30.110.0/24 [120/1] via 209.165.200.234, 00:00:09, Serial0/0/1
   209.165.200.0/30 is subnetted, 2 subnets
C     209.165.200.232 is directly connected, Serial0/0/1
C     209.165.200.228 is directly connected, Serial0/0/0
   10.0.0.0/16 is subnetted, 1 subnets
C     10.1.0.0 is directly connected, FastEthernet0/0
S  192.168.0.0/16 is directly connected, Null0
```

- The routing table for **R2 now contains the individual subnets for 172.30.0.0/16.**
- Notice that a **single summary route with two equal-cost paths no longer exists**.
- Each subnet and mask has its own specific entry, along with the exit interface and next-hop address to reach that subnet.

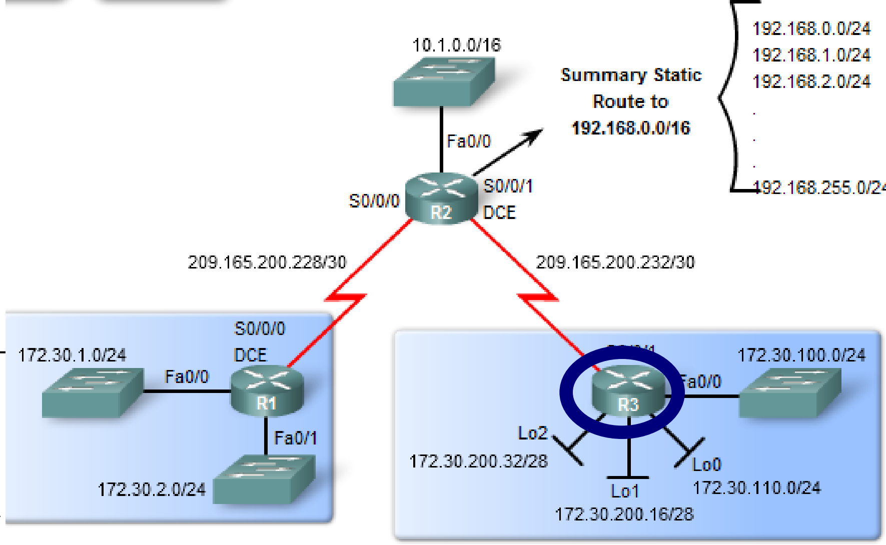
# Verifying RIPv2 Updates




```
R1# show ip route

   172.30.0.0/16 is variably subnetted,
R      172.30.200.32/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R      172.30.200.16/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
C      172.30.2.0/24 is directly connected, Loopback0
C      172.30.1.0/24 is directly connected, FastEthernet0/0
R      172.30.100.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R      172.30.110.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
   209.165.200.0/30 is subnetted, 2 subnets
R      209.165.200.232 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
C      209.165.200.228 is directly connected, Serial0/0/0
   10.0.0.0/16 is subnetted, 1 subnets
R     10.1.0.0 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
R     192.168.0.0/16 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
```

- Fully converged routing tables.

# Verifying RIPv2 Updates

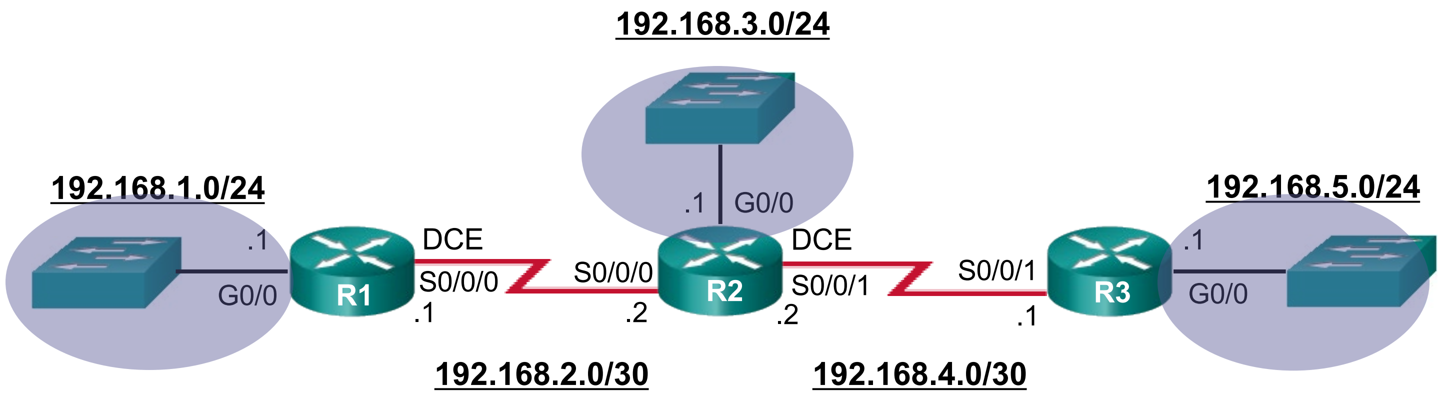



```
R3# show ip route

    172.30.0.0/16 is variably subnetted,
C       172.30.200.32/28 is directly connected, Loopback2
C       172.30.200.16/28 is directly connected, Loopback1
R       172.30.2.0/24 [120/2] via 209.165.200.233, 00:00:01, Serial0/0/1
R       172.30.1.0/24 [120/2] via 209.165.200.233, 00:00:01, Serial0/0/1
C       172.30.100.0/24 is directly connected, FastEthernet0/0
C       172.30.110.0/24 is directly connected, Loopback0
    209.165.200.0/30 is subnetted, 2 subnets
C       209.165.200.232 is directly connected, Serial0/0/1
R       209.165.200.228 [120/1] via 209.165.200.233, 00:00:02, Serial0/0/1
    10.0.0.0/16 is subnetted, 1 subnets
R       10.1.0.0 [120/1] via 209.165.200.233, 00:00:02, Serial0/0/1
R   192.168.0.0/16 [120/1] via 209.165.200.233, 00:00:02, Serial0/0/1
```



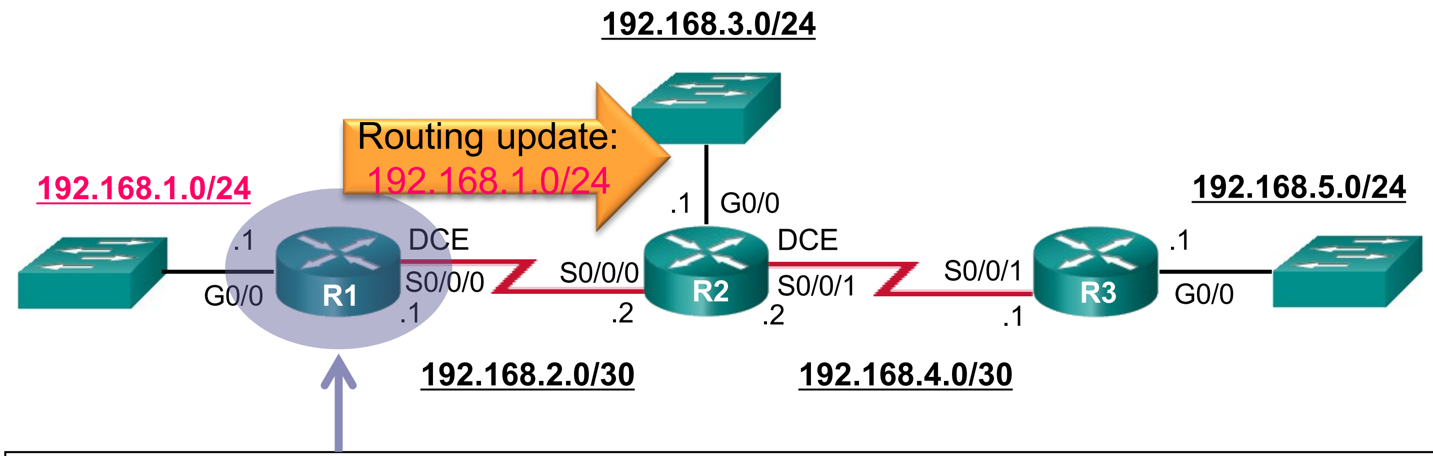
- Fully converged routing tables.

# RIP Passive Interfaces




- Sending out unneeded updates on a LAN:
  - Wastes Bandwidth
  - Wastes Resources
  - Security Risk
- The **`passive-interface`**
  - Stops routing updates out the specified interface.
  - The **network** that the specified interface belongs to is still advertised in routing updates that are sent out other interfaces.
  - Should be configured on interfaces which do not connect to other RIP routers.

# Configuring a Passive Interface on R1




```
R1(config)# router rip
R1(config-router)# passive-interface g0/0
R1(config-router)# end
R1#
R1# show ip protocols | begin Default
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Serial0/0/0           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.1.0
    192.168.2.0
  Passive Interface(s):
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.2          120      00:00:06
  Distance: (default is 120)
```

# Configuring a Passive Interface on R2

```
R2(config)# router rip
R2(config-router)# passive-interface g0/0
R2(config-router)# end
R2#
*Mar 10 16:33:32.391: %SYS-5-CONFIG_I: Configured from console by
console
R2# show ip protocols | begin Default
Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Serial0/0/0           2     2
    Serial0/0/1           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.2.0
    192.168.3.0
    192.168.4.0
  Passive Interface(s):
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.2.1          120      00:00:24
    Gateway         Distance      Last Update
    192.168.4.1          120      00:00:23
  Distance: (default is 120)

R2#
```

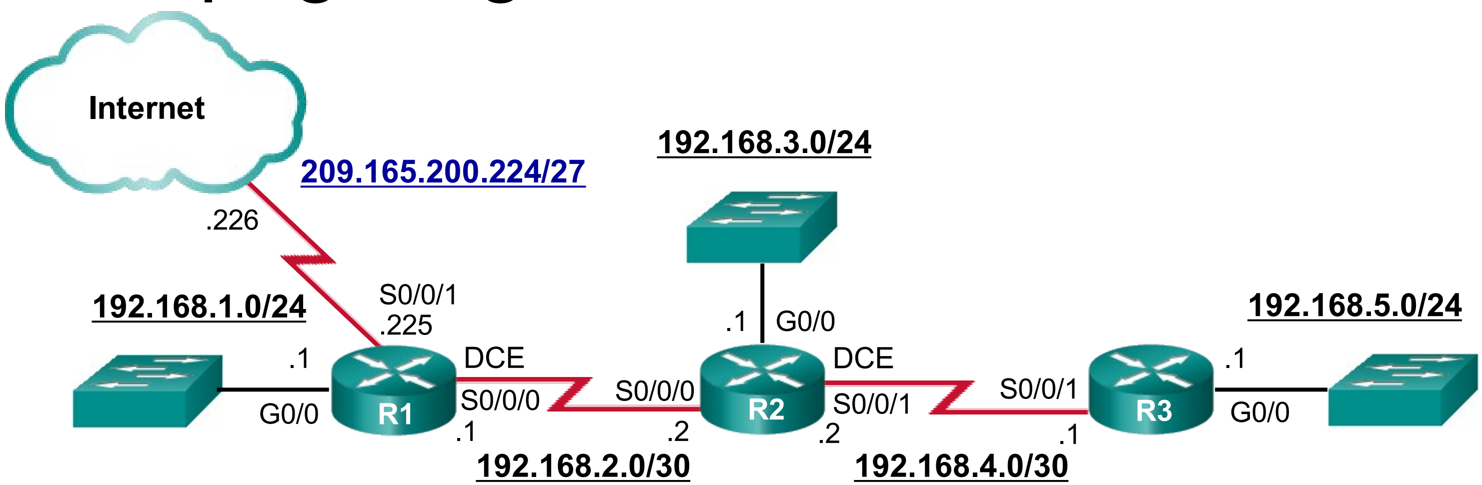
# Configuring a Passive Interface on R3

```
R3(config)# router rip
R3(config-router)# passive-interface default
R3(config-router)# no passive-interface s0/0/1
R3(config-router)# end
R3#
*Mar 10 16:34:28.899: %SYS-5-CONFIG_I: Configured
console
R3# show ip protocols | begin Default
Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    Serial0/0/1            2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.4.0
    192.168.5.0
  Passive Interface(s):
    Embedded-Service-Engine0/0
    GigabitEthernet0/0
    GigabitEthernet0/1
    GigabitEthernet0/3
    Serial0/0/0
Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.4.2          120      00:00:23
  Distance: (default is 120)

R3#
```

NOTE:

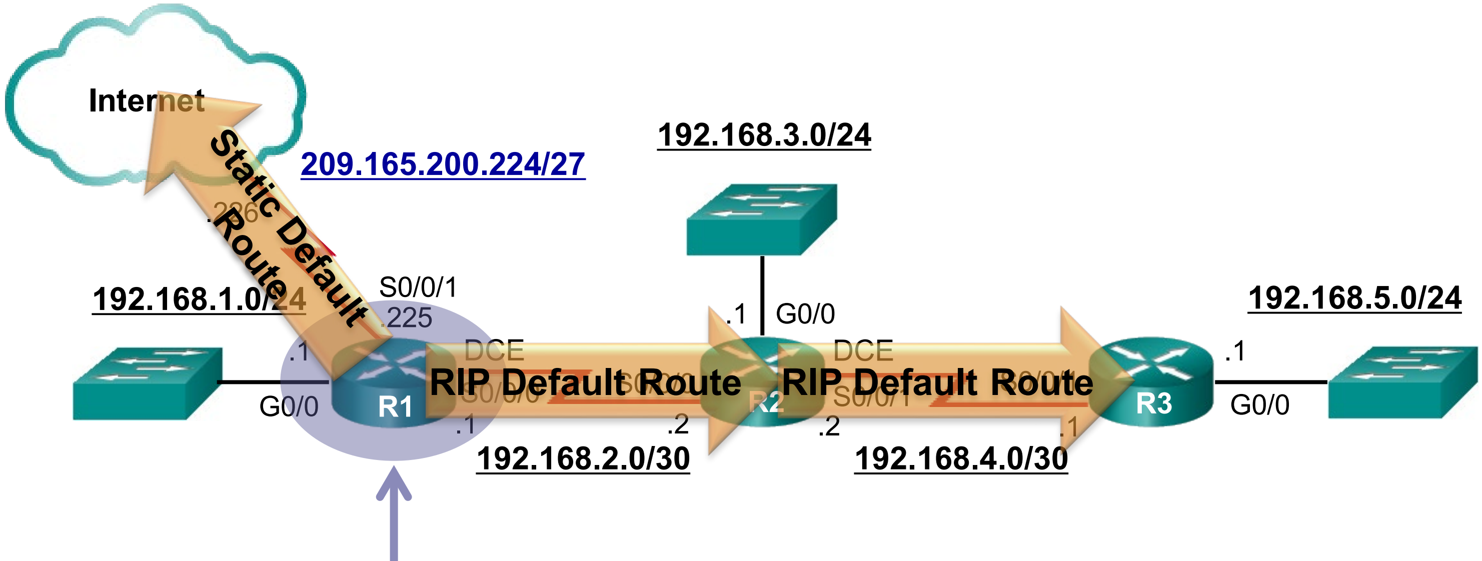
- As an alternative, all interfaces can be made passive using the **passive-interface default** command.
- Interfaces that should not be passive can be re-enabled using the **no passive-interface** command.

# RIP Propagating a Default Static Route




- It is common to configure a default static route on an edge router and then propagating the default route throughout the routing domain using the routing protocol.
  - Otherwise, you would have to individually configure default static routes on all internal routers.
- Edge router must be configured with default static route:
  - **`ip route 0.0.0.0 0.0.0.0`** *`exit-intf next-hop-ip`*
- Propagated to other routers via RIP using:
  - **`default-information originate`**

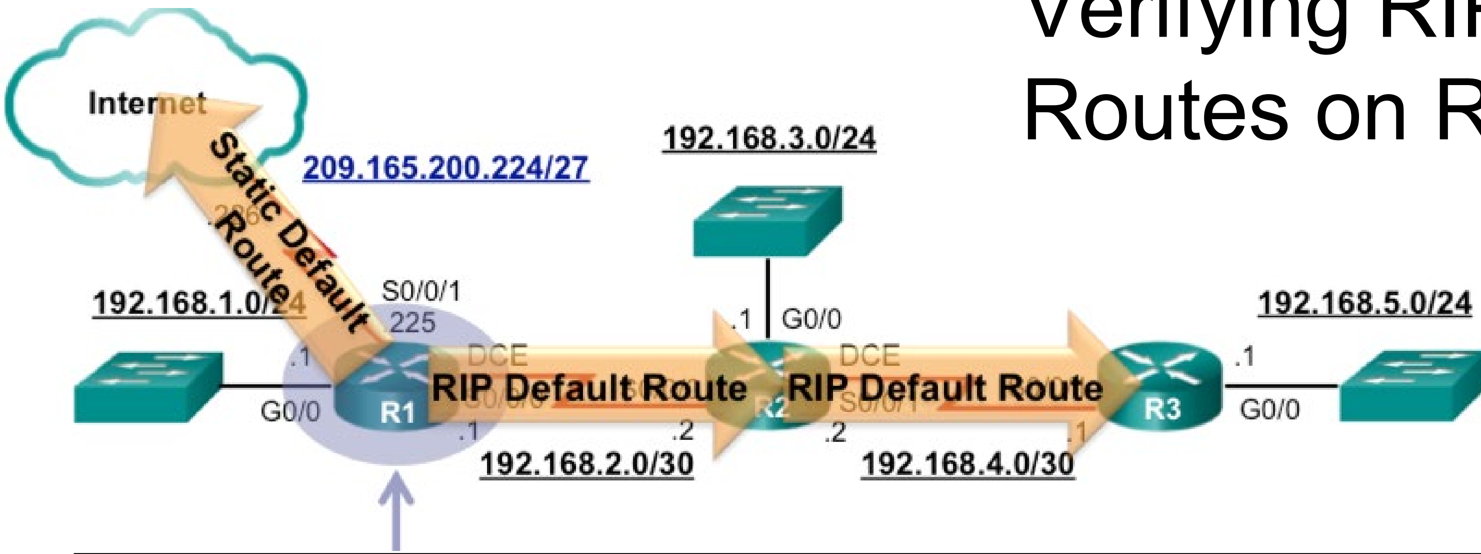
# Propagating a Default Route on R1




```
R1(config)# ip route 0.0.0.0 0.0.0.0 S0/0/1 209.165.200.226
R1(config)# router rip
R1(config-router)# default-information originate
R1(config-router)# ^Z
R1#
*Mar 10 23:33:51.801: %SYS-5-CONFIG_I: Configured from
console by console
```

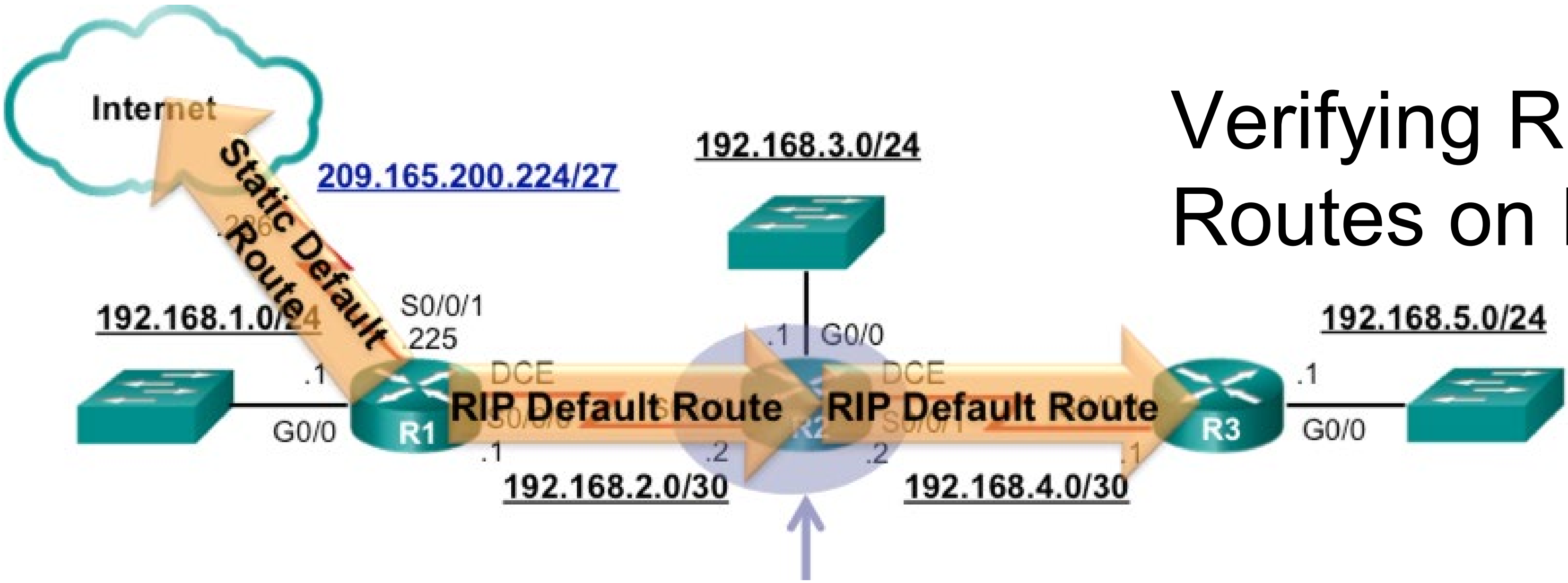
Note: Best to use only a next-hop route when CEF is enabled.

# Verifying RIP Routes on R1



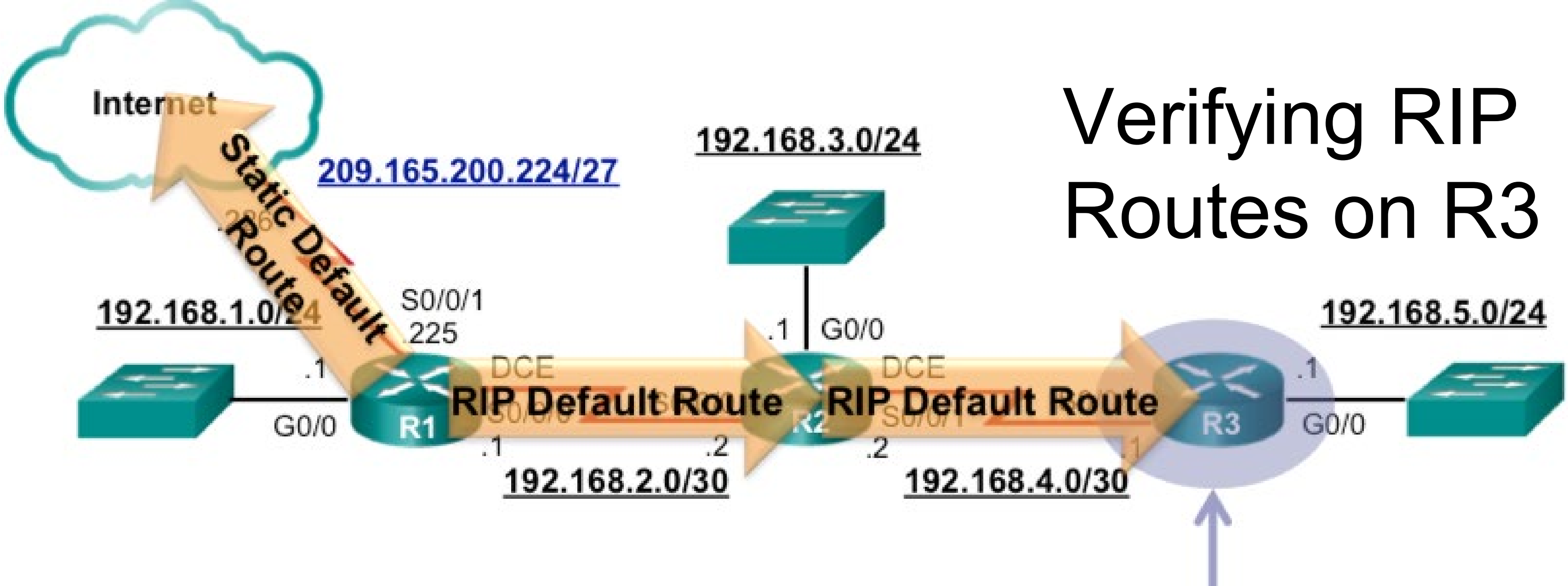

```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.226 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.226, Serial0/0/1
      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
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Serial0/0/0
L        192.168.2.1/32 is directly connected, Serial0/0/0
R     192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0
R     192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0
R     192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:08, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        209.165.200.0/24 is directly connected, Serial0/0/1
L        209.165.200.225/27 is directly connected, Serial0/0/1
R1#
```

# Verifying RIP Routes on R2




```
R2# show ip route | begin Gateway
Gateway of last resort is 192.168.2.1 to network 0.0.0.0

R*     0.0.0.0/0 [120/1] via 192.168.2.1, 00:00:21, Serial0/0/0
R      192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:25, Serial0/0/0
       192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.2.0/24 is directly connected, Serial0/0/0
L         192.168.2.2/32 is directly connected, Serial0/0/0
       192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.3.0/24 is directly connected, GigabitEthernet0/0
L         192.168.3.1/32 is directly connected, GigabitEthernet0/0
       192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.4.0/24 is directly connected, Serial0/0/1
L         192.168.4.2/32 is directly connected, Serial0/0/1
R      192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:15, Serial0/0/1
R2#
```

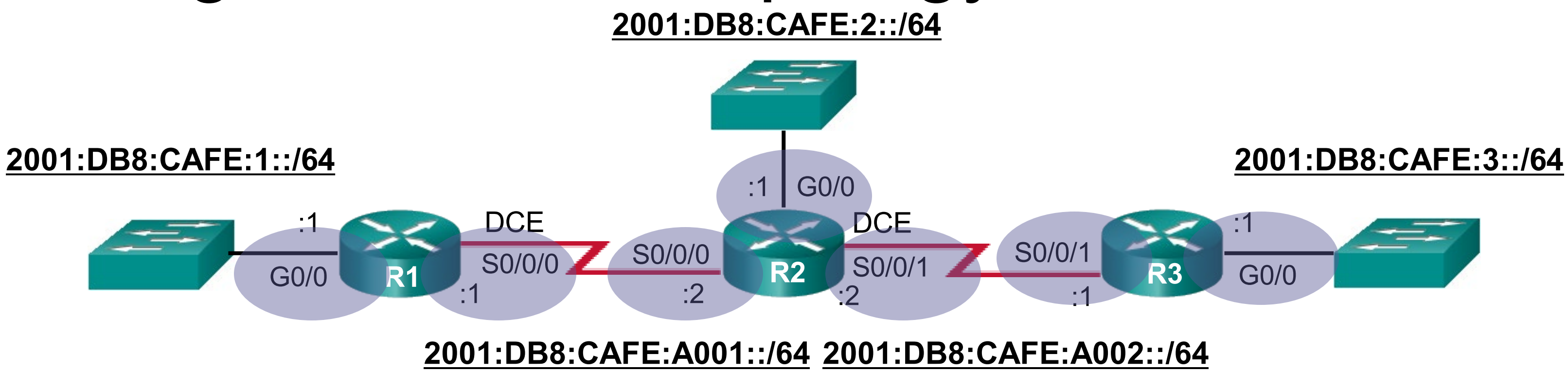
# Verifying RIP Routes on R3




```
R3# show ip route | begin Gateway
Gateway of last resort is 192.168.4.2 to network 0.0.0.0

R*     0.0.0.0/0 [120/2] via 192.168.4.2, 00:00:00, Serial0/0/1
R      192.168.1.0/24 [120/2] via 192.168.4.2, 00:00:00, Serial0/0/1
R      192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:00, Serial0/0/1
R      192.168.3.0/24 [120/1] via 192.168.4.2, 00:00:00, Serial0/0/1
       192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.4.0/24 is directly connected, Serial0/0/1
L         192.168.4.1/32 is directly connected, Serial0/0/1
       192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.5.0/24 is directly connected, GigabitEthernet0/0
L         192.168.5.1/32 is directly connected, GigabitEthernet0/0
R3#
```

# Configuring RIPng

# RIPng Reference Topology

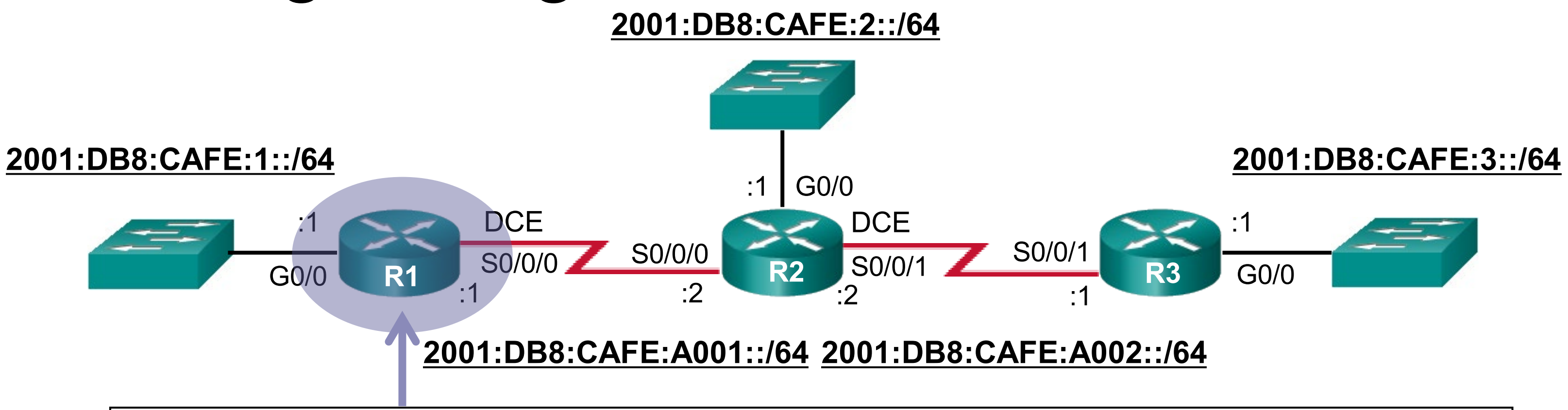



```
R2(config)# router rip
R2(config-router)# network 192.168.2.0
```

- No **network** command with **router rip**
- interface configuration command:

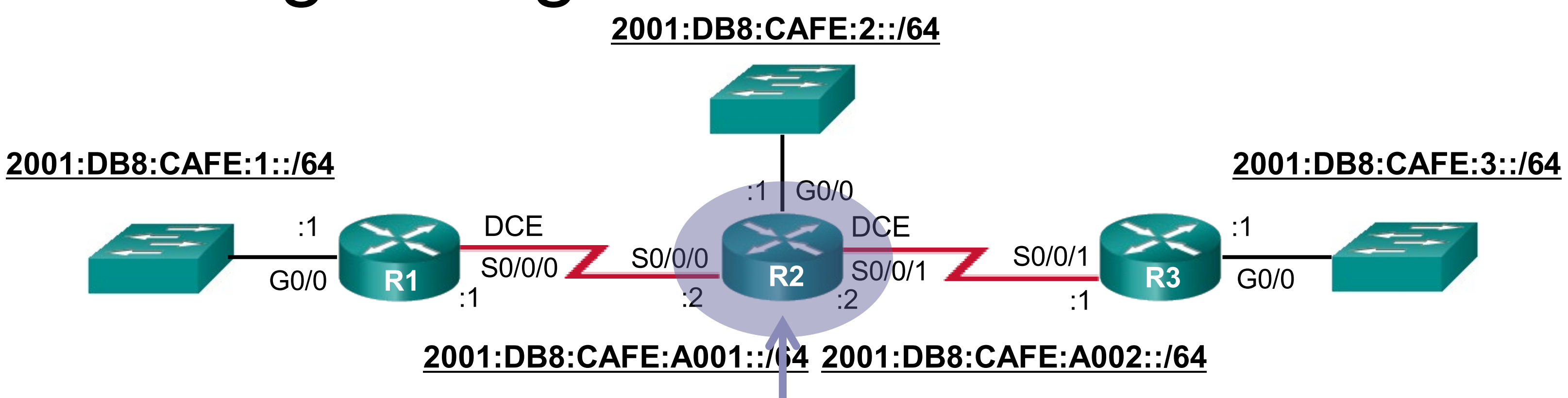
  `Router(config-if)#` **`ipv6 rip`** *`domain-name`* **`enable`**

# Enabling RIPng on the R1 Interfaces



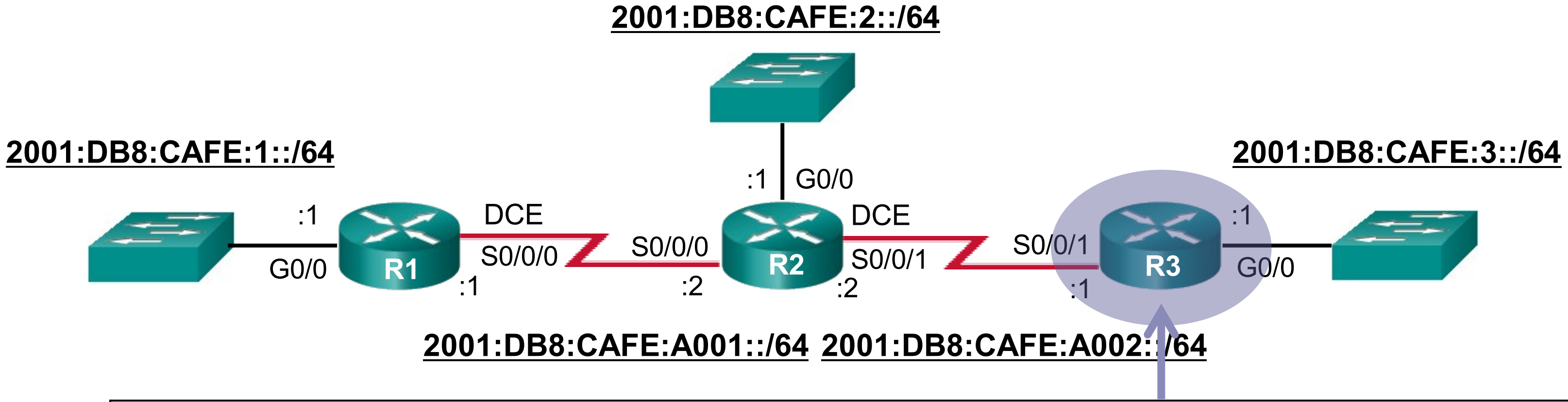

```
R1(config)# ipv6 unicast-routing
R1(config)#
R1(config)# interface gigabitethernet 0/0
R1(config-if)# ipv6 rip RIP-AS enable
R1(config-if)# exit
R1(config)#
R1(config)#interface serial 0/0/0
R1(config-if)# ipv6 rip RIP-AS enable
R1(config-if)#
```

# Enabling RIPng on the R2 Interfaces



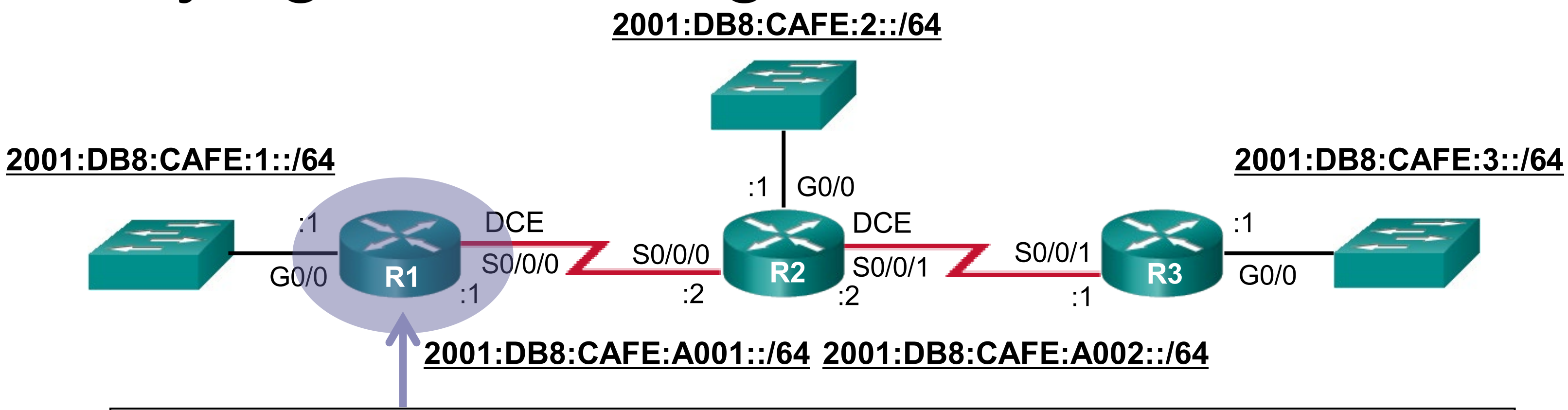

```
R2(config)# ipv6 unicast-routing
R2(config)#
R2(config)# interface gigabitethernet 0/0
R2(config-if)# ipv6 rip RIP-AS enable
R2(config-if)# exit
R2(config)#
R2(config)# interface serial 0/0/0
R2(config-if)# ipv6 rip RIP-AS enable
R2(config-if)# exit
R2(config)#
R2(config-if)# interface serial 0/0/1
R2(config-if)# ipv6 rip RIP-AS enable
R2(config-if)#
```

# Enabling RIPng on the R3 Interfaces




```
R3(config)# ipv6 unicast-routing
R3(config)#
R3(config)# interface gigabitethernet 0/0
R3(config-if)# ipv6 rip RIP-AS enable
R3(config-if)# exit
R3(config)#
R3(config)# interface serial 0/0/1
R3(config-if)# ipv6 rip RIP-AS enable
R3(config-if)#
R3#
*Mar 12 14:17:06.103: %SYS-5-CONFIG_I: Configured from
console by console)
```

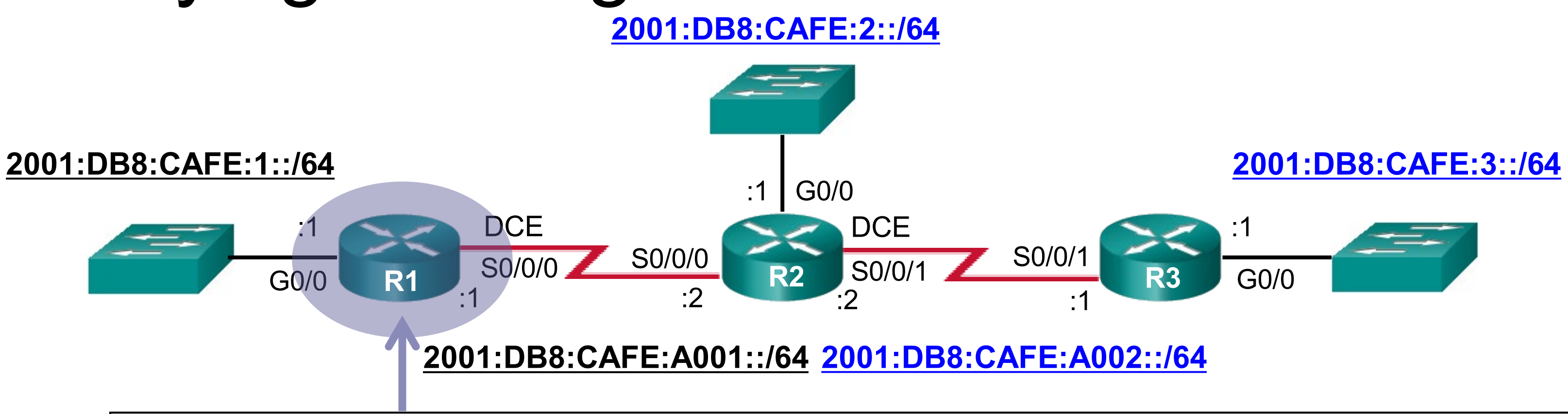
# Verifying RIP Settings on R1




```
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "rip RIP-AS"
  Interfaces:
    Serial0/0/0
    GigabitEthernet0/0
  Redistribution:
    None
R1#
```

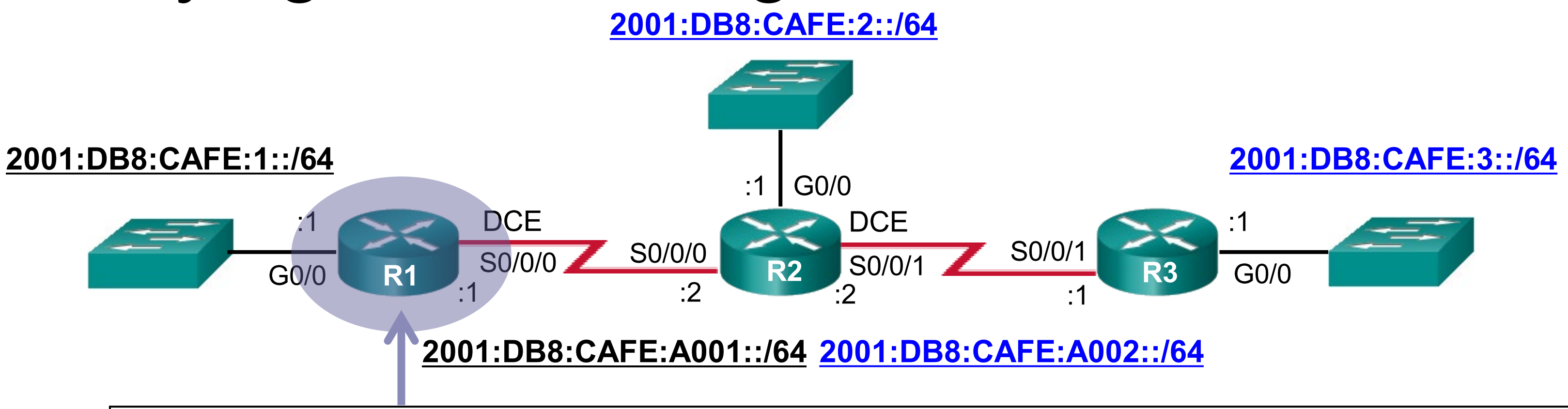
1
2

# Verifying Routing Table on R1



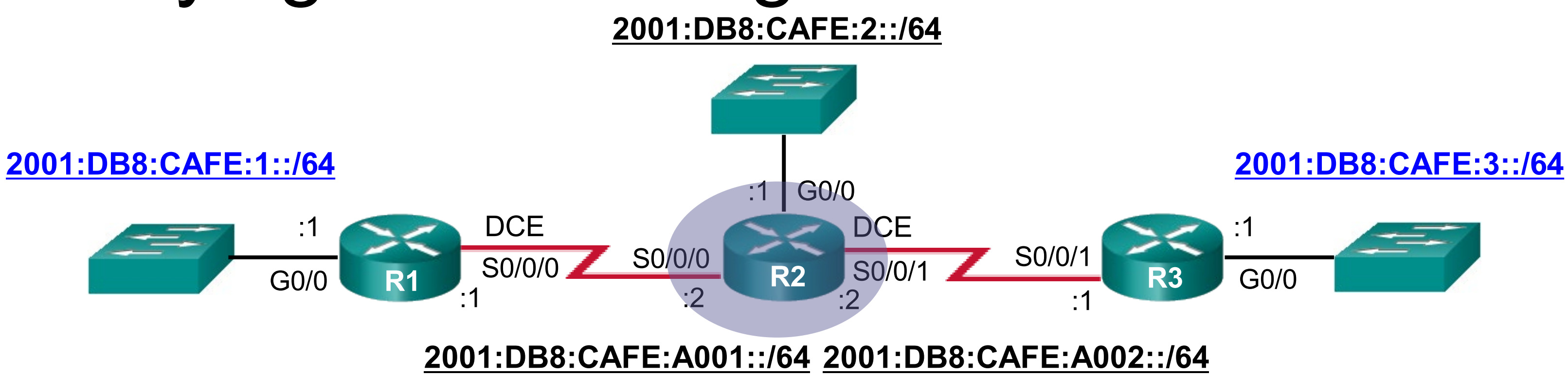

```
R1# show ipv6 route

<Output omitted>

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
R    2001:DB8:CAFE:2::/64 [120/2]
      via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R    2001:DB8:CAFE:3::/64 [120/3]
      via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
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
R    2001:DB8:CAFE:A002::/64 [120/2]
      via FE80::FE99:47FF:FE71:78A0, Serial0/0/0

<Output omitted>
```

# Verifying RIP Routing Table on R1



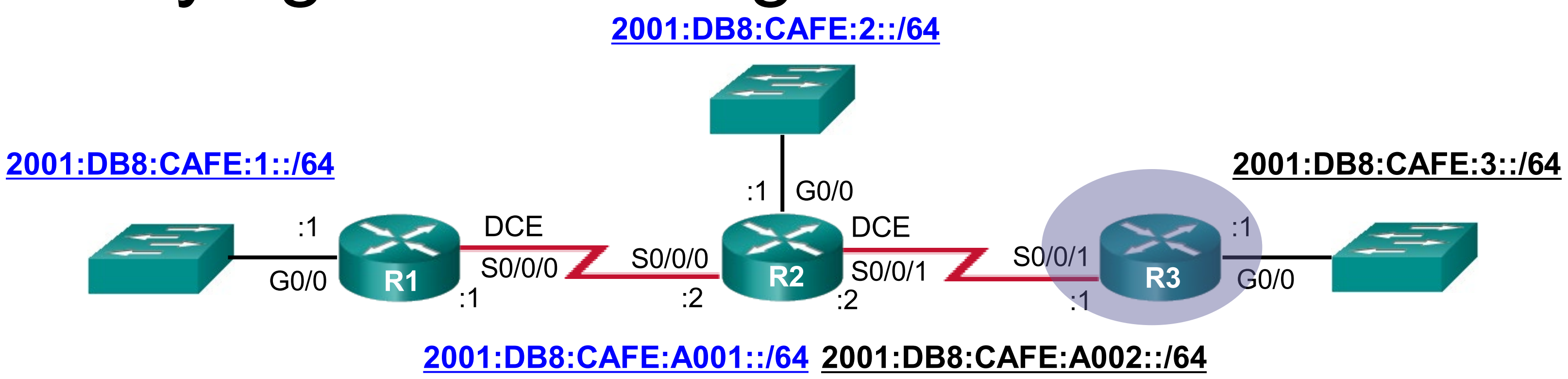

```
R1# show ipv6 route rip
IPv6 Routing Table - default - 8 entries
<output omitted>

R   2001:DB8:CAFE:2::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R   2001:DB8:CAFE:3::/64 [120/3]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R   2001:DB8:CAFE:A002::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/0
R1#
```

# Verifying RIP Routing Table on R2




```
R2# show ipv6 route rip
IPv6 Routing Table - default - 9 entries
<output omitted>

R   2001:DB8:CAFE:1::/64 [120/2]
     via FE80::FE99:47FF:FE75:C3E0, Serial0/0/0
R   2001:DB8:CAFE:3::/64 [120/2]

R2#
```

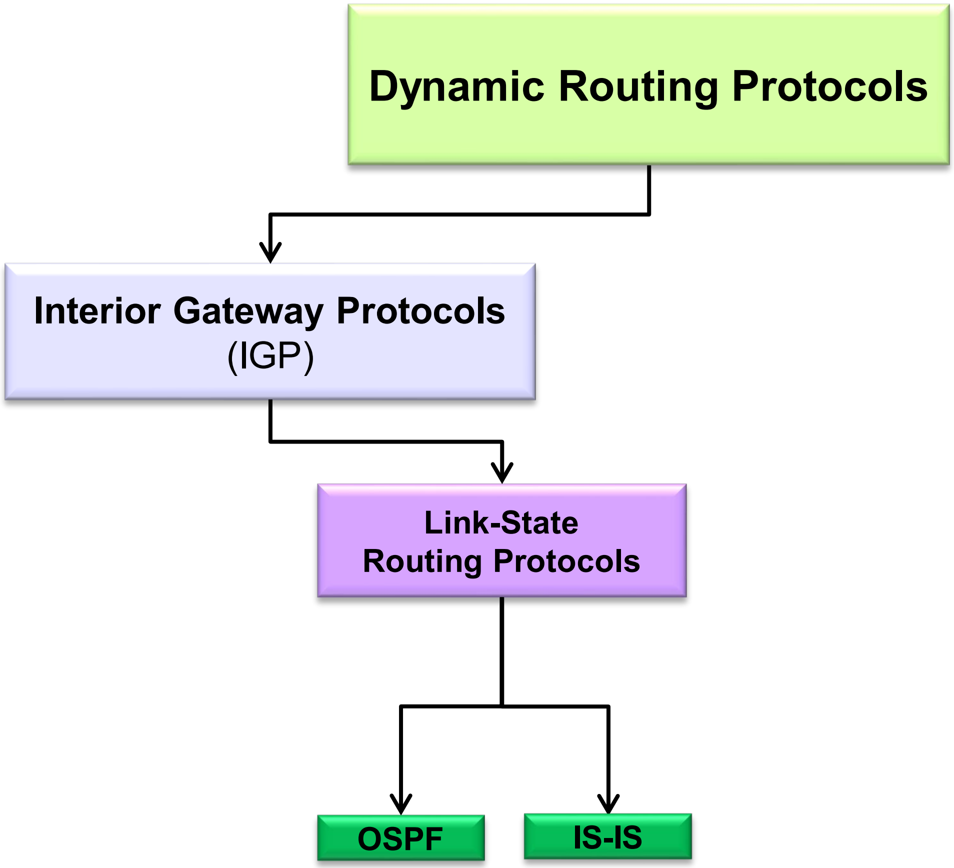
# Verifying RIP Routing Table on R3




```
R3# show ipv6 route rip
<Output omitted>

R   2001:DB8:CAFE:1::/64 [120/3]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/1
R   2001:DB8:CAFE:2::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/1
R   2001:DB8:CAFE:A001::/64 [120/2]
     via FE80::FE99:47FF:FE71:78A0, Serial0/0/1
R3#
```

# Link-State Routing Protocols

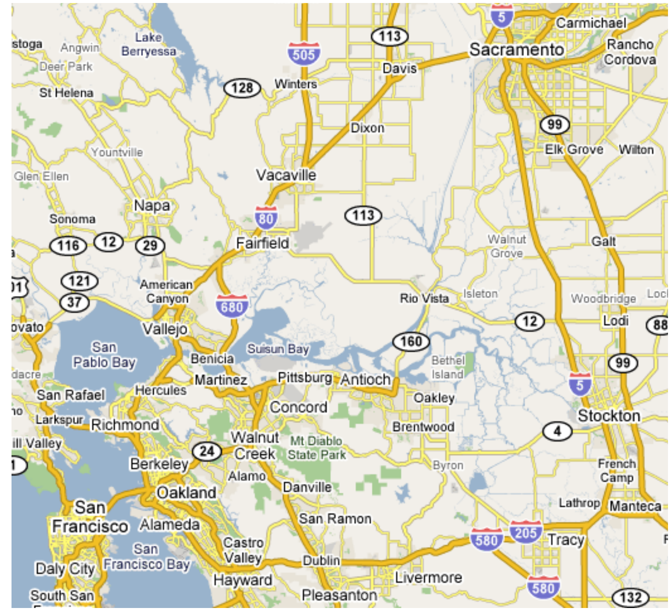
# Classifying Routing Protocols




- **Open Shortest Path First (OSPF)** is the most common IGP routing protocol implemented within an organizational AS.
  - Began in 1987
  - OSPFv2 - OSPF for IPv4 networks
  - OSPFv3 - OSPF for IPv6 networks
- **Intermediate-System to Intermediate-System (IS-IS)** is a less popular link-state protocol sometimes used within service provider networks.
  - IS-IS was designed by International Organization for Standardization (ISO )

# Shortest Path First (SPF) Algorithm


Distance Vector




Link-State

- Link-state routing protocols (a.k.a. shortest path first protocols) are based on Edsger Dijkstra's shortest path first (SPF) algorithm.

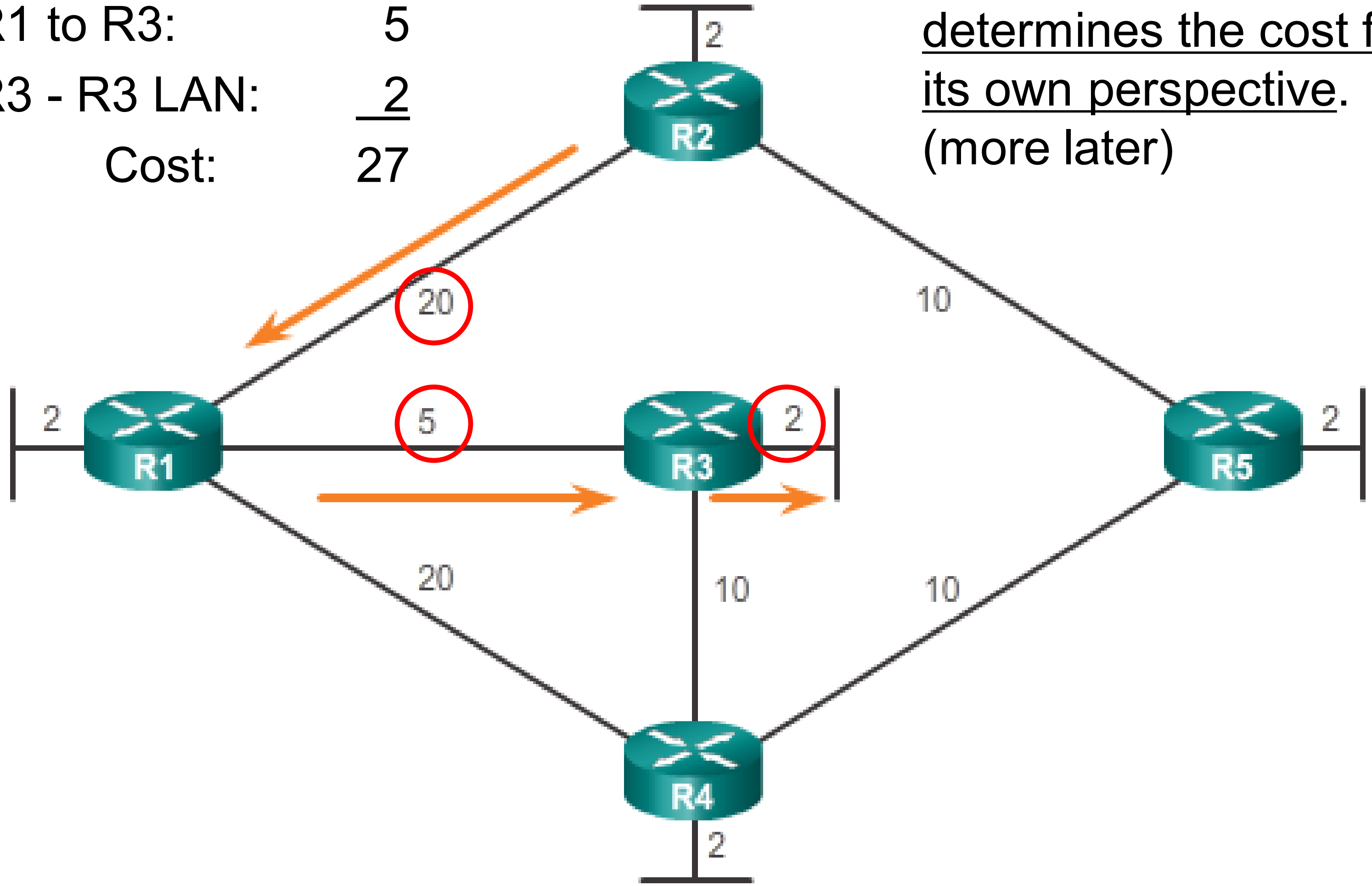
- The SPF algorithm is used to calculate the best path route.
  - The SPF algorithm uses accumulated costs along each path, from source to destination, to determine the total cost of a route.

# Shortest Path First (SPF) Algorithm

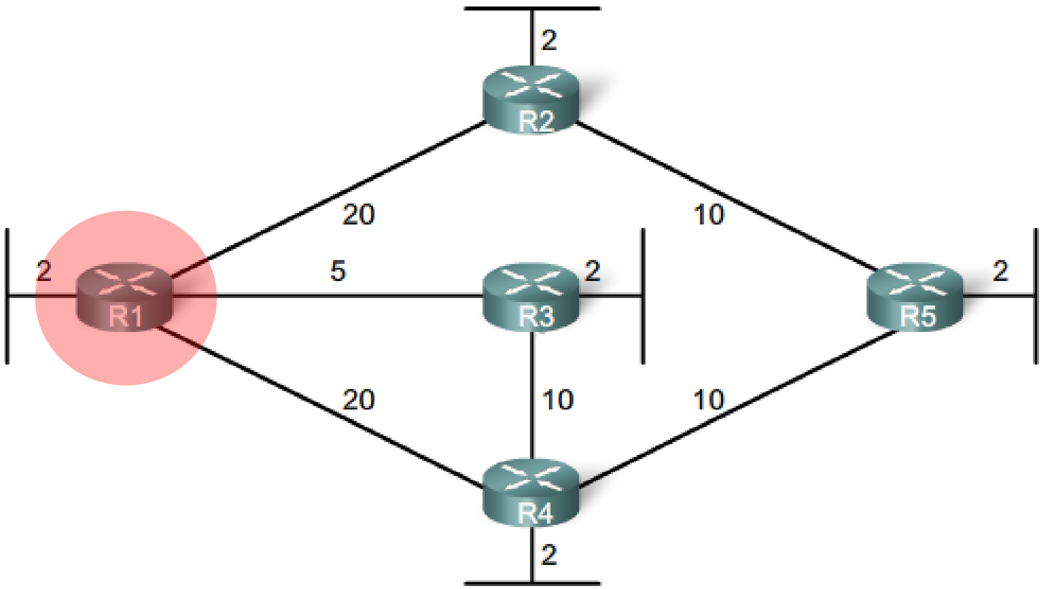
- Shortest path from R2 to the R3 LAN:
  - R2 to R1: 20
  - R1 to R3: 5
  - R3 - R3 LAN: 2

  Cost: 27

- Each router calculates the SPF algorithm and determines the cost from its own perspective. (more later)

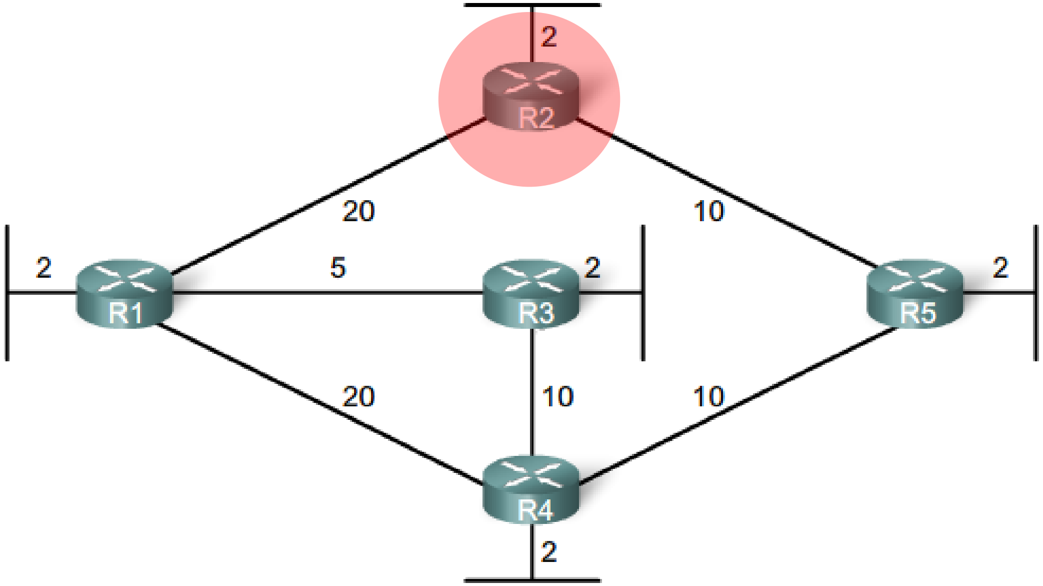


# R1 SPF Tree



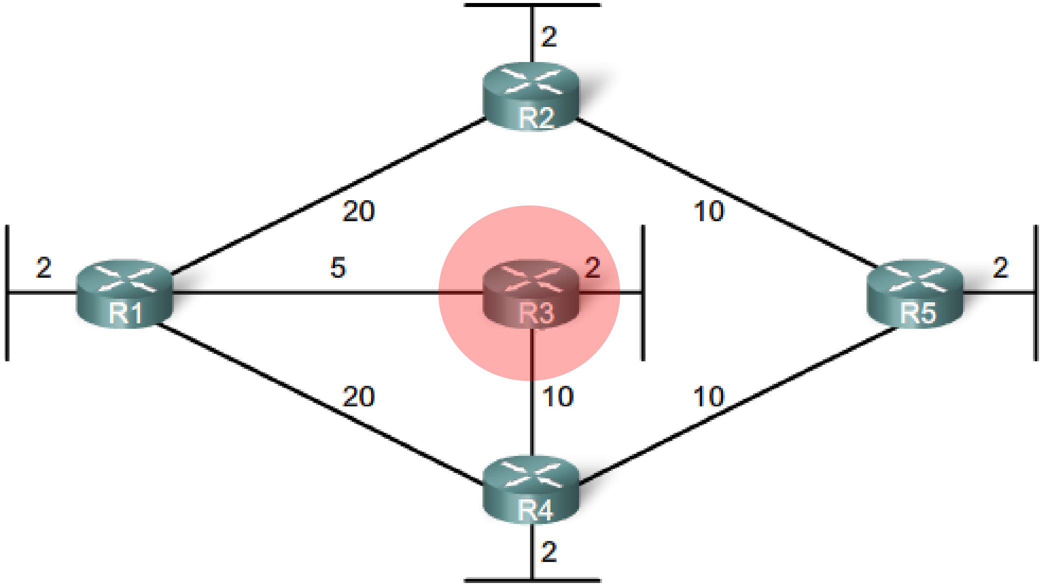

| Destination | Shortest Path | Cost |
|---|---|---|
| R2 LAN | R1 to R2 | 22 |
| R3 LAN | R1 to R3 | 7 |
| R4 LAN | R1 to R3 to R4 | 17 |
| R5 LAN | R1 to R3 to R4 to R5 | 27 |

# R2 SPF Tree



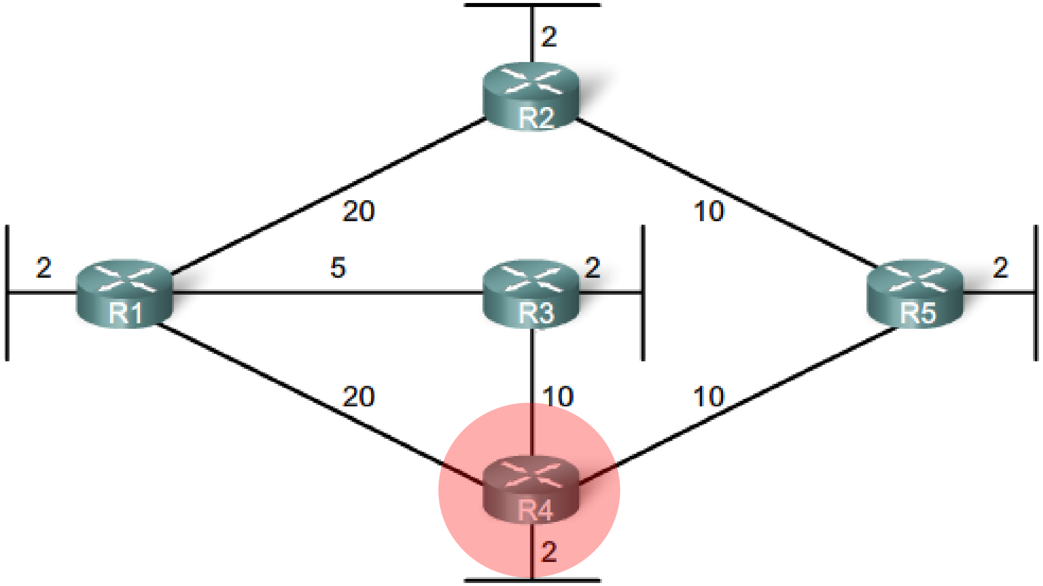

| Destination | Shortest Path | Cost |
|---|---|---|
| R1 LAN | R2 to R1 | 22 |
| R3 LAN | R2 to R1 to R3 | 27 |
| R4 LAN | R2 to R5 to R4 | 22 |
| R5 LAN | R2 to R5 | 12 |

# R3 SPF Tree



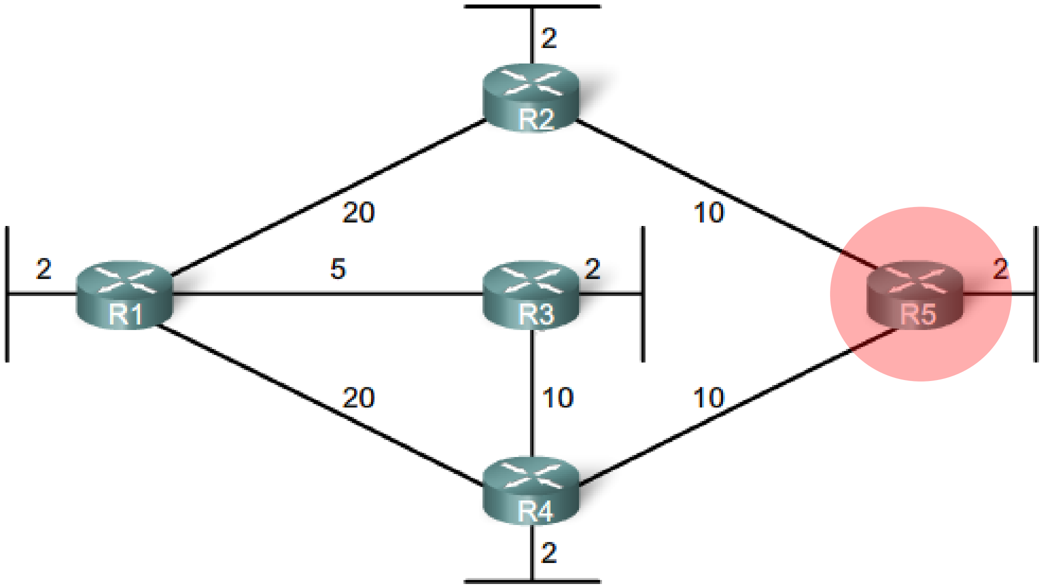

| Destination | Shortest Path | Cost |
|---|---|---|
| R1 LAN | R3 to R1 | 7 |
| R2 LAN | R3 to R1 to R2 | 27 |
| R4 LAN | R3 to R4 | 12 |
| R5 LAN | R3 to R4 to R5 | 22 |

# R4 SPF Tree




| Destination | Shortest Path | Cost |
|---|---|---|
| R1 LAN | R4 to R3 to R1 | 17 |
| R2 LAN | R4 to R5 to R2 | 22 |
| R3 LAN | R4 to R3 | 12 |
| R5 LAN | R4 to R5 | 12 |

# R5 SPF Tree




| Destination | Shortest Path | Cost |
|---|---|---|
| R1 LAN | R5 to R4 to R3 to R1 | 27 |
| R2 LAN | R5 to R2 | 12 |
| R3 LAN | R5 to R4 to R3 | 22 |
| R4 LAN | R5 to R4 | 12 |

# Link-State Routing Process

**1.** Each router learns about its own links, its own directly connected networks. *(Interface is "up")*

**2.** Each router is responsible for meeting its neighbors on directly connected networks. *(OSPF Hello packets)*

**3.** Each router builds a ***link-state packet (LSP)*** containing the state of each directly connected link. *(neighbor ID, link type, and bandwidth)*

**4.** Each router floods the LSP to all neighbors, who then store all LSPs received in a database.

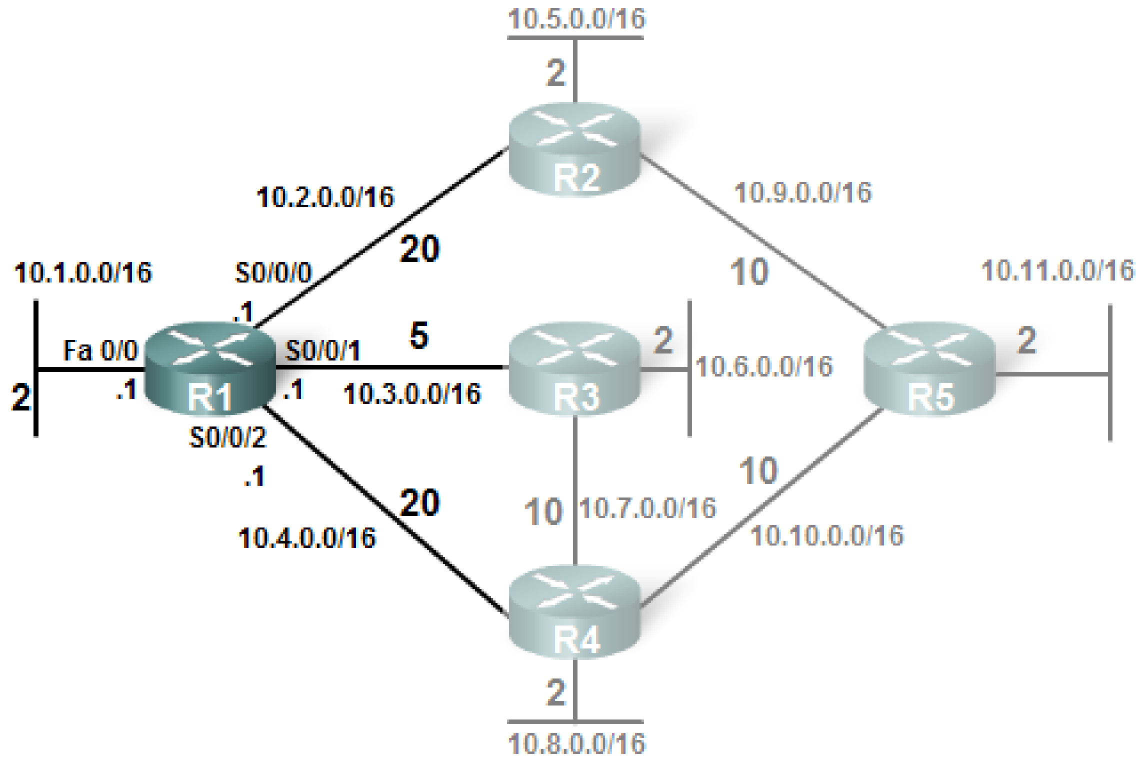
- Neighbors then flood the LSPs to their neighbors until all routers in the area have received the LSPs.

**5.** Each router uses the database to construct a complete map of the topology and computes the best path to each destination network.

- The SPF algorithm is used to construct the map of the topology and to determine the best path to each network. *(Road map)*
- All routers will have a common map or tree of the topology, but each router will independently determine the best path to each network within that topology.

- Detail and explanations are coming next!

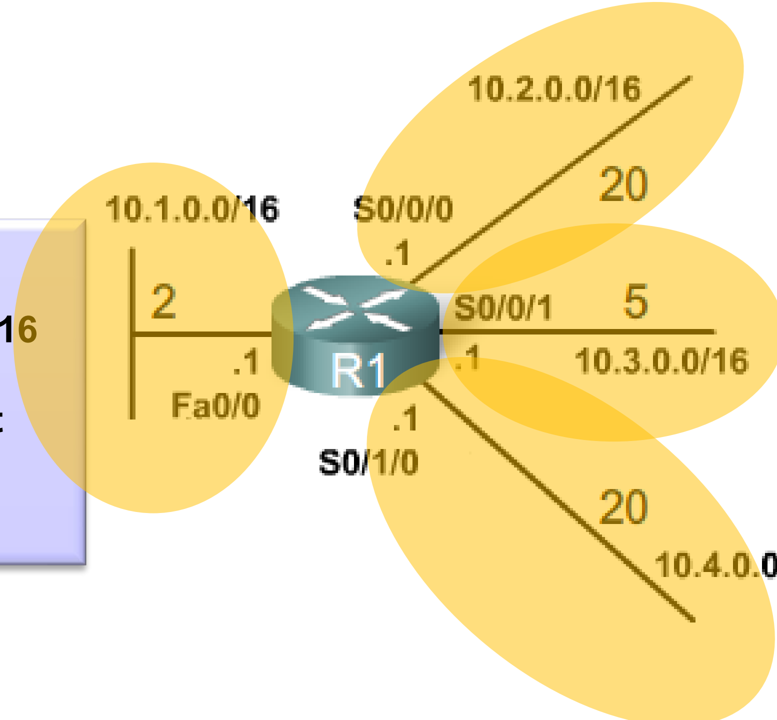
# Step 1: Learning About Directly Connected Networks




- ***Step 1: Each router learns about its own links, its own directly connected networks.***
  - Interface configured with an IP address/subnet mask.
  - Directly connected networks are now part of the routing table
    - Regardless of the routing protocols used.
- A ***link*** is an interface on a router.
- For the link participate in the link-state routing process, it must be:
  - In the up state.
  - Included in the routing protocol (coming).

# Step 1

**Link 1**
- Network: **10.1.0.0/16**
- IP address: **10.1.0.1**
- Type of network: **Ethernet**
- Cost of that link: **2**
- Neighbors: **None**



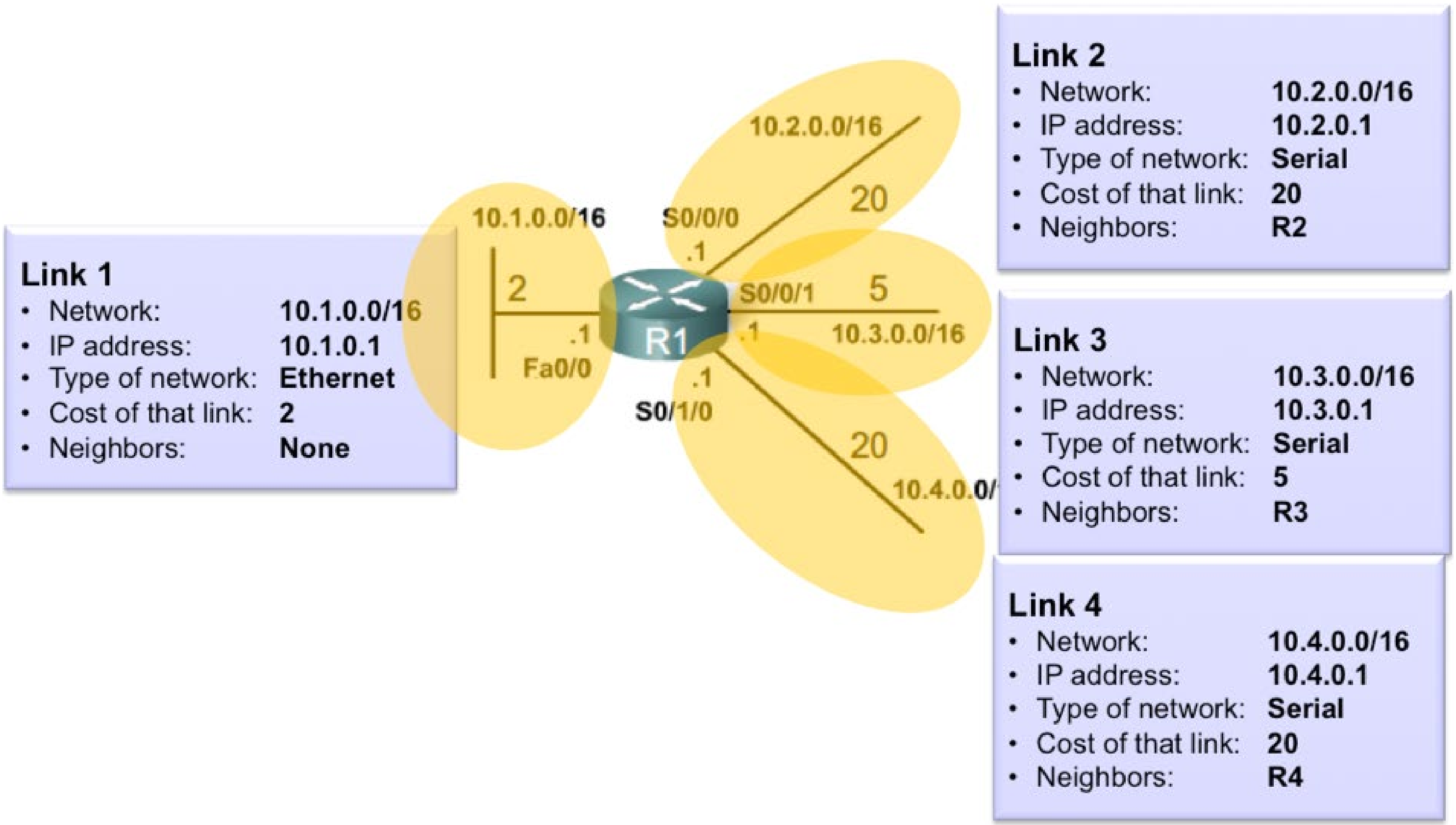

**Link 2**
- Network: **10.2.0.0/16**
- IP address: **10.2.0.1**
- Type of network: **Serial**
- Cost of that link: **20**
- Neighbors: **R2**

**Link 3**
- Network: **10.3.0.0/16**
- IP address: **10.3.0.1**
- Type of network: **Serial**
- Cost of that link: **5**
- Neighbors: **R3**

**Link 4**
- Network: **10.4.0.0/16**
- IP address: **10.4.0.1**
- Type of network: **Serial**
- Cost of that link: **20**
- Neighbors: **R4**

- ***Link states* -** Information about the state of a router’ s links
- This information includes interface’ s:
  - IP address/mask
  - Type of network
    - Ethernet (broadcast) or serial point-to-point link
  - Cost of that link
  - Any neighbor routers on that link

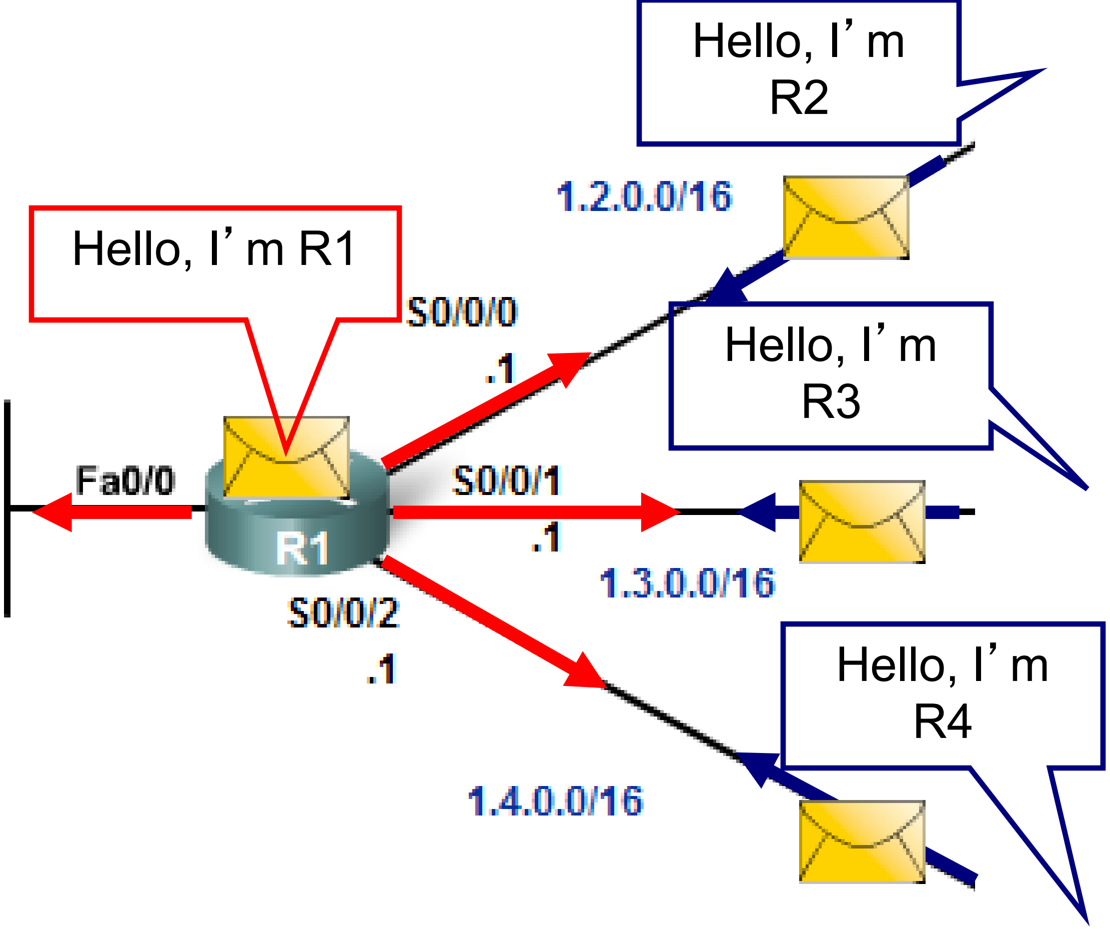
# Step 1




Initially:

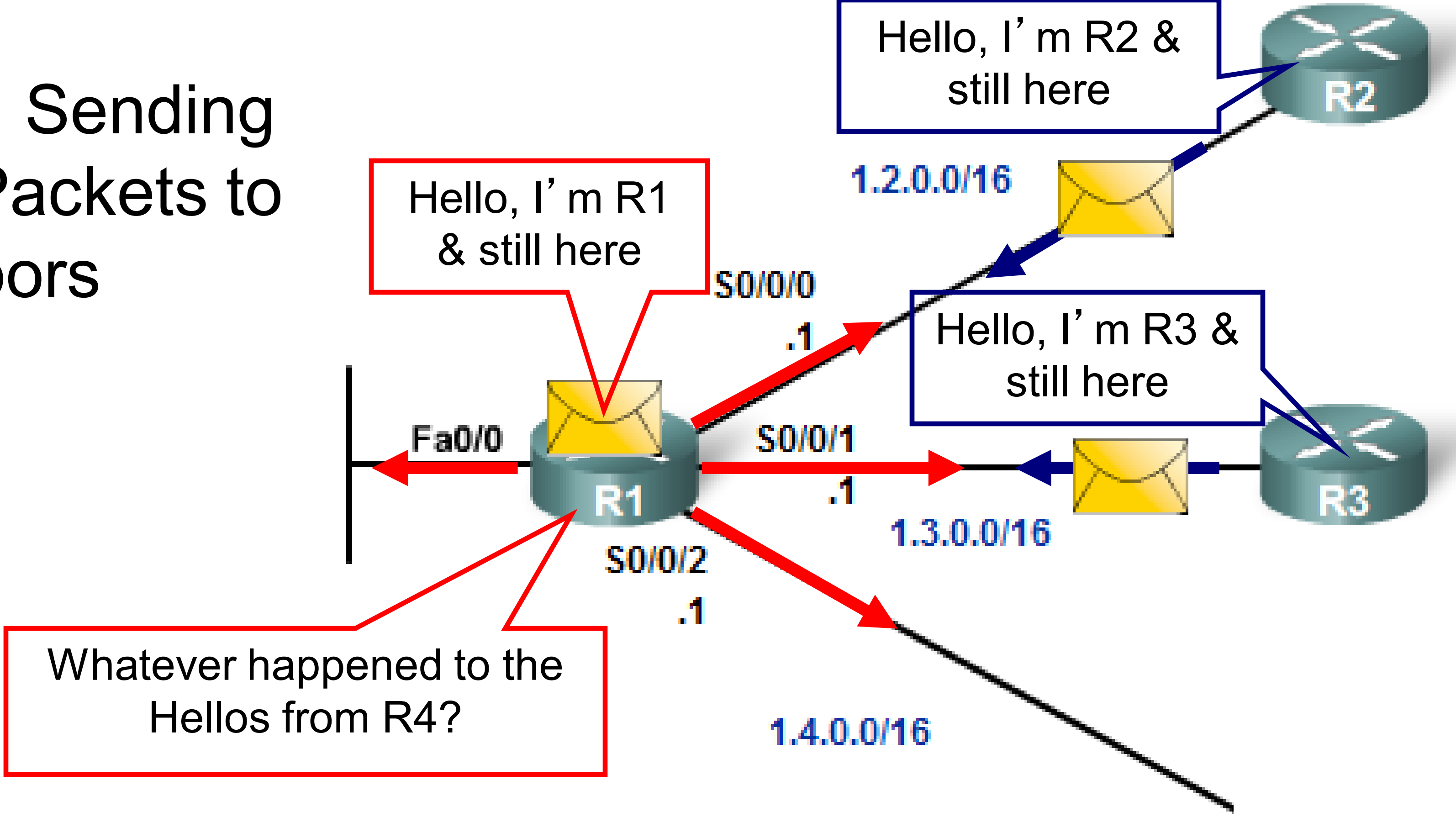
- Router unaware of any neighbor routers on the link.
- Learns of neighbor when receives a Hello packet from the adjacent neighbor.

# Step 2: Sending Hello Packets to Neighbors



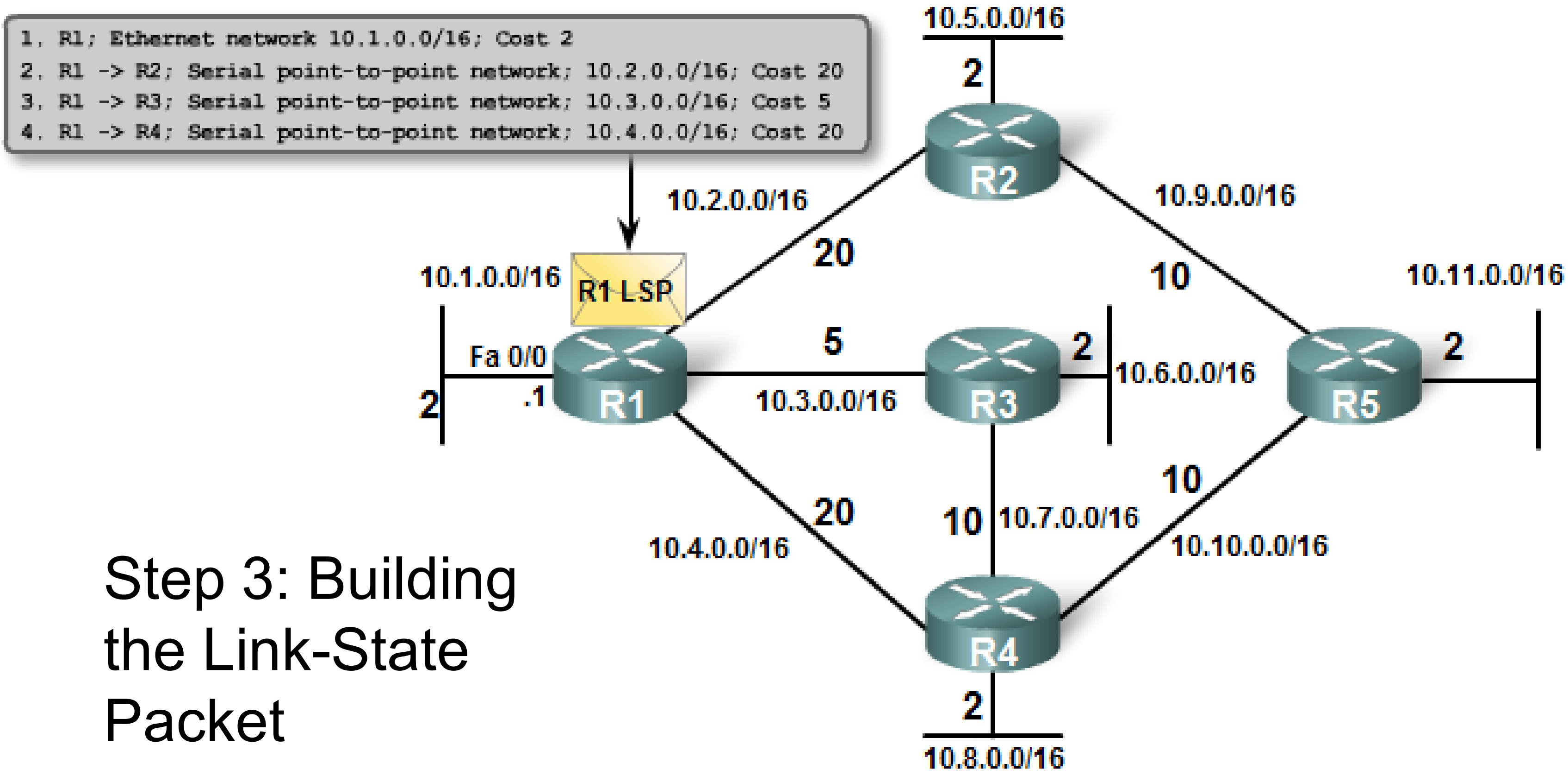

- **Step 2:** ***Each router is responsible for meeting its neighbors on directly connected networks.***
  - Use a Hello protocol to discover any neighbors on their links.
  - A **neighbor** is any other router that is enabled with the same link-state routing protocol.

# Step 2: Sending Hello Packets to Neighbors




**Hello packets**

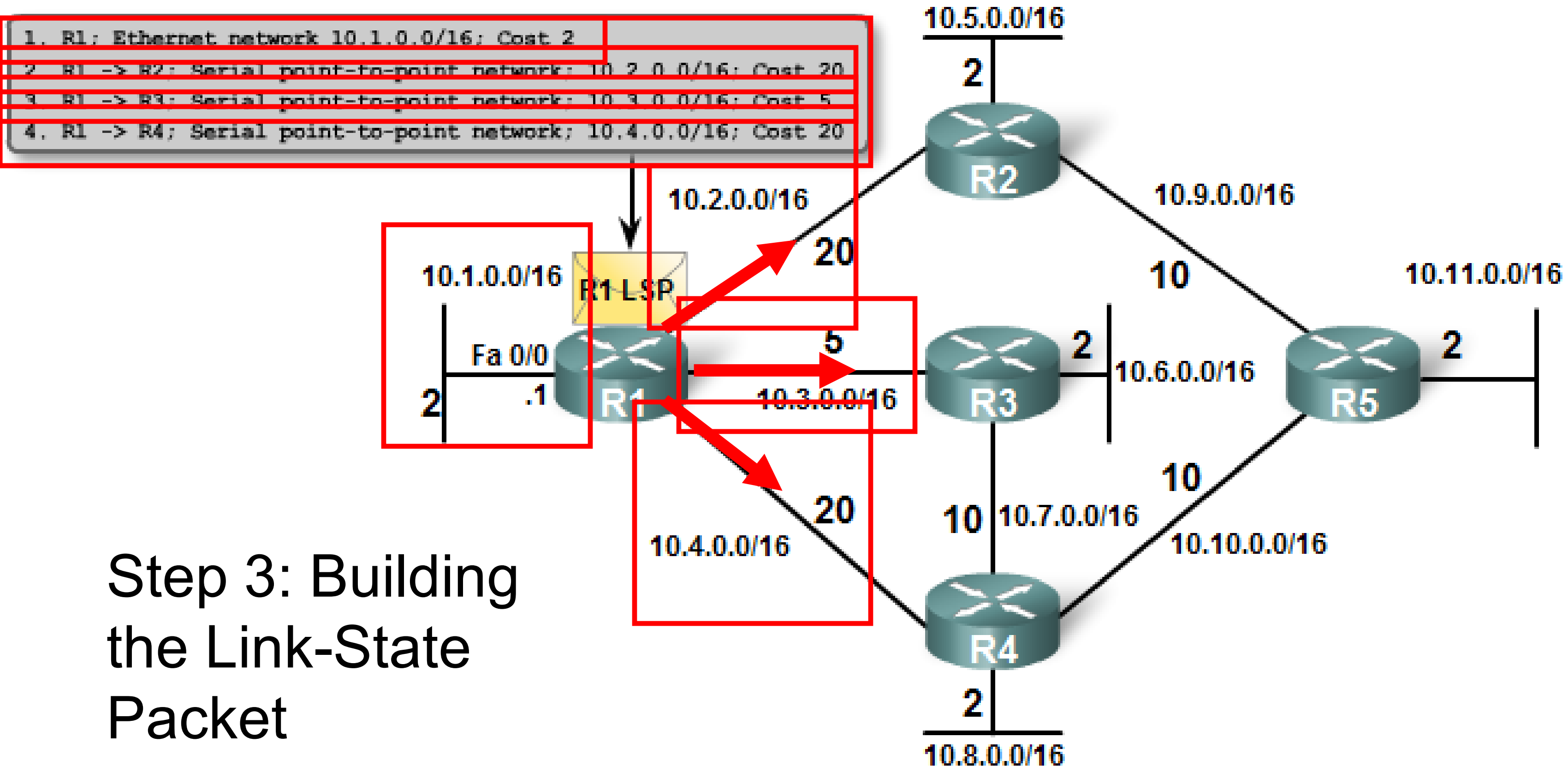
- “Keepalive” function
- Stops receiving Hello packets from a neighbor, that neighbor is considered unreachable and the adjacency is broken.




Step 3: Building the Link-State Packet

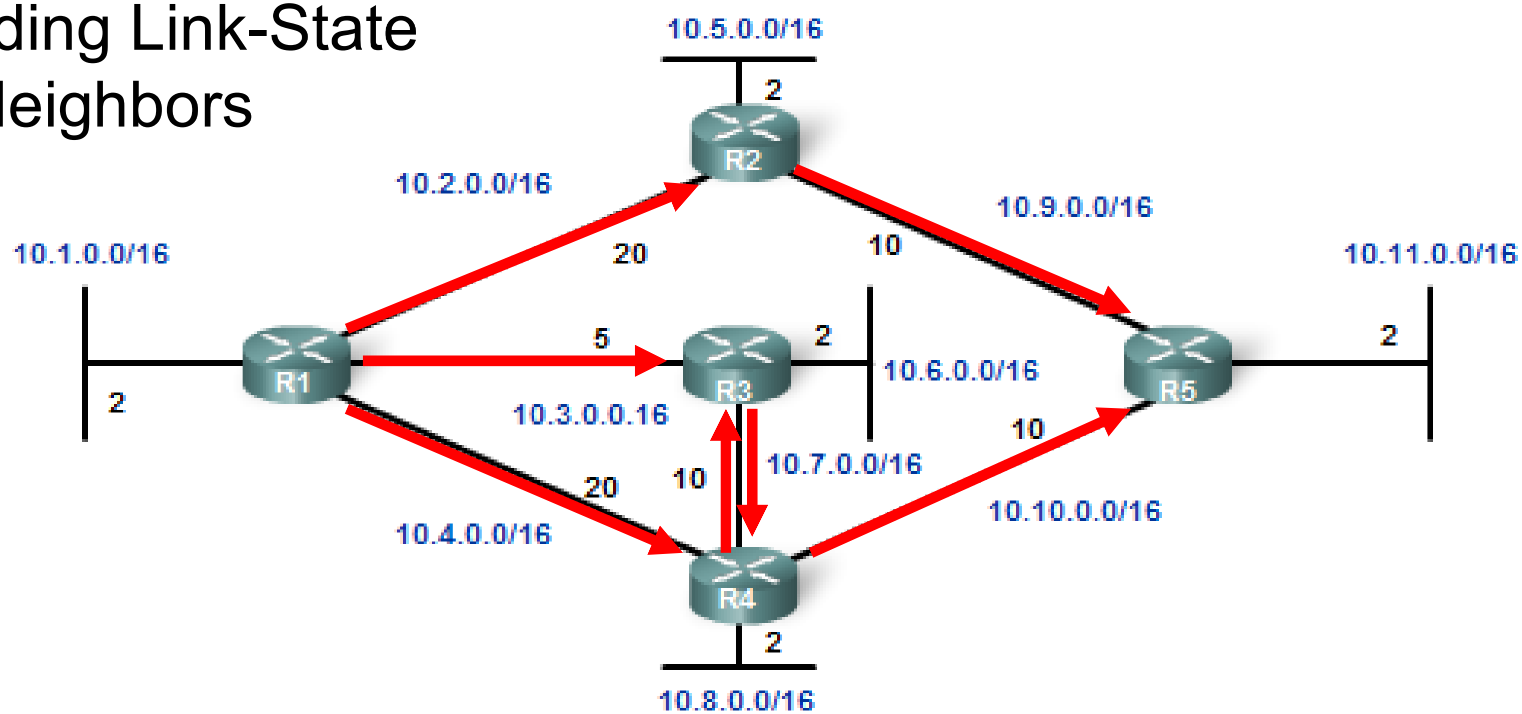
- **Step 3:** ***Each router builds a link-state packet (LSP) containing the state of each directly connected link.***

# Step 3: Building the Link-State Packet



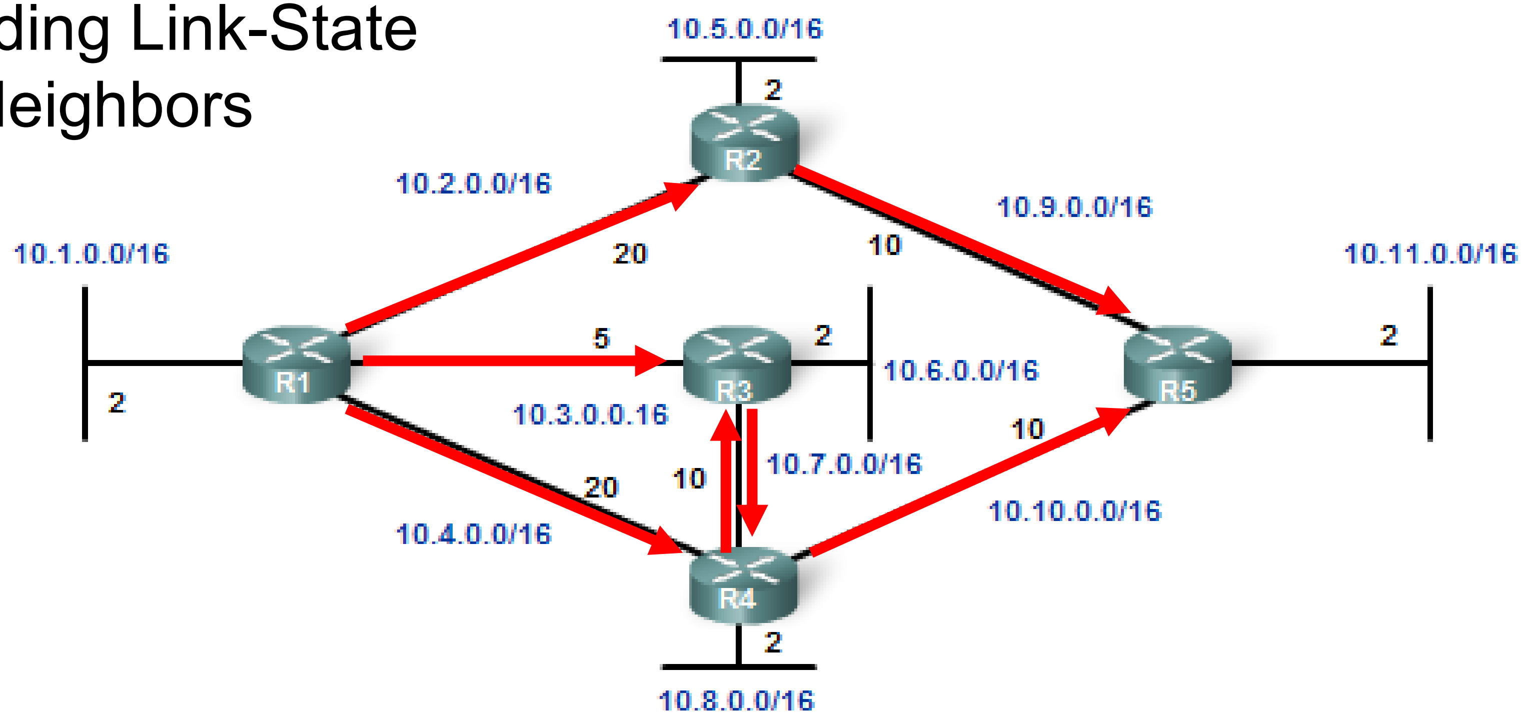

- **After established its adjacencies**
  - Builds its LSPs
    - Link-state information about its links.
  - Sends LSPs out interfaces where it has established adjacencies with other routers.
    - R1 not sent LSPs out its Ethernet interface.

# Step 4: Flooding Link-State Packets to Neighbors




- **Step 4: *Each router floods the LSP to all neighbors, who then store all LSPs received in a database.***
  - Each router <u>floods its link-state information</u> to all other link-state routers.
  - When a router <u>receives an LSP</u> from a neighboring router, <u>sends that LSP out all other interfaces</u>, except the interface that received the LSP.
  - <u>Flooding effect</u> of LSPs throughout the routing area.
- Link-state routing protocols <u>calculate the SPF algorithm after the flooding is complete.</u>

# Step 4: Flooding Link-State Packets to Neighbors




- An LSP needs to be sent only:
  - During initial startup of the router or of the routing protocol process on that router
  - Whenever there is a change in the topology,
    - link going down
    - link coming up
    - neighbor adjacency being established
    - neighbor adjacency being broken

*Link State Database for R1*

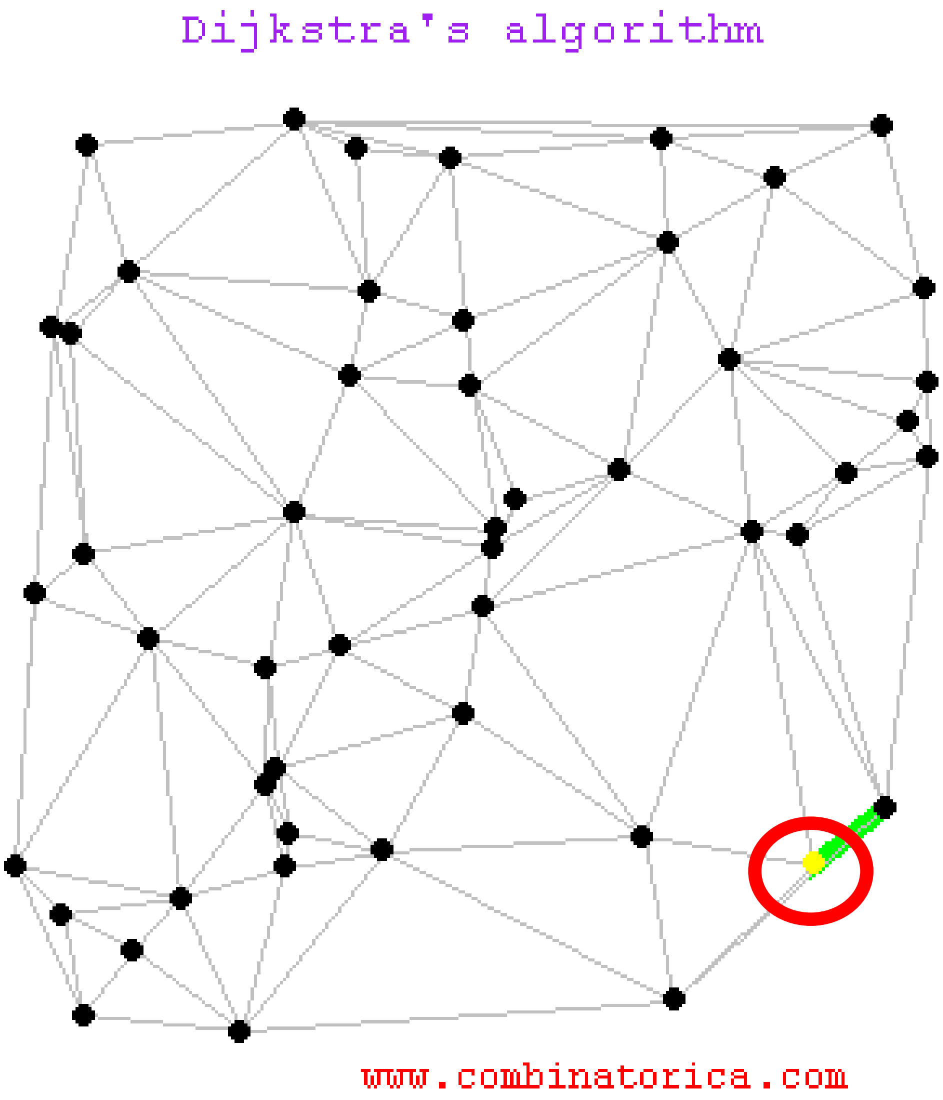
# Step 5: Constructing a Link-State Database

- Step 5 (Final Step): *Each router uses the database to construct a complete map of the topology and computes the best path to each destination network.*

| | |
|---|---|
| **LSPs from R2** | Connected to neighbor R1 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R5 on network 10.9.0.0/16, cost of 10 |
| | Has a network 10.5.0.0/16, cost of 2 |
| **LSPs from R3** | Connected to neighbor R1 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.7.0.0/16, cost of 10 |
| | Has a network 10.6.0.0/16, cost of 2 |
| **LSPs from R4** | Connected to neighbor R1 on network 10.4.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.7.0.0/16, cost of 10 |
| | Connected to neighbor R5 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.8.0.0/16, cost of 2 |
| **LSPs from R5** | Connected to neighbor R2 on network 10.9.0.0/16, cost of 10 |
| | Connected to neighbor R4 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.11.0.0/16, cost of 2 |
| **R1 link states** | Connected to neighbor R2 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.4.0.0/16, cost of 20 |
| | Has a network 10.1.0.0/16, cost of 2 |

- After propagation of LSPs
  - Each router will then have an LSP from every link-state router.
  - LSPs stored in the link-state database.

# Running SPF Algorithm




- Each router in the routing area can now use the SPF algorithm to construct the SPF trees that you saw earlier.

# Step 5: Constructing a Link-State Database

*SPF Tree for R1*

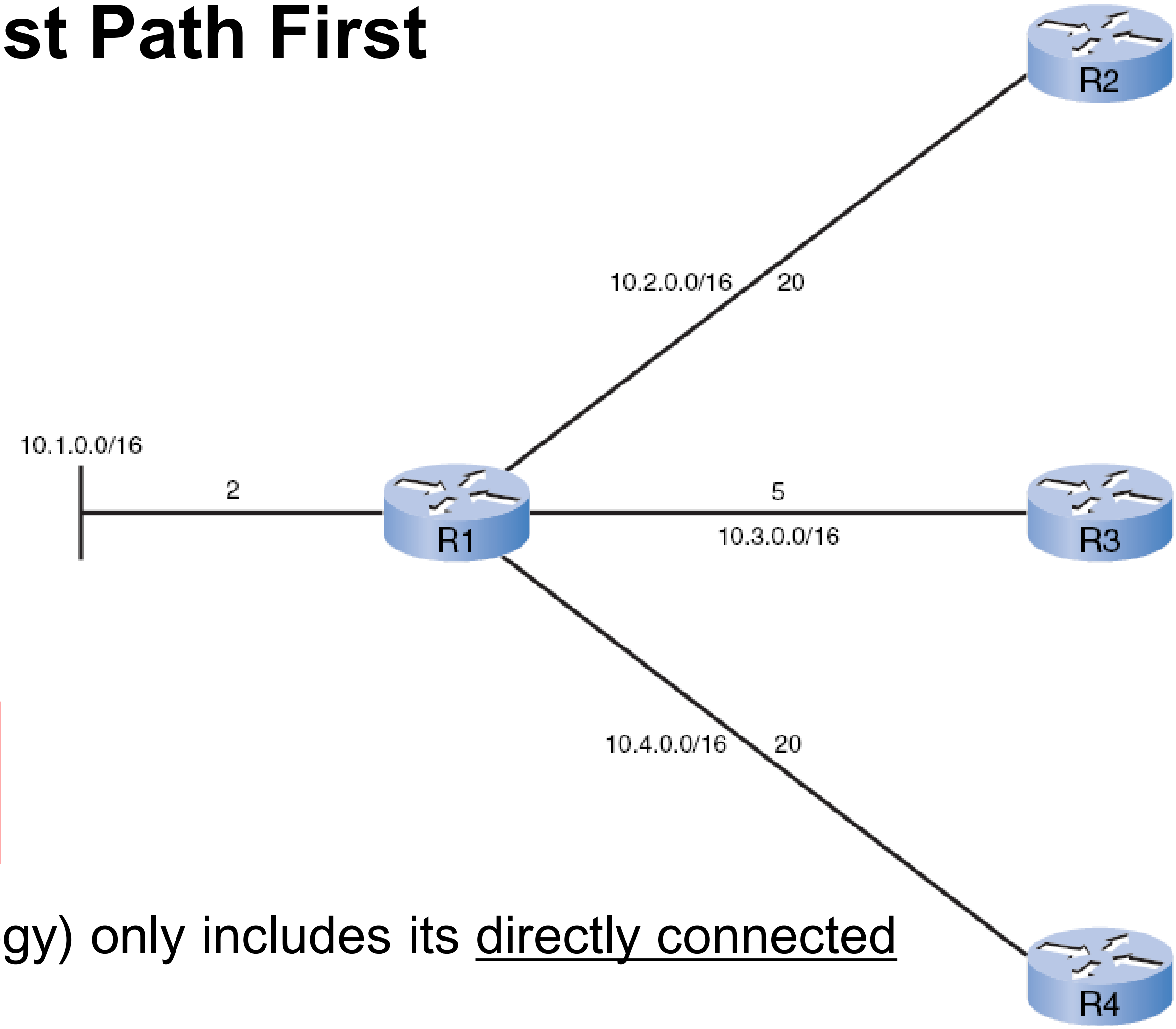
| Destination | Shortest Path | Cost |
|---|---|---|
| R2 LAN | R1 to R2 | 22 |
| R3 LAN | R1 to R3 | 7 |
| R4 LAN | R1 to R3 to R4 | 17 |
| R5 LAN | R1 to R3 to R4 to R5 | 27 |

- With a complete link-state database, R1 can use shortest path first (SPF) algorithm to calculate shortest path to each network.
- SPF algorithm results in an SPF tree.

# Building the Shortest Path First (SPF) Tree

*Link State Database for R1*

| | |
|---|---|
| **LSPs from R2** | Connected to neighbor R1 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R5 on network 10.9.0.0/16, cost of 10 |
| | Has a network 10.5.0.0/16, cost of 2 |
| **LSPs from R3** | Connected to neighbor R1 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.7.0.0/16, cost of 10 |
| | Has a network 10.6.0.0/16, cost of 2 |
| **LSPs from R4** | Connected to neighbor R1 on network 10.4.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.7.0.0/16, cost of 10 |
| | Connected to neighbor R5 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.8.0.0/16, cost of 2 |
| **LSPs from R5** | Connected to neighbor R2 on network 10.9.0.0/16, cost of 10 |
| | Connected to neighbor R4 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.11.0.0/16, cost of 2 |
| **R1 link states** | Connected to neighbor R2 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.4.0.0/16, cost of 20 |
| | Has a network 10.1.0.0/16, cost of 2 |

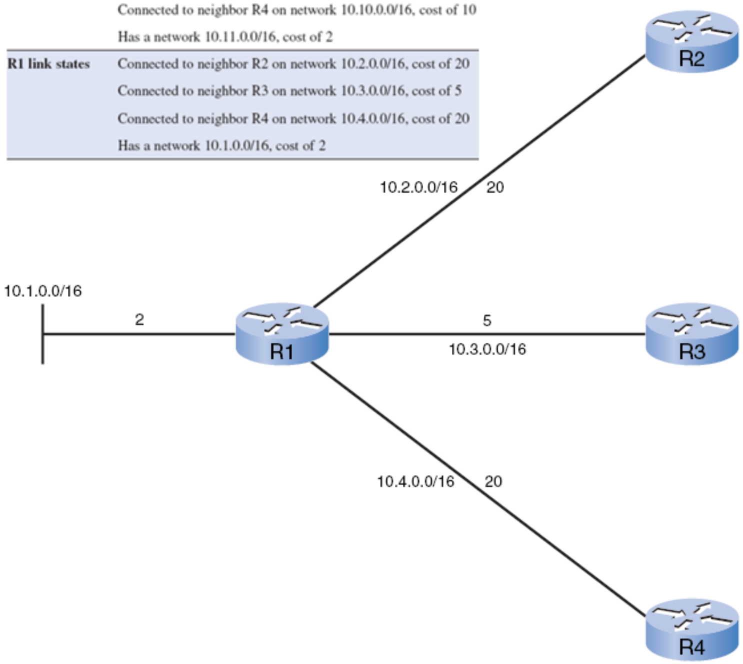



- At first, the tree (topology) only includes its directly connected neighbors.
- Using the link-state information from all other routers, R1 can now begin to construct an SPF tree of the network with itself at the root of the tree.

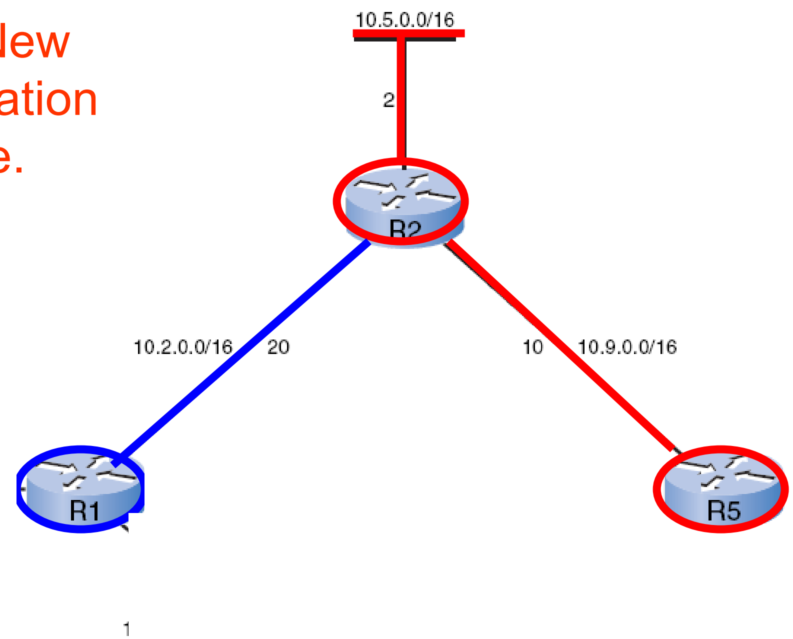
# R1 Processes the LSPs from R2

*Link State Database for R1*

| | |
|---|---|
| LSPs from R2 | Connected to neighbor R1 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R5 on network 10.9.0.0/16, cost of 10 |
| | Has a network 10.5.0.0/16, cost of 2 |
| LSPs from R3 | Connected to neighbor R1 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.7.0.0/16, cost of 10 |
| | Has a network 10.6.0.0/16, cost of 2 |
| LSPs from R4 | Connected to neighbor R1 on network 10.4.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.7.0.0/16, cost of 10 |
| | Connected to neighbor R5 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.8.0.0/16, cost of 2 |
| LSPs from R5 | Connected to neighbor R2 on network 10.9.0.0/16, cost of 10 |
| | Connected to neighbor R4 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.11.0.0/16, cost of 2 |
| R1 link states | Connected to neighbor R2 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.4.0.0/16, cost of 20 |
| | Has a network 10.1.0.0/16, cost of 2 |




Red: New information for tree.



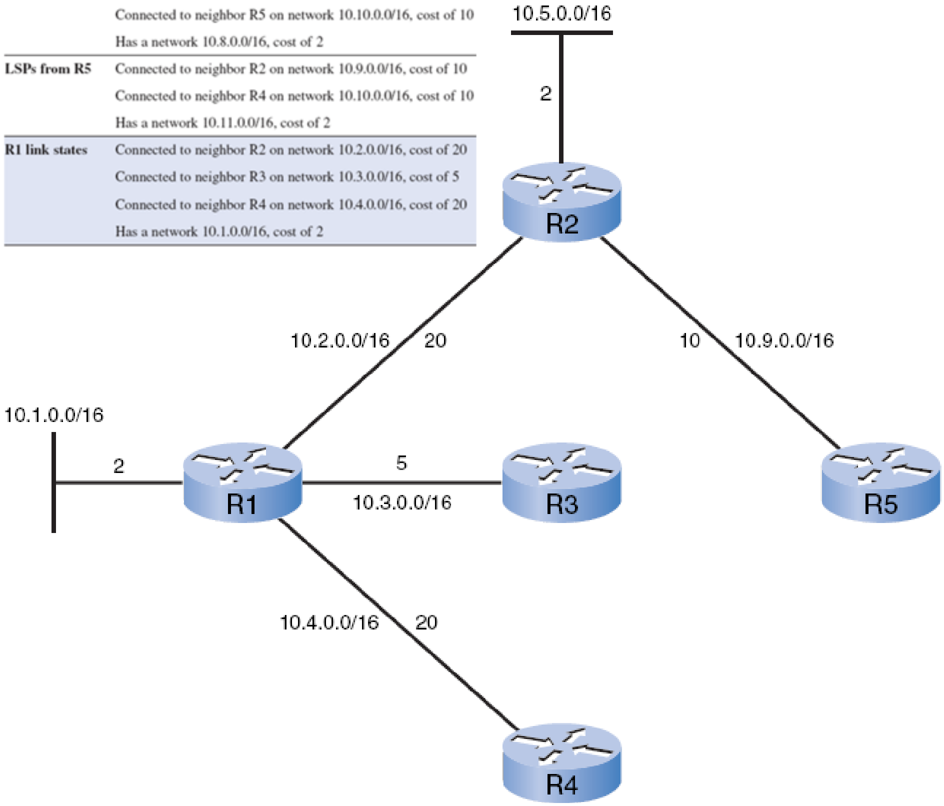

- The SPF algorithm begins by <u>processing the following LSP information from R2</u>:
  - Connected to neighbor R1 on network 10.2.0.0/16, cost of 20
  - Connected to neighbor R5 on network 10.9.0.0/16, cost of 10
  - Has a network 10.5.0.0/16, cost of 2

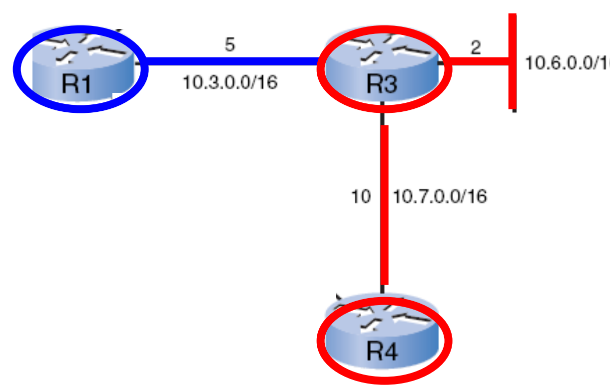
*Link State Database for R1*

| | |
|---|---|
| **LSPs from R2** | Connected to neighbor R1 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R5 on network 10.9.0.0/16, cost of 10 |
| | Has a network 10.5.0.0/16, cost of 2 |
| **LSPs from R3** | Connected to neighbor R1 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.7.0.0/16, cost of 10 |
| | Has a network 10.6.0.0/16, cost of 2 |
| **LSPs from R4** | Connected to neighbor R1 on network 10.4.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.7.0.0/16, cost of 10 |
| | Connected to neighbor R5 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.8.0.0/16, cost of 2 |
| **LSPs from R5** | Connected to neighbor R2 on network 10.9.0.0/16, cost of 10 |
| | Connected to neighbor R4 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.11.0.0/16, cost of 2 |
| **R1 link states** | Connected to neighbor R2 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.4.0.0/16, cost of 20 |
| | Has a network 10.1.0.0/16, cost of 2 |

# R1 Processes the LSPs from R3

Red: New information for tree.



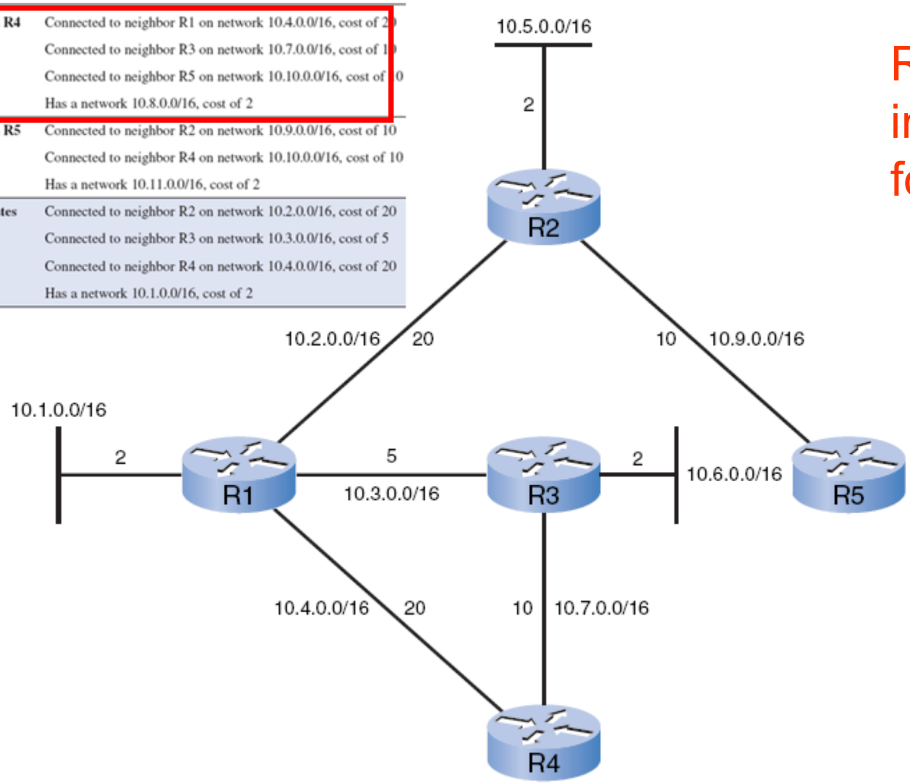




- The SPF algorithm begins by processing the following LSP information from R3:
  - Connected to neighbor R1 on network 10.3.0.0/16, cost of 5
  - Connected to neighbor R4 on network 10.7.0.0/16, cost of 10
  - Has a network 10.6.0.0/16, cost of 2

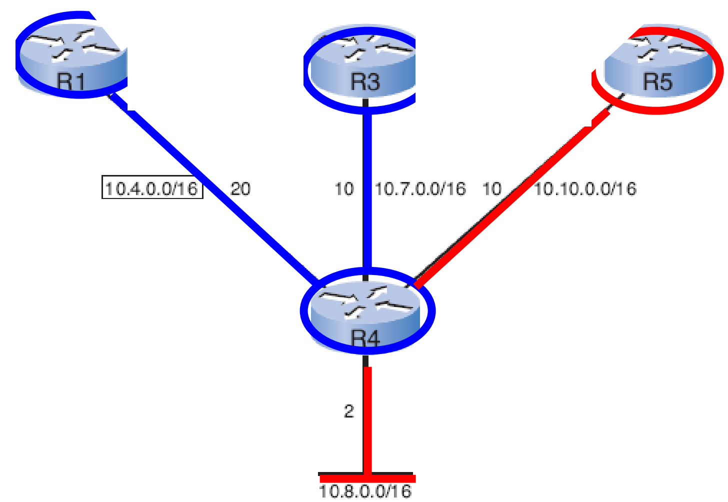
*Link State Database for R1*

| | |
|---|---|
| LSPs from R2 | Connected to neighbor R1 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R5 on network 10.9.0.0/16, cost of 10 |
| | Has a network 10.5.0.0/16, cost of 2 |
| LSPs from R3 | Connected to neighbor R1 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.7.0.0/16, cost of 10 |
| | Has a network 10.6.0.0/16, cost of 2 |
| LSPs from R4 | Connected to neighbor R1 on network 10.4.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.7.0.0/16, cost of 10 |
| | Connected to neighbor R5 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.8.0.0/16, cost of 2 |
| LSPs from R5 | Connected to neighbor R2 on network 10.9.0.0/16, cost of 10 |
| | Connected to neighbor R4 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.11.0.0/16, cost of 2 |
| R1 link states | Connected to neighbor R2 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.4.0.0/16, cost of 20 |
| | Has a network 10.1.0.0/16, cost of 2 |

# R1 Processes the LSPs from R4




Red: New information for tree.

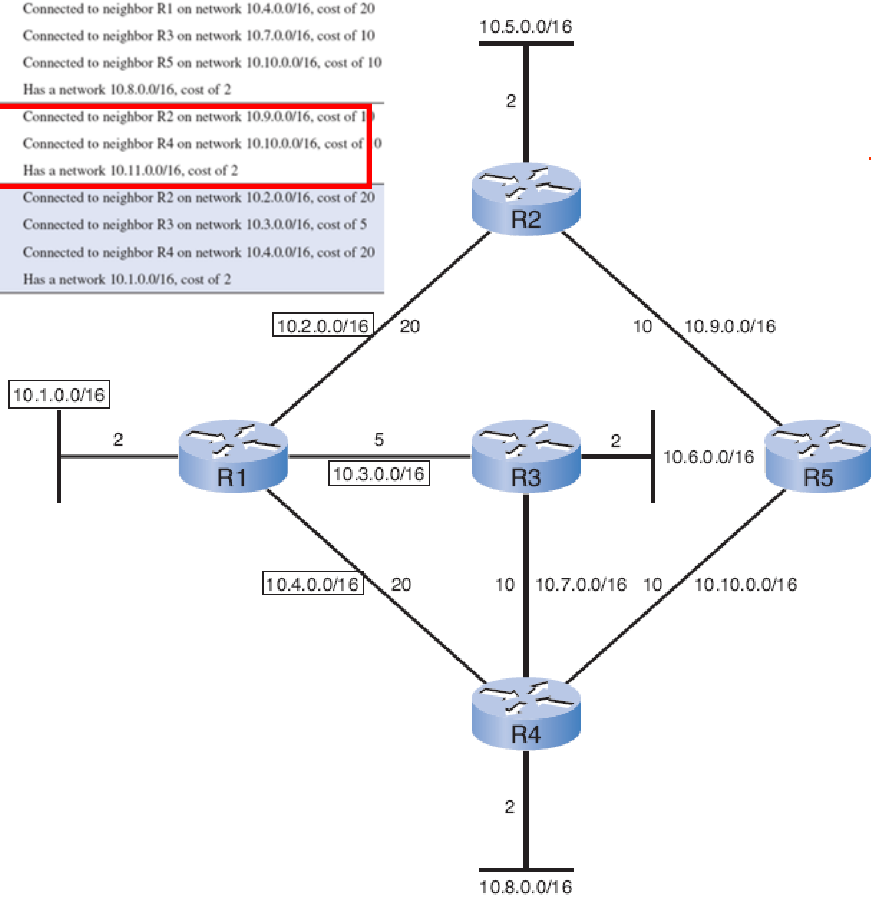



- The SPF algorithm begins by processing the following LSP information from R4:
  - Connected to neighbor R1 on network 10.4.0.0/16, cost of 20
  - Connected to neighbor R3 on network 10.7.0.0/16, cost of 10
  - Connected to neighbor R5 on network 10.10.0.0/16, cost of 10
  - Has a network 10.8.0.0/16, cost of 2

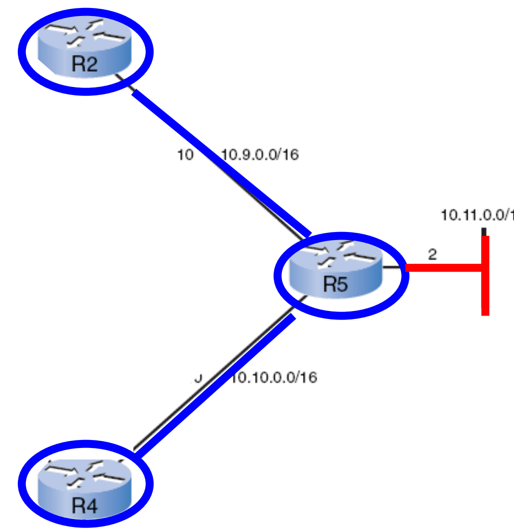
*Link State Database for R1*

| | |
|---|---|
| LSPs from R2 | Connected to neighbor R1 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R5 on network 10.9.0.0/16, cost of 10 |
| | Has a network 10.5.0.0/16, cost of 2 |
| LSPs from R3 | Connected to neighbor R1 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.7.0.0/16, cost of 10 |
| | Has a network 10.6.0.0/16, cost of 2 |
| LSPs from R4 | Connected to neighbor R1 on network 10.4.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.7.0.0/16, cost of 10 |
| | Connected to neighbor R5 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.8.0.0/16, cost of 2 |
| LSPs from R5 | Connected to neighbor R2 on network 10.9.0.0/16, cost of 10 |
| | Connected to neighbor R4 on network 10.10.0.0/16, cost of 10 |
| | Has a network 10.11.0.0/16, cost of 2 |
| R1 link states | Connected to neighbor R2 on network 10.2.0.0/16, cost of 20 |
| | Connected to neighbor R3 on network 10.3.0.0/16, cost of 5 |
| | Connected to neighbor R4 on network 10.4.0.0/16, cost of 20 |
| | Has a network 10.1.0.0/16, cost of 2 |

# R1 Processes the LSPs from R5

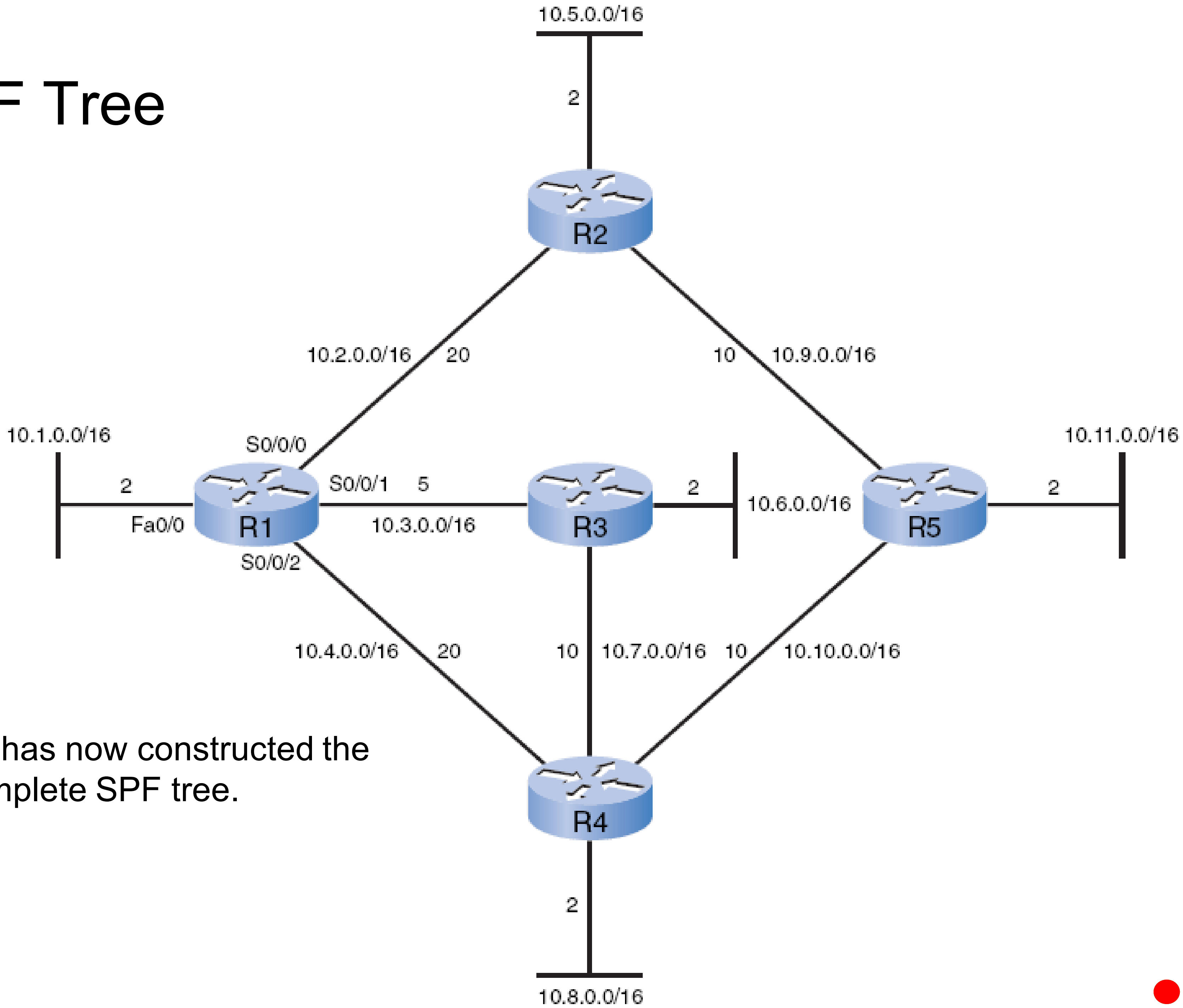



Red: New information for tree.



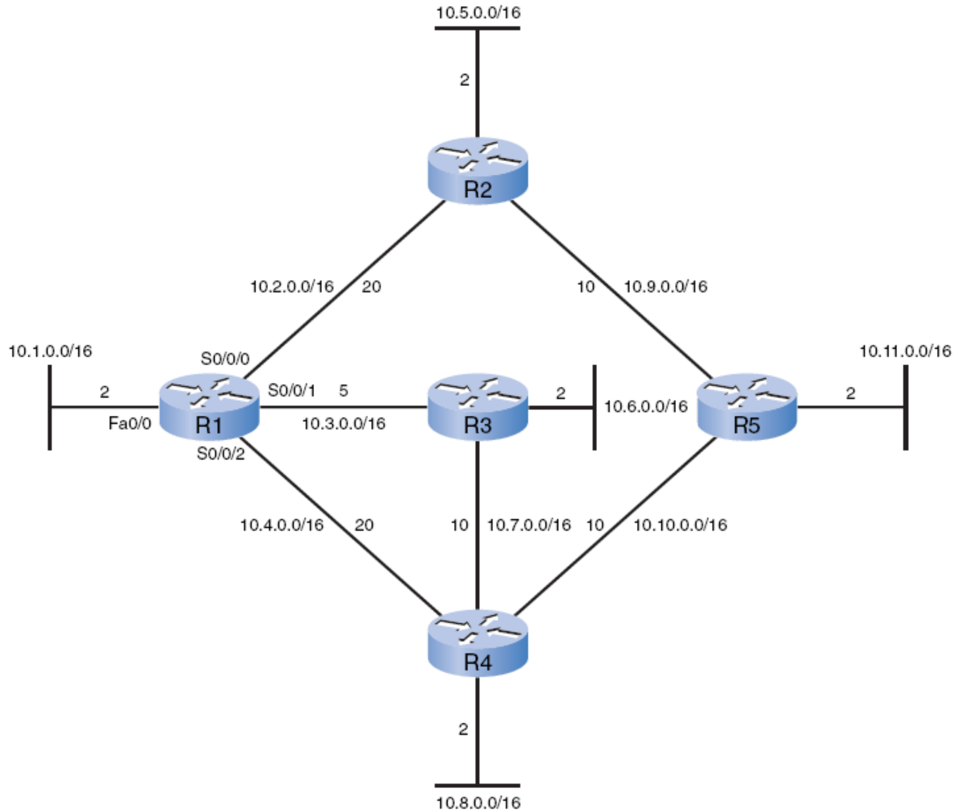

- The SPF algorithm begins by processing the following LSP information from R5:
  - Connected to neighbor R2 on network 10.9.0.0/16, cost of 10
  - Connected to neighbor R4 on network 10.10.0.0/16, cost of 10
  - Has a network 10.11.0.0/16, cost of 2

# SPF Tree




- R1 has now constructed the complete SPF tree.

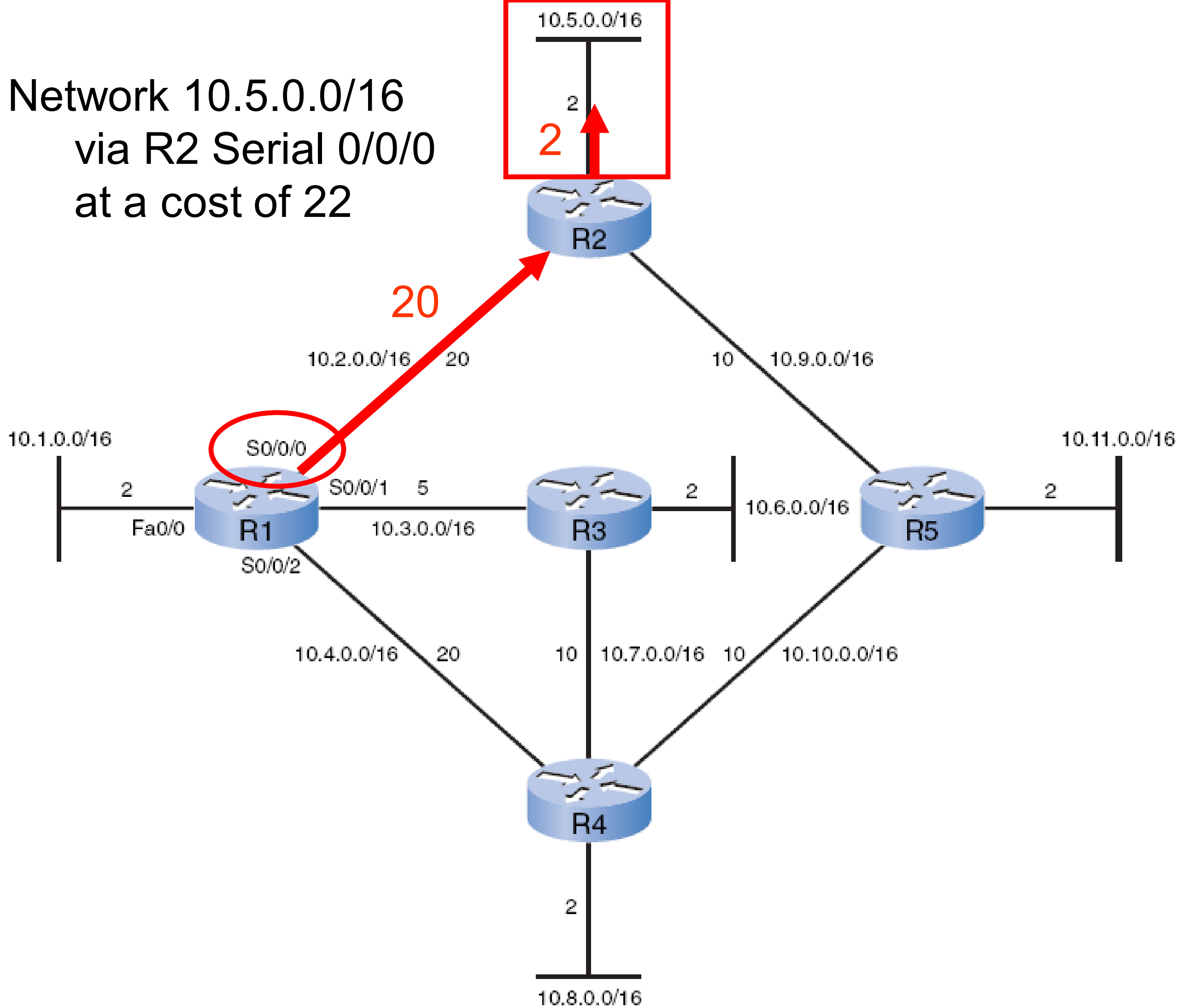
# Determining the Shortest Path



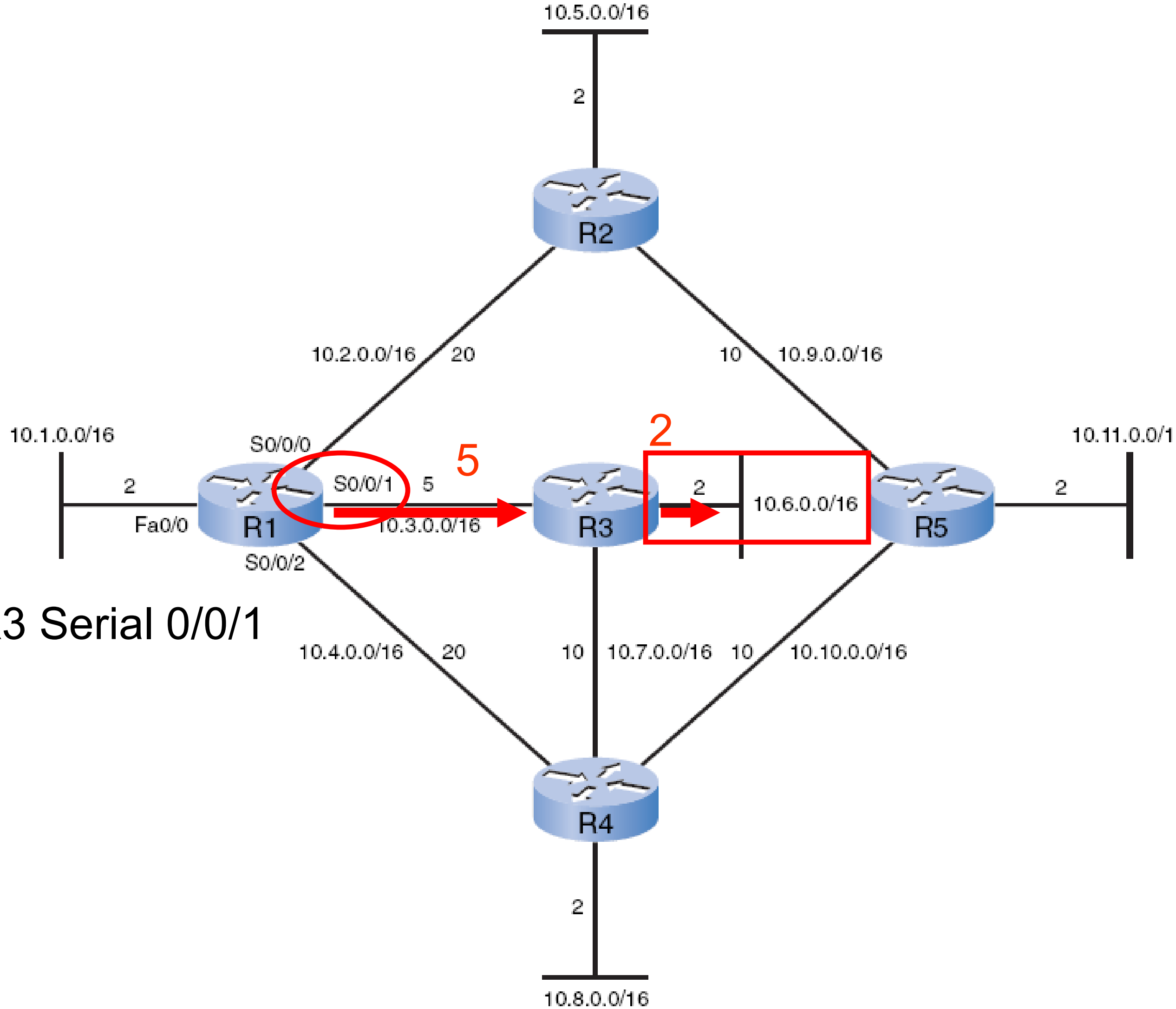

- Using the SPF tree, SPF algorithm results in the shortest path to each network.
  - Note: Only the LANs are shown in the table, but SPF can also be used to determine the shortest path to each WAN link network.

# Determining the Shortest Path

Network 10.5.0.0/16
via R2 Serial 0/0/0
at a cost of 22

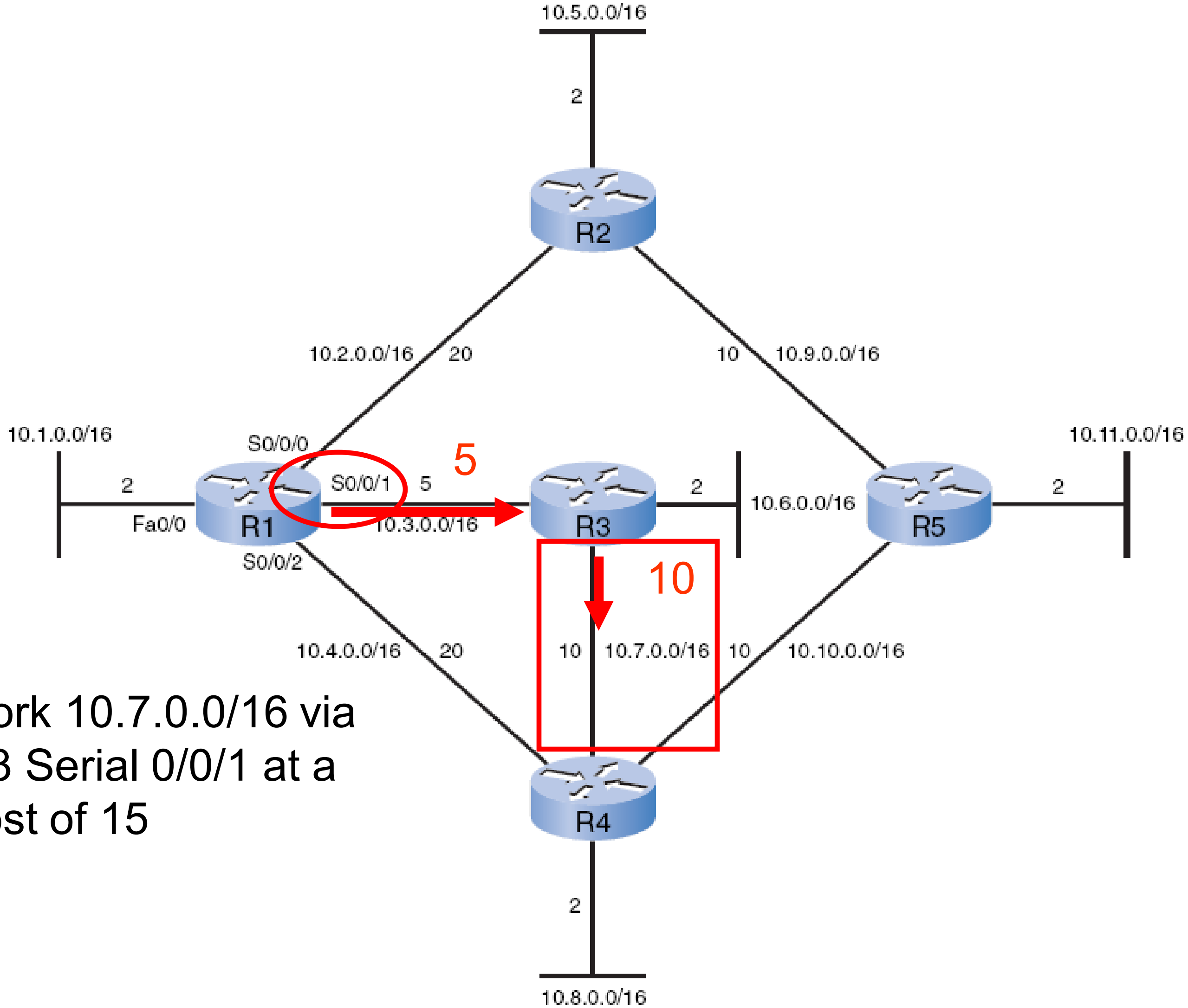


# Determining the Shortest Path



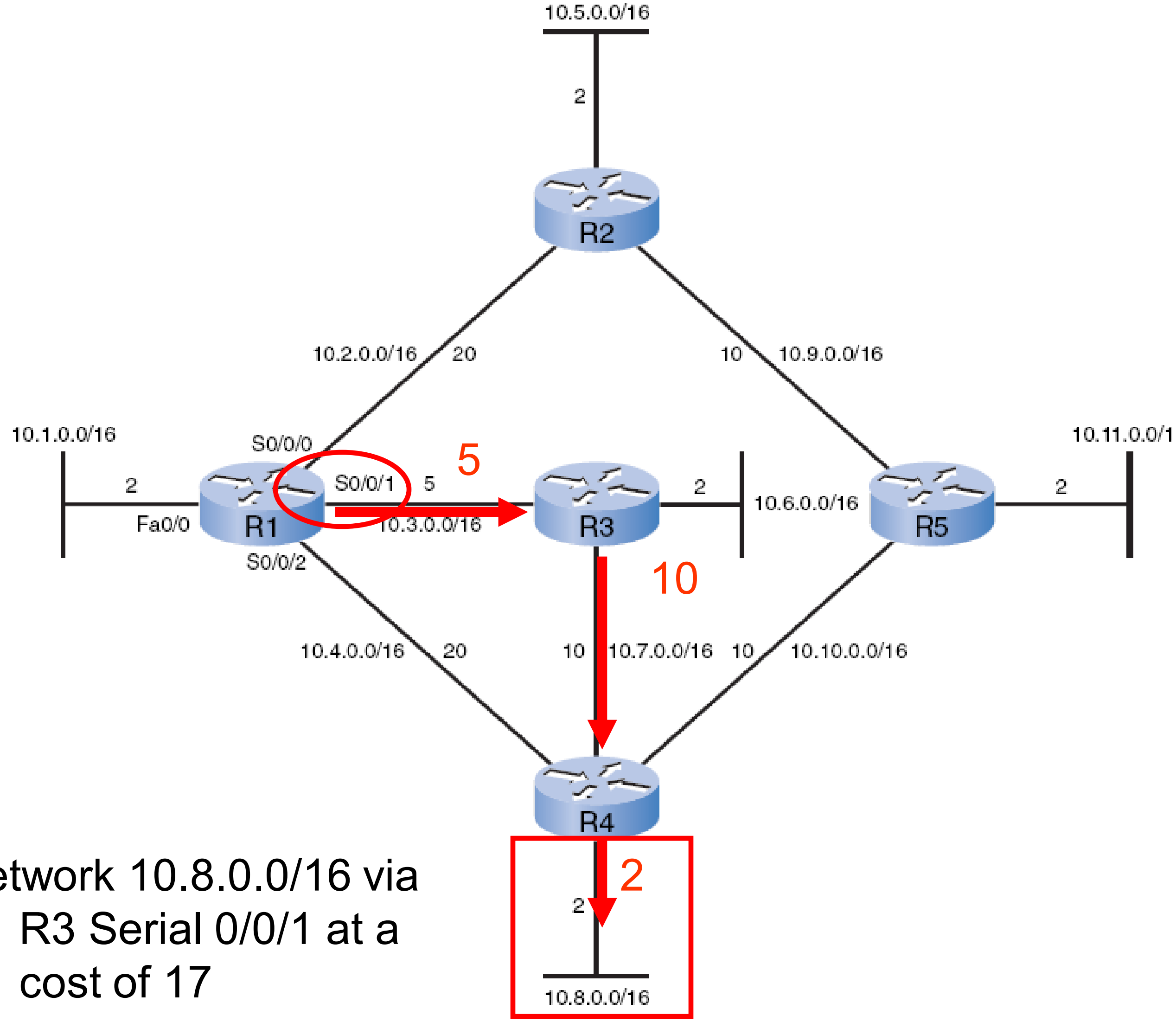

Network 10.6.0.0/16 via R3 Serial 0/0/1 at a cost of 7

# Determining the Shortest Path



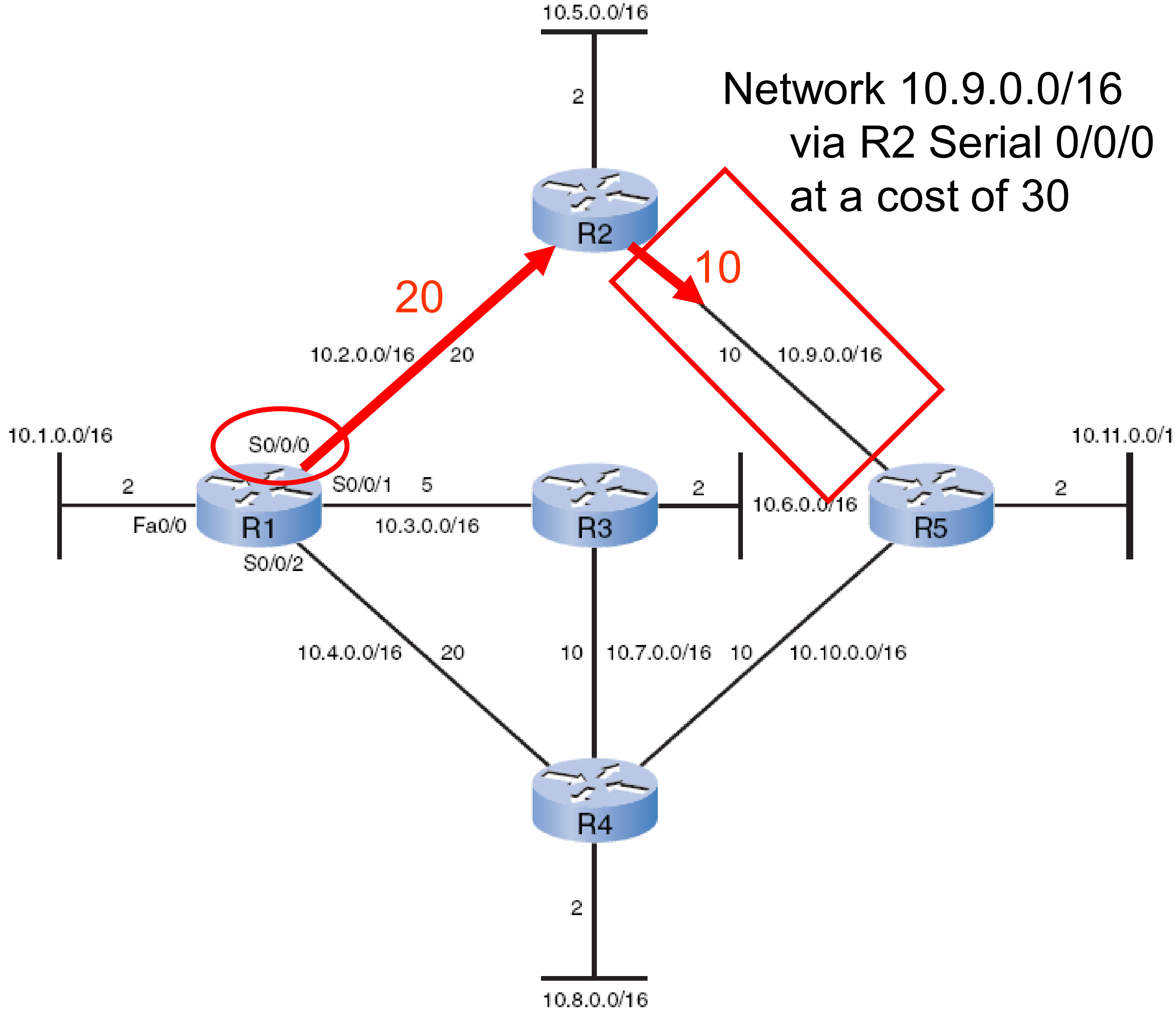

Network 10.7.0.0/16 via
R3 Serial 0/0/1 at a
cost of 15

# Determining the Shortest Path

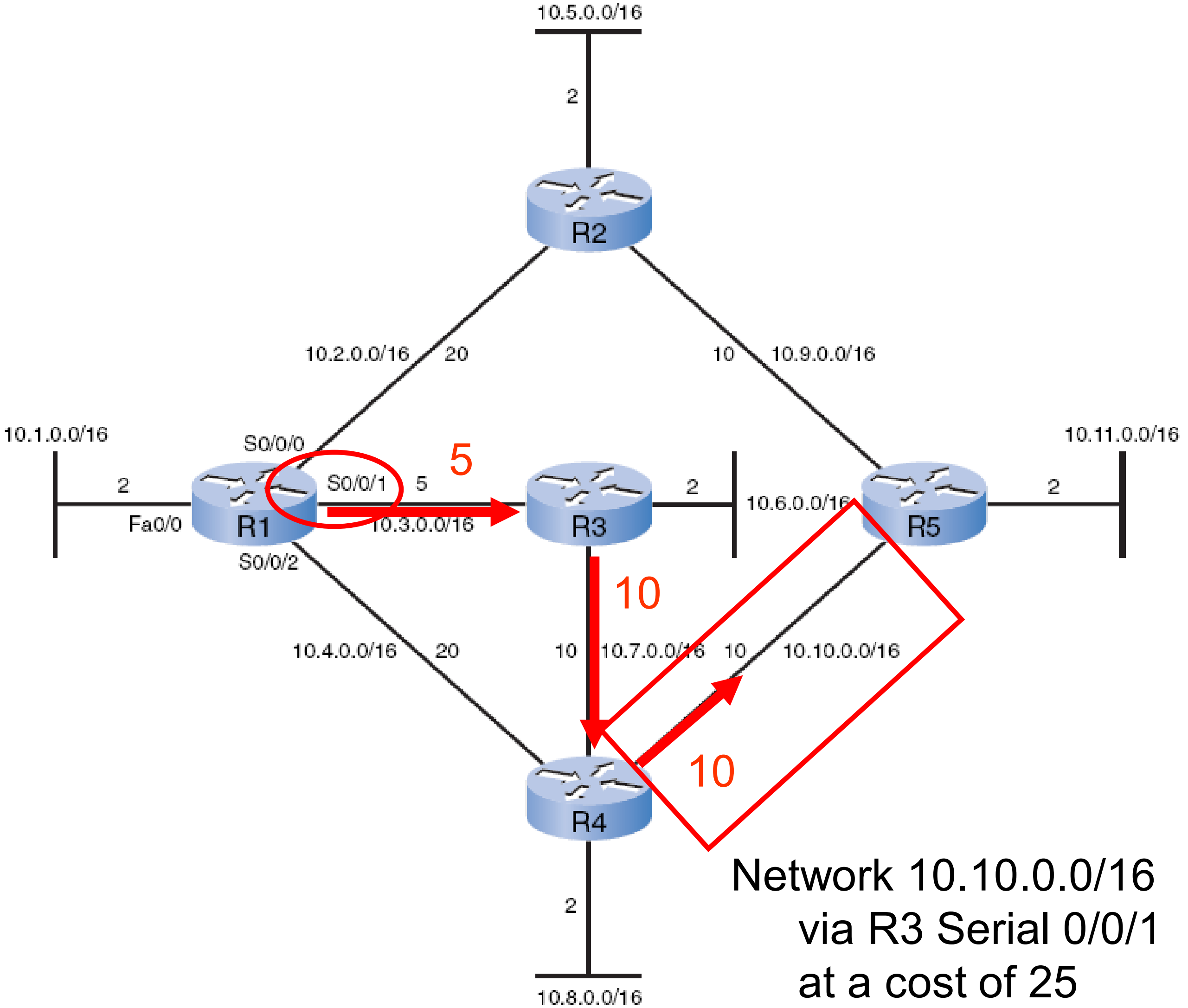



Network 10.8.0.0/16 via R3 Serial 0/0/1 at a cost of 17

# Determining the Shortest Path

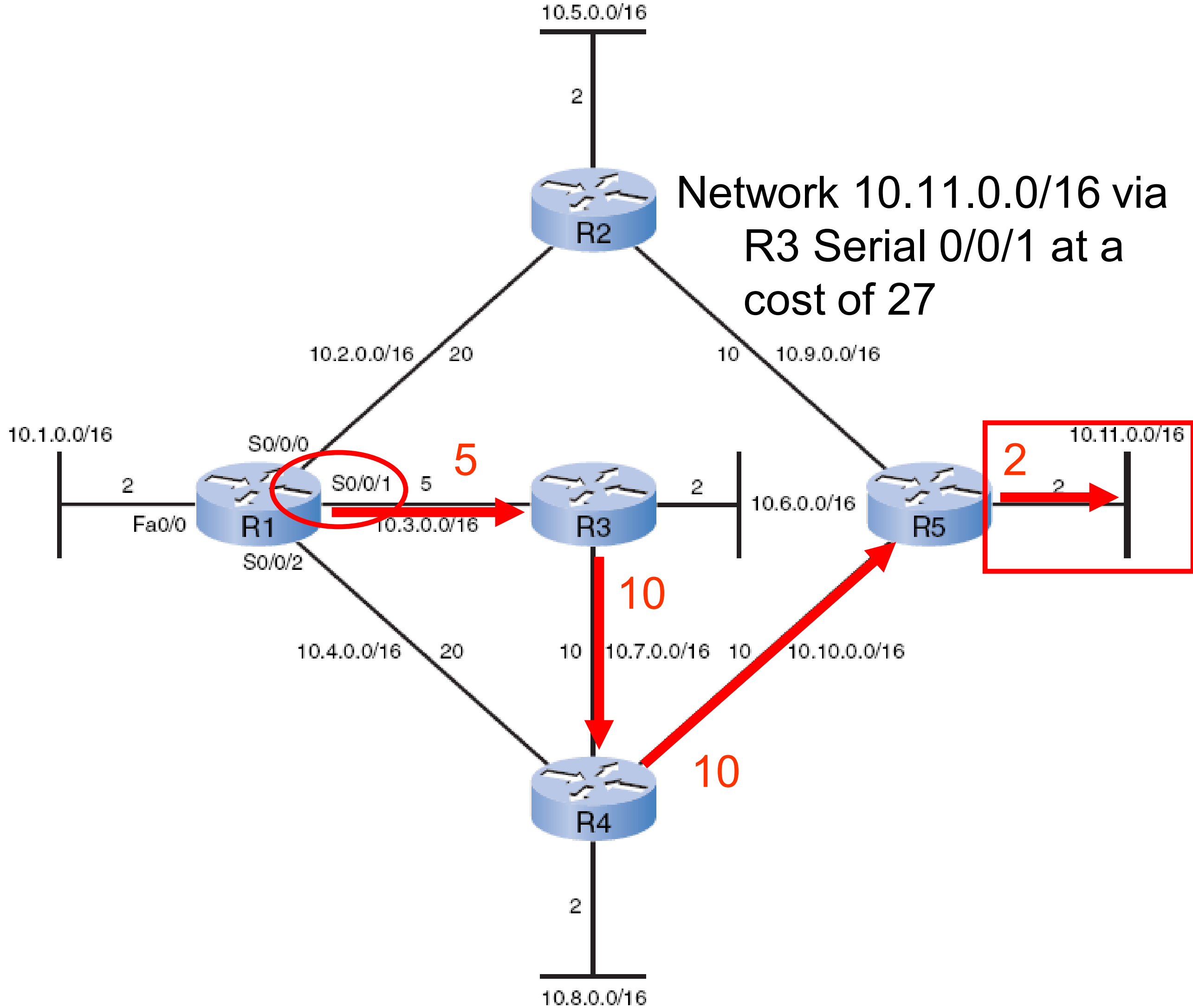


# Determining the Shortest Path

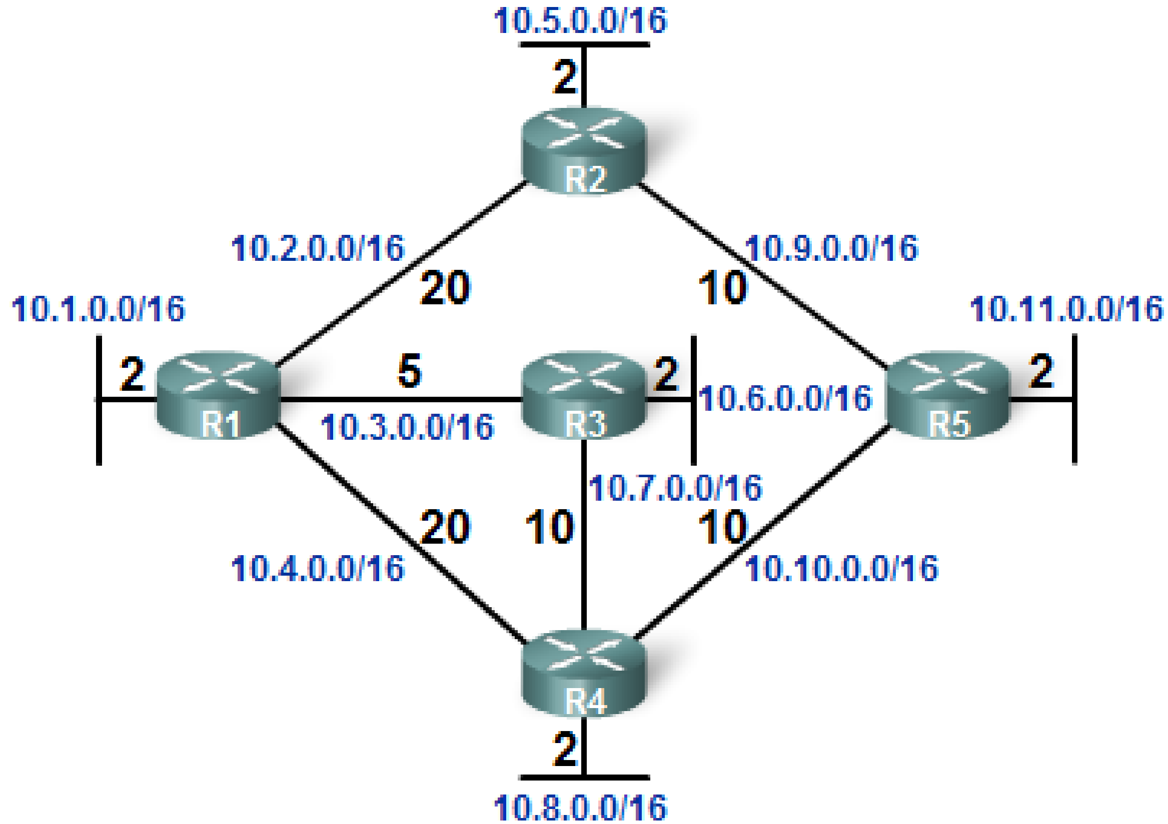



Network 10.10.0.0/16
via R3 Serial 0/0/1
at a cost of 25

# Determining the Shortest Path



# Determining the Shortest Path




- Each router constructs its own SPF tree independently from all other routers.
- Link-state databases must be identical on all routers.

# Generating a Routing Table from the SPF Tree

*SPF Tree for R1*

**SPF Information**

- Network 10.5.0.0/16 via R2 serial 0/0/0 at a cost of 22
- Network 10.6.0.0/16 via R3 serial 0/0/1 at a cost of 7
- Network 10.7.0.0/16 via R3 serial 0/0/1 at a cost of 15
- Network 10.8.0.0/16 via R3 serial 0/0/1 at a cost of 17
- Network 10.9.0.0/16 via R2 serial 0/0/0 at a cost of 30
- Network 10.10.0.0/16 via R3 serial 0/0/1 at a cost of 25
- Network 10.11.0.0/16 via R3 serial 0/0/1 at a cost of 27

**R1 Routing Table**

Directly Connected Networks

- 10.1.0.0/16 Directly Connected Network
- 10.2.0.0/16 Directly Connected Network
- 10.3.0.0/16 Directly Connected Network
- 10.4.0.0/16 Directly Connected Network

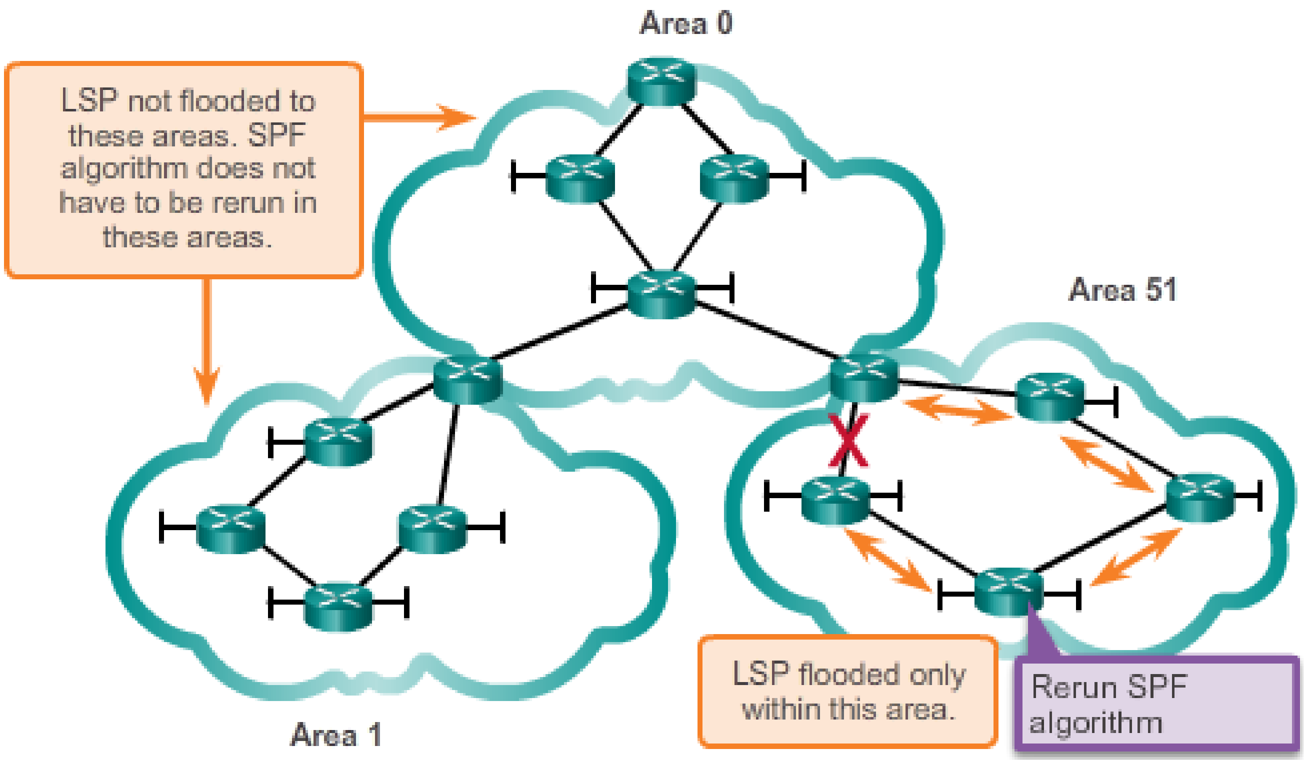
Remote Networks

- 10.5.0.0/16 via R2 serial 0/0/0, cost = 22
- 10.6.0.0/16 via R3 serial 0/0/1, cost = 7
- 10.7.0.0/16 via R3 serial 0/0/1, cost = 15
- 10.8.0.0/16 via R3 serial 0/0/1, cost = 17
- 10.9.0.0/16 via R2 serial 0/0/0, cost = 30
- 10.10.0.0/16 via R3 serial 0/0/1, cost = 25
- 10.11.0.0/16 via R3 serial 0/0/1, cost = 27

- These paths listed previously can now be added to the routing table.
- The routing table will also include
  - Directly connected networks
  - Routes from any other sources, such as static routes.
- Packets will now be forwarded according to these entries in the routing table.

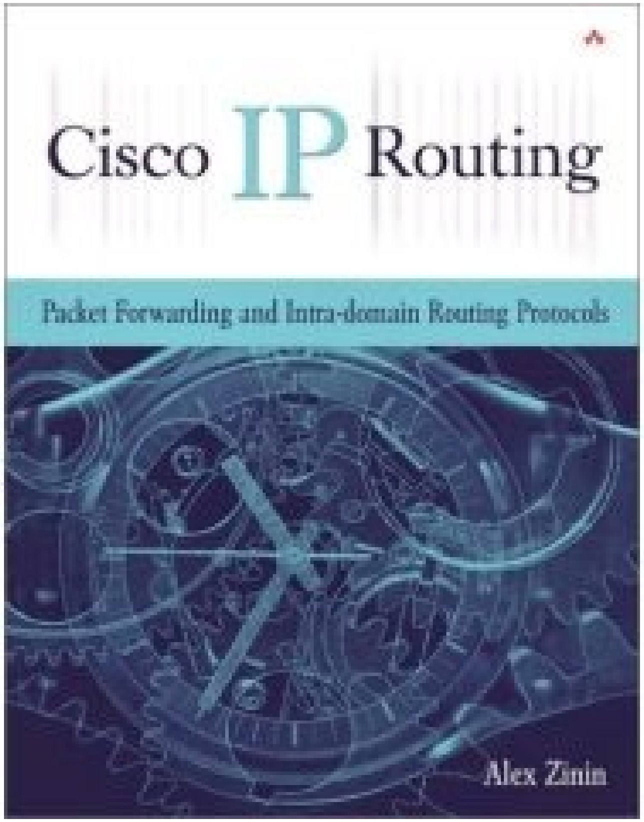
# Addressing Disadvantages

- Create areas to minimize the router memory requirements, processing requirements, and bandwidth requirenments.



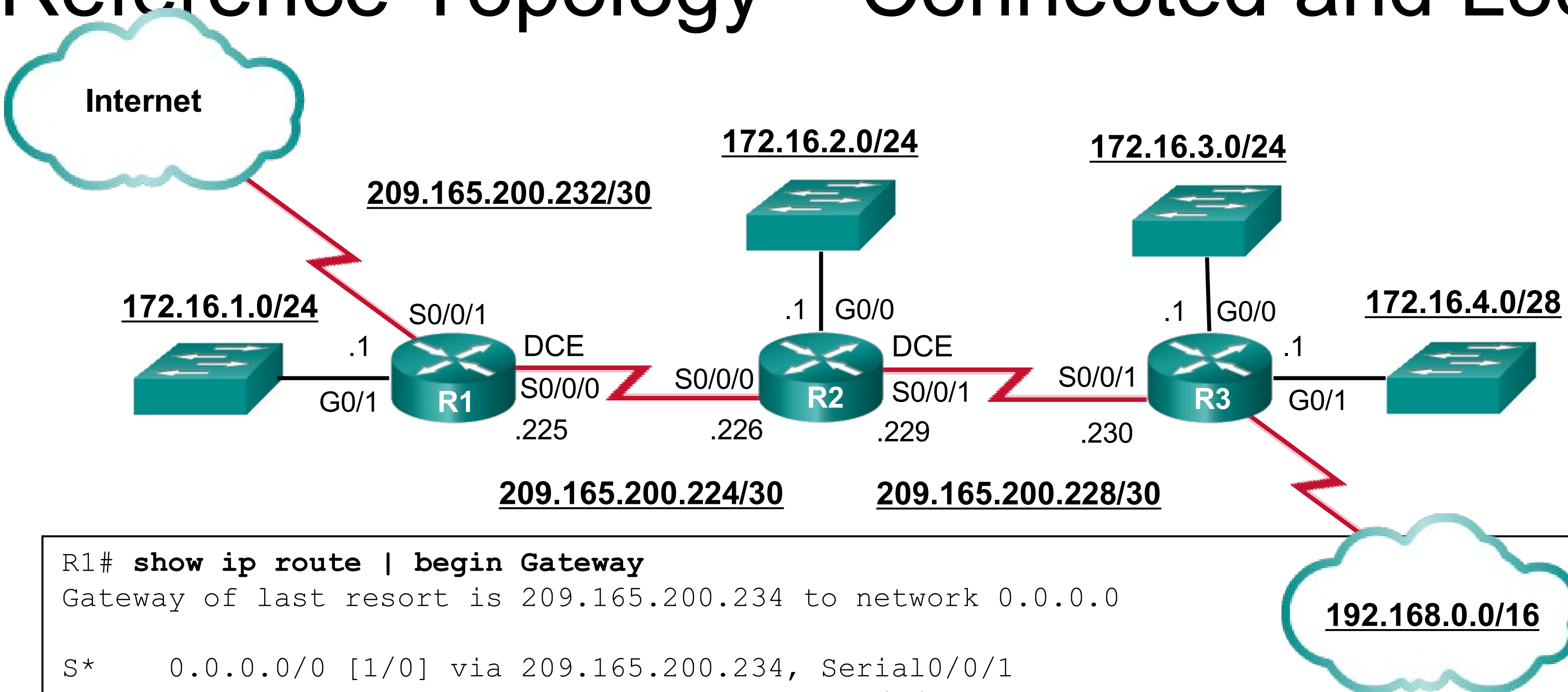
# Routing Table

# Understanding the Routing Table



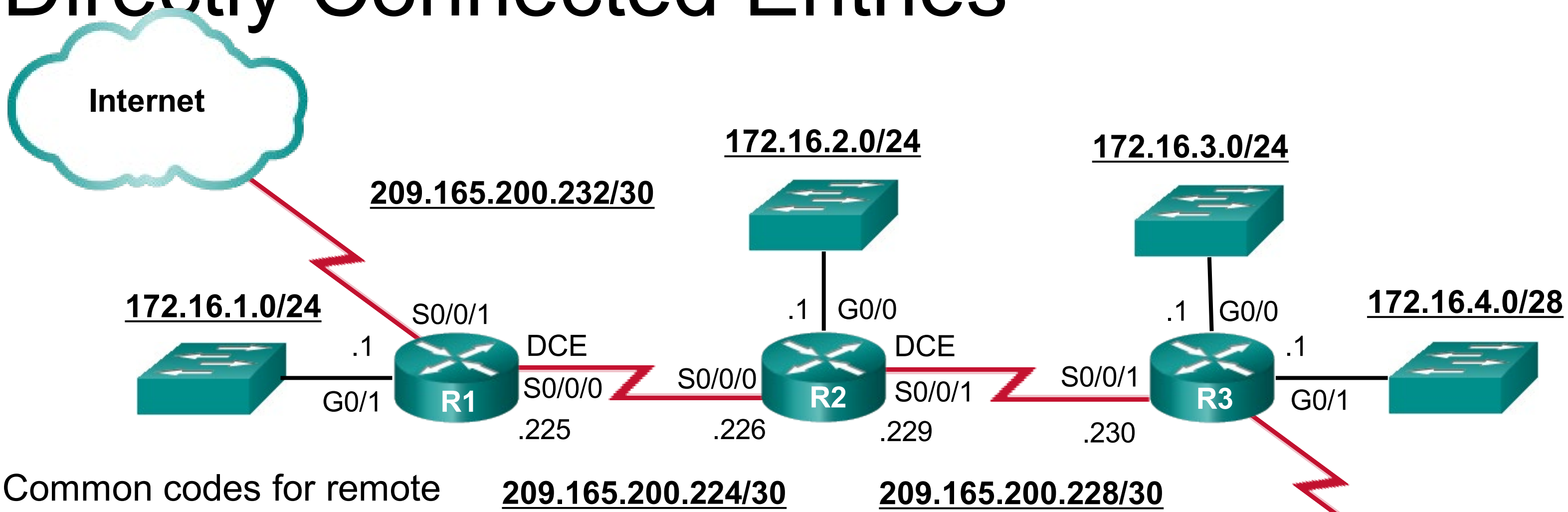

- For more details: ***Cisco IP Routing*, by Alex Zinin** (ISBN 0-201-60473-6)

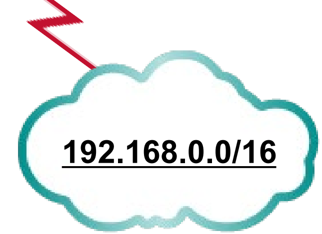
# Reference Topology – Connected and Local




```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C        209.165.200.224/30 is directly connected, Serial0/0/0
L        209.165.200.225/32 is directly connected, Serial0/0/0
R        209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
C        209.165.200.232/30 is directly connected, Serial0/0/1
L        209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

# Directly Connected Entries



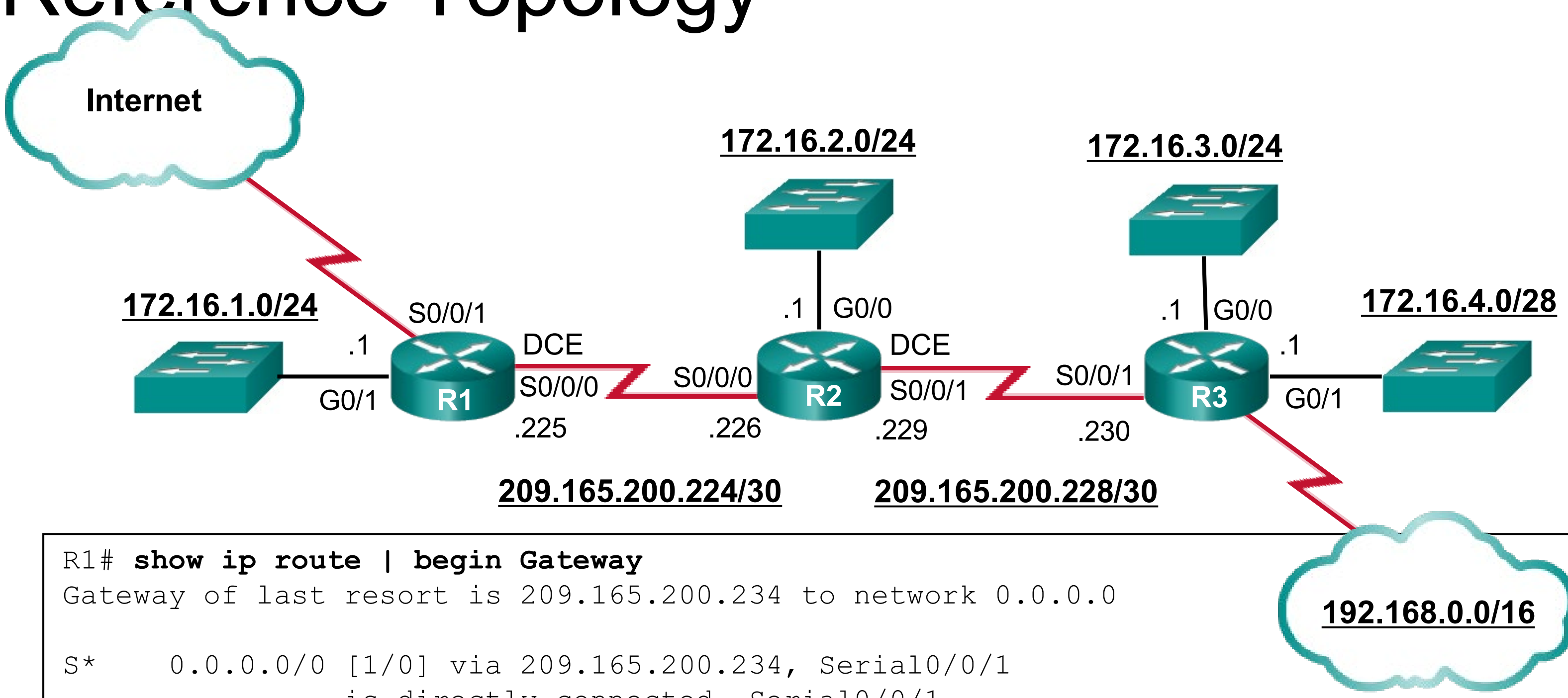

Common codes for remote networks include:

- **S** - Static route
- **D** - EIGRP routing protocol
- **O** - OSPF routing protocol
- **R** - RIP routing protocol

```
C   172.16.1.0/24 is directly connected, GigabitEthernet0/0
L   172.16.1.1/32 is directly connected, GigabitEthernet0/0
```

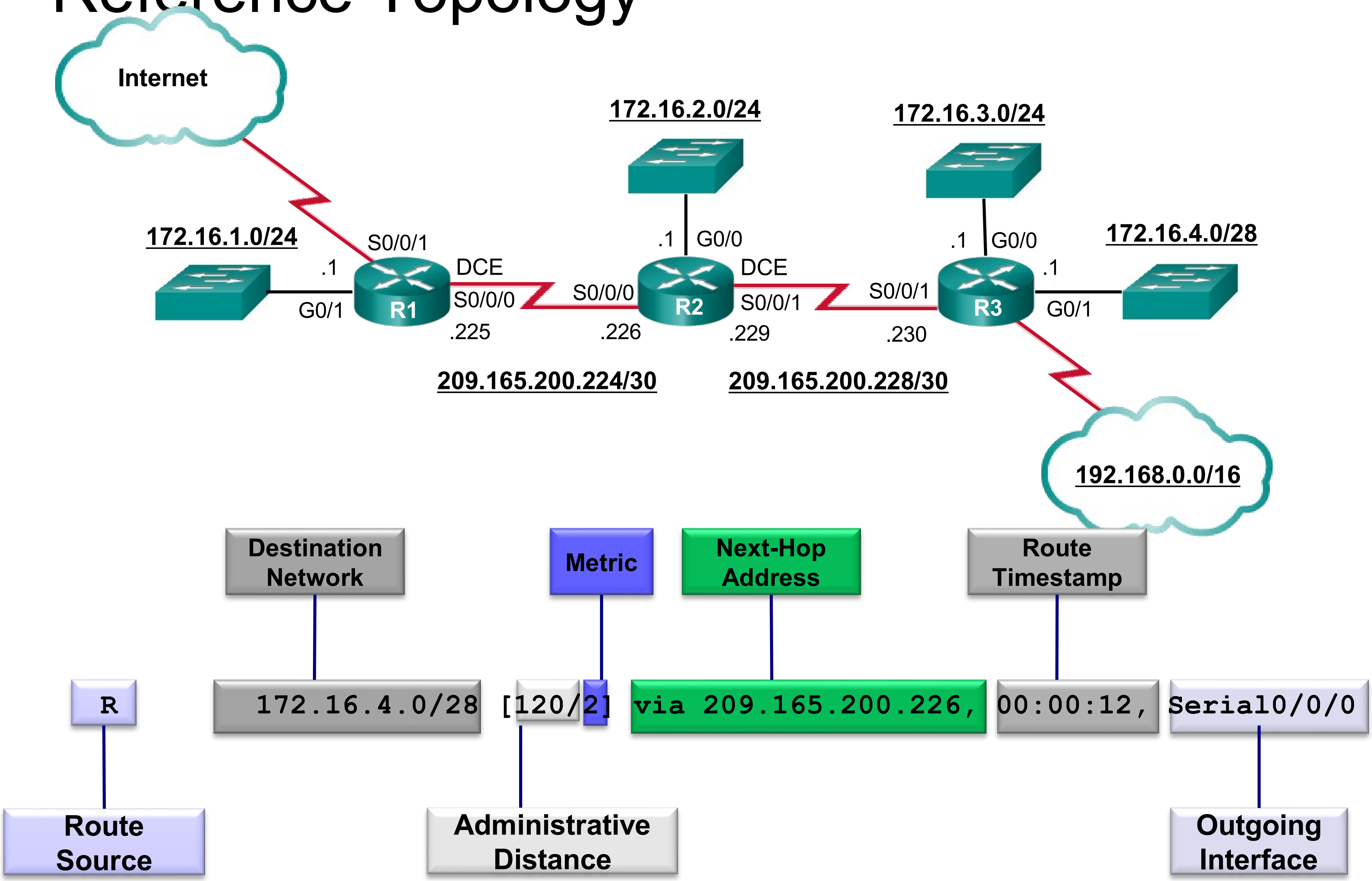
**Route Source** | **Destination Network** | **Outgoing Interface**

# Reference Topology




```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C        209.165.200.224/30 is directly connected, Serial0/0/0
L        209.165.200.225/32 is directly connected, Serial0/0/0
R        209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
C        209.165.200.232/30 is directly connected, Serial0/0/1
L        209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

# Reference Topology



IPv4 Routing Table:

Ultimate route
Level 1 route
Level 1 parent route
Level 2 child routes

# Ultimate Route

- An ultimate route is a routing table entry that contains either:
  - a next-hop IP address or
  - an exit interface
  - This can be directly connected, dynamic, static, and link local routes are ultimate routes.

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C        209.165.200.224/30 is directly connected, Serial0/0/0
L        209.165.200.225/32 is directly connected, Serial0/0/0
R        209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
C        209.165.200.232/30 is directly connected, Serial0/0/1
L        209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

# Level 1 Routes

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

C    192.168.1.0/24 is directly connected, Serial0/0/1
```

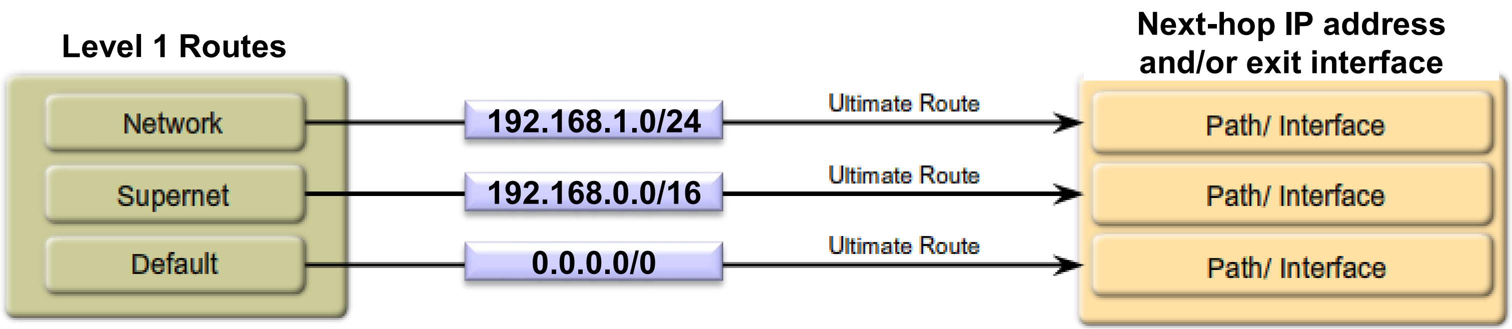
**This is a Level 1 Route**

**Network Address is a Class C Address**

**Subnet Mask is equal to Class C Mask**

- A ***level 1 route*** is a route with a **subnet mask equal to or less than the classful mask of the network address**.

# Level 1 Route




- **level 1 route** can function as any of the following:
  - **Default route:**
    - A default route is a static route with the address 0.0.0.0/0.
  - **Supernet route:**
    - Route with Network address with a mask less than the classful mask.
  - **Network route:**
    - Route is a route that has a subnet mask equal to that of the classful mask.

# Level 1 Routes

```
C        192.168.1.0/24 is directly connected, Serial0/0
```

**Class C Mask** — **Exit Interface**

- Why is directly connected network **192.168.1.0/24** a:
  - **Level 1 network route?**
    - Subnet mask that is the same as its classful mask.
  - **Ultimate route?**
    - Contains the exit interface Serial 0/0/1.

# Level 1 Parent Route

- A level 1 parent route is a level 1 route that is subnetted.
  - A **level 1 parent route** is automatically created any time a subnet is added to the routing table.
  - A parent route can never be an ultimate route.
  - Each entry displays the classful network address, the number of subnets and the number of different subnet masks that the classful address has been subdivided into.

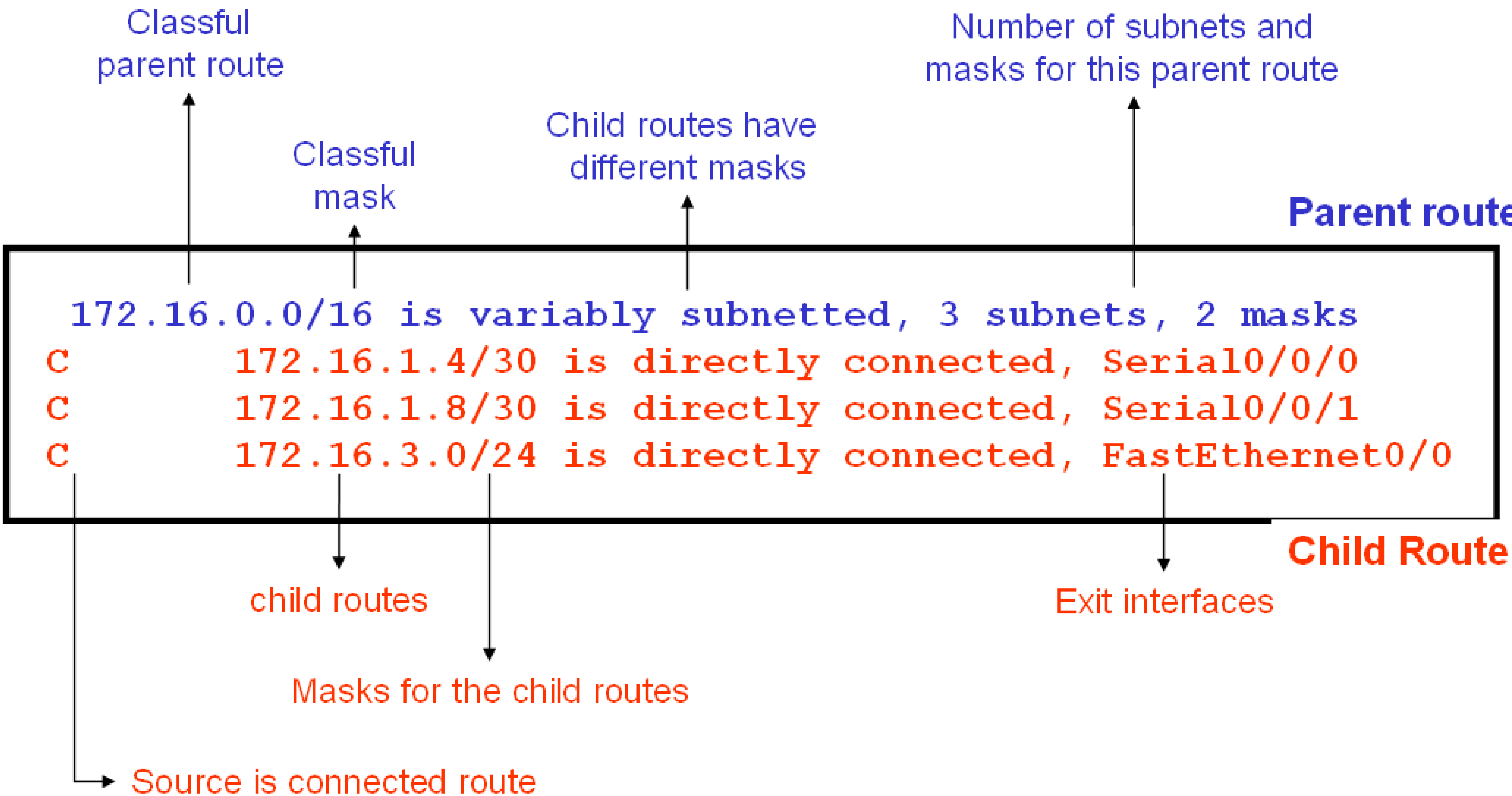
```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*     0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                 is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
```

# Level 2 Child Route

- A **level 2 child route** is a route that is a subnet of a classful network address.
  - Like a level 1 route, the source of a level 2 route can be a directly connected network, a static route, or a dynamically learned route. Level 2 child routes are also ultimate routes.

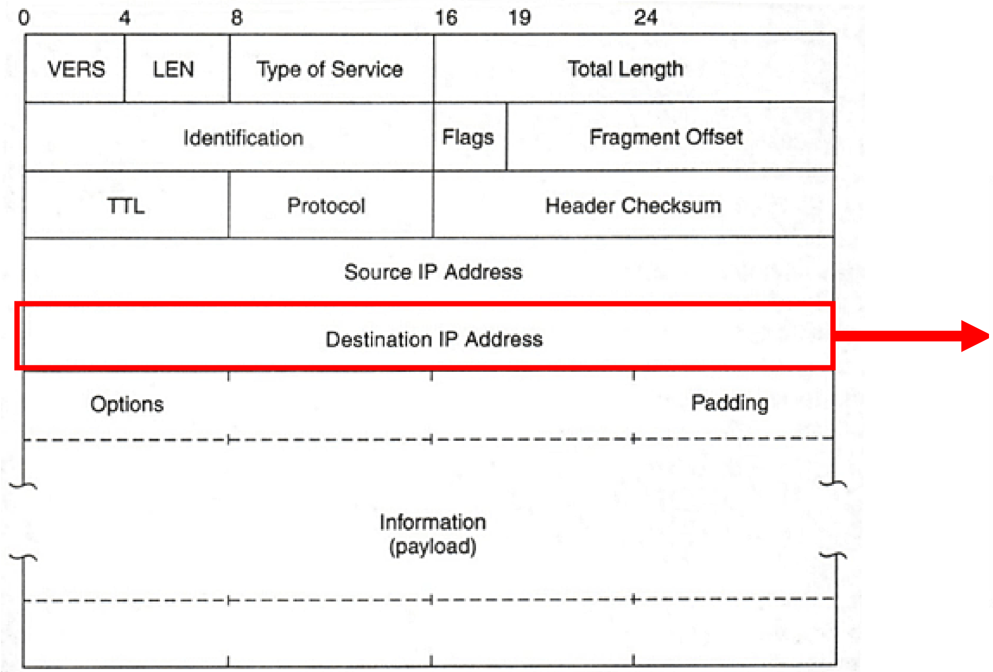
```
R1# show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
```

# Child Routes (Using 12.4 IOS, so no Local Routes)



# Routing Table Lookup Process

## IP Packet




## Routing Table

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.165.200.234 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12, Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C        209.165.200.224/30 is directly connected, Serial0/0/0
L        209.165.200.225/32 is directly connected, Serial0/0/0
R        209.165.200.228/30 [120/1] via 209.165.200.226, 00:00:12, Serial0/0/0
C        209.165.200.232/30 is directly connected, Serial0/0/1
L        209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

**Find “*best match*”**

# Route Lookup Process

1. If the best match is a level 1 ultimate route, then this route is used to forward the packet.
2. If the best match is a level 1 parent route, proceed to the next step.
   - The router examines child routes of the parent route for a best match.
3. If there is a match with a level 2 child route, that subnet is used to forward the packet.
4. If there is no match with any of the level 2 child routes, then the router continues searching level 1 supernet routes in the routing table for a match, including the default route, if there is one.
5. If there is now a lesser match with a level 1 supernet or default routes, the router uses that route to forward the packet.
6. If there is not a match with any route in the routing table, the router drops the packet.

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.16

S*     0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                 is directly connected, Serial0/0/1
       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
R         172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12,Serial0/0/0
R         172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R         172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R      192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
       209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C         209.165.200.224/30 is directly connected, Serial0/0/0
L         209.165.200.225/32 is directly connected, Serial0/0/0
R         209.165.200.228/30 [120/1]via209.165.200.226,0:0:12,Serial0/0/0
C         209.165.200.232/30 is directly connected, Serial0/0/1
L         209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

Destination IP Address **192.168.1.100**

- Packet with **Destination IPv4 address 192.168.1.100**
  - One Level 1 Ultimate Routes to choose from.
  - 192.168.0.0/16 is the best (only) match, the longest match (coming).
    - **Longest match:** Most left-most bits that match between destination IP address and route in routing table
    - At a minimum it must match the number of bits in the route as indicated by the subnet mask.

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.16

S*     0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                 is directly connected, Serial0/0/1
       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
R         172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12,Serial0/0/0
R         172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R         172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R      192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
       209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C         209.165.200.224/30 is directly connected, Serial0/0/0
L         209.165.200.225/32 is directly connected, Serial0/0/0
R         209.165.200.228/30 [120/1]via209.165.200.226,0:0:12,Serial0/0/0
C         209.165.200.232/30 is directly connected, Serial0/0/1
L         209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

Destination IP Address **172.16.3.10**

- Packet with **Destination IPv4 address 172.16.3.10**
  - No Level 1 Ultimate Routes match
  - Matches Level 1 Parent Route (minimum match is classful mask).
    - Look for matching Child route
    - At a minimum it must match the number of bits in the route as indicated by the subnet mask.
    - If more than one child route matches choose routes with longest match.

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.16

S*     0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                 is directly connected, Serial0/0/1
       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
R         172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12,Serial0/0/0
R         172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R         172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R      192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
       209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C         209.165.200.224/30 is directly connected, Serial0/0/0
L         209.165.200.225/32 is directly connected, Serial0/0/0
R         209.165.200.228/30 [120/1]via209.165.200.226,0:0:12,Serial0/0/0
C         209.165.200.232/30 is directly connected, Serial0/0/1
L         209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

Destination IP Address **10.1.2.3**

- Packet with **Destination IPv4 address 10.1.2.3**
  - No Level 1 Ultimate Routes to choose match
  - No Level 1 Parent Routes match
  - Look for supernet route (mask less than classful mask), including default route
    - Choose longest match.
    - Default route always matches because 0 bits have to match.
    - 0 matching bits makes it the “gateway of last resort”

```
R1#show ip route | begin Gateway
```

Destination IP Address **10.1.2.3**

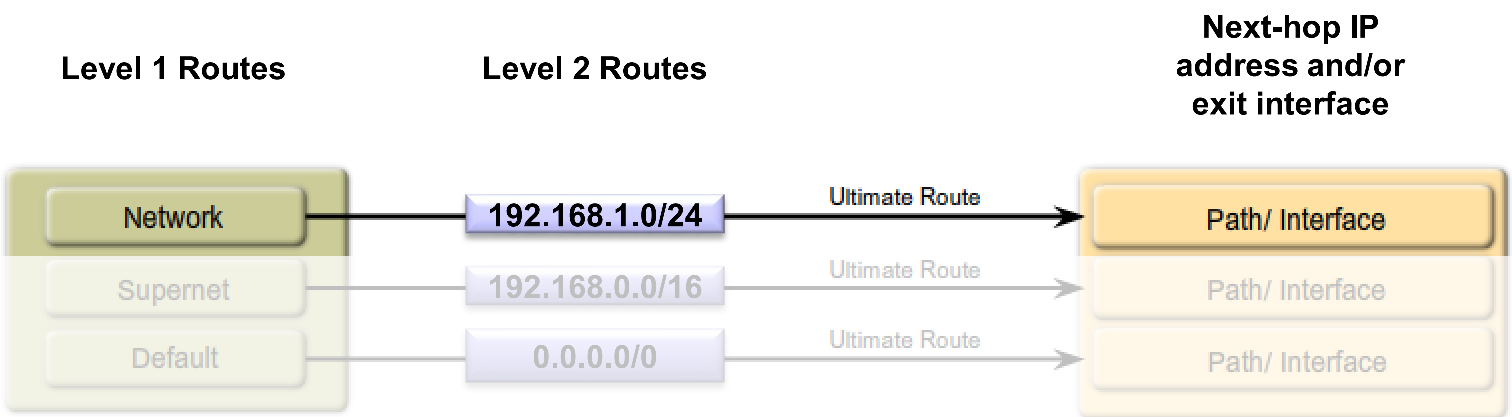
```
      172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
C        172.16.1.0/24 is directly connected, GigabitEthernet0/0
L        172.16.1.1/32 is directly connected, GigabitEthernet0/0
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12,Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C        209.165.200.224/30 is directly connected, Serial0/0/0
L        209.165.200.225/32 is directly connected, Serial0/0/0
R        209.165.200.228/30 [120/1]via209.165.200.226,0:0:12,Serial0/0/0
C        209.165.200.232/30 is directly connected, Serial0/0/1
L        209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

- Packet with **Destination IPv4 address 10.1.2.3**
  - No default route
  - No Level 1 Ultimate Routes to choose match
  - No Level 1 Parent Routes match
  - No supernet routes match
  - Drop packet

# Route Lookup Process (FYI)

1. If the best match is a level 1 ultimate route, then this route is used to forward the packet.

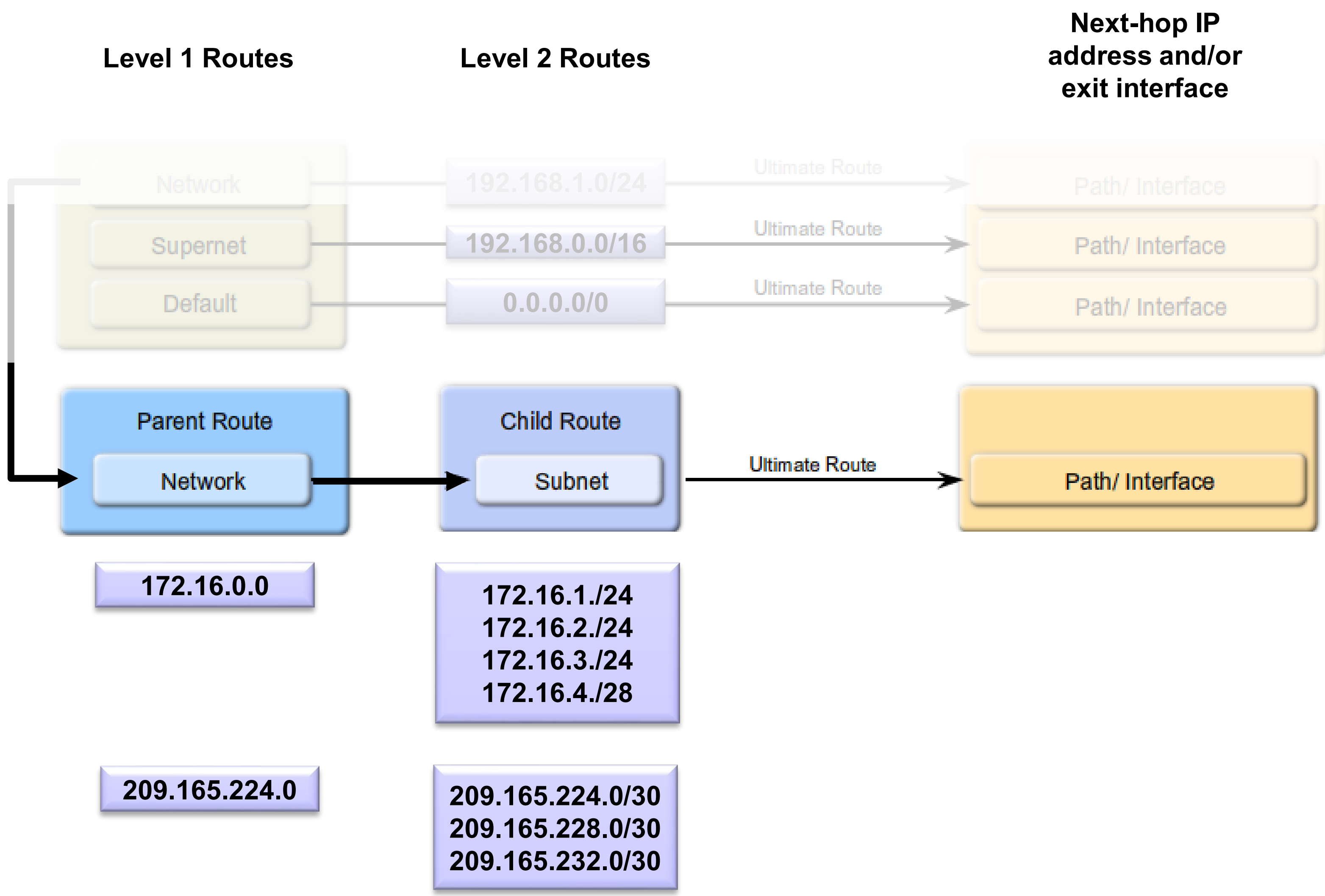
# Match Level 1 Routes (FYI)



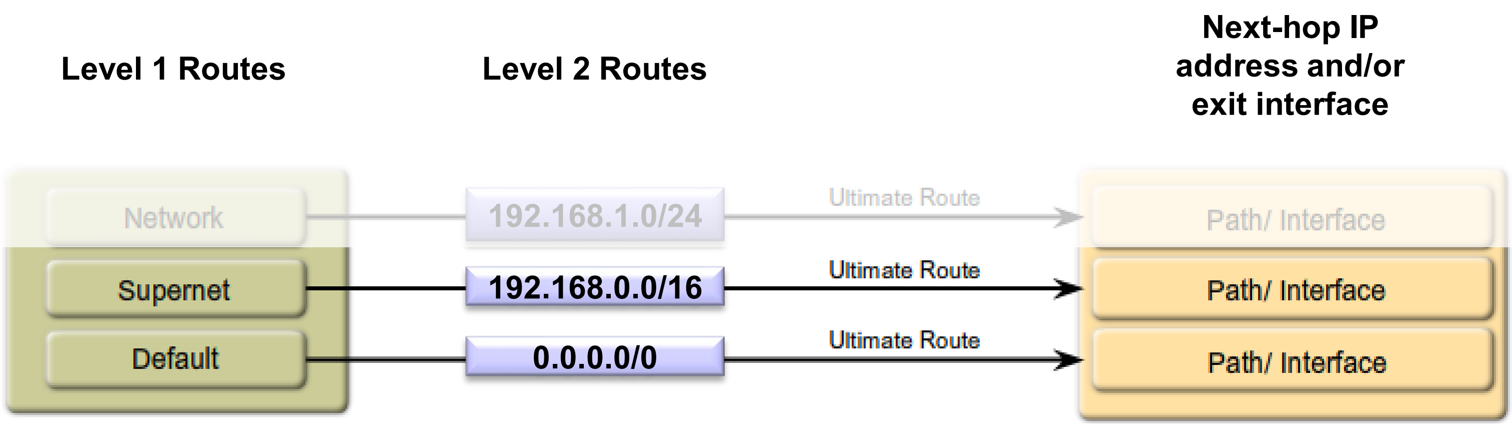
# Route Lookup Process (FYI)

1. If the best match is a level 1 ultimate route, then this route is used to forward the packet.
2. If the best match is a level 1 parent route then the router examines child routes (the subnet routes) of the parent route for a best match.
3. If there is a match with a level 2 child route, that subnet is used to forward the packet.

# Match Level 2 Child Routes (FYI)



# Match Supernet and Then Default Route (FYI)



# Best Route = Longest Match (FYI)

- The best match is the route in the routing table that has the most number of far left matching bits with the destination IPv4 address of the packet.

- The route with the greatest number of equivalent far left bits, or the longest match, is always the preferred route.

# A packet is destined for 172.16.0.10

- The router has three possible routes that match this packet:
  - 172.16.0.0/12
  - 172.16.0.0/18
  - 172.16.0.0/26.

| | | |
|---|---|---|
| IP Packet Destination | 172.16.0.10 | 10101100.00010000.00000000.00001010 |

| | | |
|---|---|---|
| Route 1 | 172.16.0.0/12 | 10101100.00010000.00000000.00000000 |
| Route 2 | 172.16.0.0/18 | 10101100.00010000.00000000.00000000 |
| Route 3 | 172.16.0.0/26 | 10101100.00010000.00000000.00000000 |

Longest Match to IP Packet Destination

- Of the three routes, 172.16.0.0/26 has the longest match and is therefore chosen to forward the packet.

Destination IP Address **172.16.3.10**

```
R1#show ip route | begin Gateway
Gateway of last resort is 209.16

S*    0.0.0.0/0 [1/0] via 209.165.200.234, Serial0/0/1
                is directly connected, Serial0/0/1
      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
R        172.16.2.0/24 [120/1] via 209.165.200.226, 00:00:12,Serial0/0/0
R        172.16.3.0/24 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R        172.16.4.0/28 [120/2] via 209.165.200.226, 00:00:12,Serial0/0/0
R     192.168.0.0/16 [120/2] via 209.165.200.226, 00:00:03, Serial0/0/0
      209.165.200.0/24 is variably subnetted, 5 subnets, 2 masks
C        209.165.200.224/30 is directly connected, Serial0/0/0
L        209.165.200.225/32 is directly connected, Serial0/0/0
R        209.165.200.228/30 [120/1]via209.165.200.226,0:0:12,Serial0/0/0
C        209.165.200.232/30 is directly connected, Serial0/0/1
L        209.165.200.233/30 is directly connected, Serial0/0/1
R1#
```

# Longest Match Example

Destination IP Address **172.16.3.10**

| | | |
|---|---|---|
| Destination of IP Packet | 172.16.3.10 | 10101100 00010000 00000011 00001010 |
| Level 1 Parent Route | 172.16.0.0/16 | 10101100 00010000 00000000 00000000 |
| Level 2 Child Route | 172.16.1.0/24 | 10101100 00010000 00000001 00000000 |
| Level 2 Child Route | 172.16.2.0/24 | 10101100 00010000 00000010 00000000 |
| Level 2 Child Route | 172.16.3.0/24 | 10101100 00010000 00000011 00000000 |

Matches Parent Route.

# Longest Match Example

Destination IP Address **172.16.3.10**

**23rd bit does not match. First 24 bits must match. Router skips this route and moves to the next route entry.**

| | | |
|---|---|---|
| Destination of IP Packet | 172.16.3.10 | 10101100 00010000 00000011 00001010 |
| Level 1 Parent Route | 172.16.0.0/16 | 10101100 00010000 00000000 00000000 |
| Level 2 Child Route | 172.16.1.0/24 | 10101100 00010000 00000001 00000000 |
| Level 2 Child Route | 172.16.2.0/24 | 10101100 00010000 00000010 00000000 |
| Level 2 Child Route | 172.16.3.0/24 | 10101100 00010000 00000011 00000000 |

**First 22 bits match.**

# Longest Match Example

Destination IP Address **172.16.3.10**

**24th** bit does not match. First 24 bits must match. Router skips this route and moves to the next route entry.

| | | |
|---|---|---|
| Destination of IP Packet | 172.16.3.10 | 10101100 00010000 00000011 00001010 |
| Level 1 Parent Route | 172.16.0.0/16 | 10101100 00010000 00000000 00000000 |
| Level 2 Child Route | 172.16.1.0/24 | 10101100 00010000 00000001 00000000 |
| Level 2 Child Route | 172.16.2.0/24 | 10101100 00010000 00000010 00000000 |
| Level 2 Child Route | 172.16.3.0/24 | 10101100 00010000 00000011 00000000 |

First 23 bits match.

# Longest Match Example

Destination IP Address 172.16.3.10

| | | |
|---|---|---|
| Destination of IP Packet | 172.16.3.10 | 10101100 00010000 00000011 00001010 |
| Level 1 Parent Route | 172.16.0.0/16 | 10101100 00010000 00000000 00000000 |
| Level 2 Child Route | 172.16.1.0/24 | 10101100 00010000 00000001 00000000 |
| Level 2 Child Route | 172.16.2.0/24 | 10101100 00010000 00000010 00000000 |
| Level 2 Child Route | 172.16.3.0/24 | 10101100 00010000 00000011 00000000 |

First 24 bits match.

# Question

- Assume that a router has four routing processes running on it, and each process has received these routes:
  - EIGRP (internal): 192.168.32.0/26
  - RIP: 192.168.32.0/24
  - OSPF: 192.168.32.0/19

- Which of these routes will be installed in the routing table?

- **All of them! Different masks so different routes.**

# Longest Match

- Although EIGRP has the best administrative distance, each of these routes has a different prefix length (subnet mask).
  - They are therefore considered different destinations and are all installed in the routing table.

```
router# show ip route
     ....
     D   192.168.32.0/26 [90/25789217] via 10.1.1.1
     R   192.168.32.0/24 [120/4] via 10.1.1.2
     O   192.168.32.0/19 [110/229840] via 10.1.1.3
     ....
```

# Longest Match

- Longer prefixes are always preferred over shorter ones when forwarding a packet.

```
router# show ip route
      ....
      D   192.168.32.0/26 [90/25789217] via 10.1.1.1
      R   192.168.32.0/24 [120/4] via 10.1.1.2
      O   192.168.32.0/19 [110/229840] via 10.1.1.3
      ....
```

# Longest Match Quiz #1

- If a packet arrives on a router interface destined for 192.168.32.1, which route would the router choose?

```
router# show ip route
     ....
     D    192.168.32.0/26 [90/25789217] via 10.1.1.1
     R    192.168.32.0/24 [120/4] via 10.1.1.2
     O    192.168.32.0/19 [110/229840] via 10.1.1.3
     ....
```

# Longest Match Quiz #1

- Because 192.168.32.1 falls within network 192.168.32.0/26.
  - Network 192.168.32.0 to 192.168.32.63.
  - It also falls within the other two routes available, but the 192.168.32.0/26 has the longest prefix.

```
router# show ip route
     ....
     D   192.168.32.0/26 [90/25789217] via 10.1.1.1
     R   192.168.32.0/24 [120/4] via 10.1.1.2
     O   192.168.32.0/19 [110/229840] via 10.1.1.3
     ....
```

# Longest Match Quiz #2

- If a packet arrives on a router interface destined for 192.168.32.100, which route would the router choose?

```
router# show ip route
      ....
      D    192.168.32.0/26 [90/25789217] via 10.1.1.1
      R    192.168.32.0/24 [120/4] via 10.1.1.2
      O    192.168.32.0/19 [110/229840] via 10.1.1.3
      ....
```

# Longest Match Quiz #2

- It is directed toward 10.1.1.2 because it falls within network 192.168.32.0/24.
  - Network 192.168.32.0 to 192.168.32.254.
  - It also falls within the OSPF route, but the 192.168.32.0/24 has the longest prefix than /19.

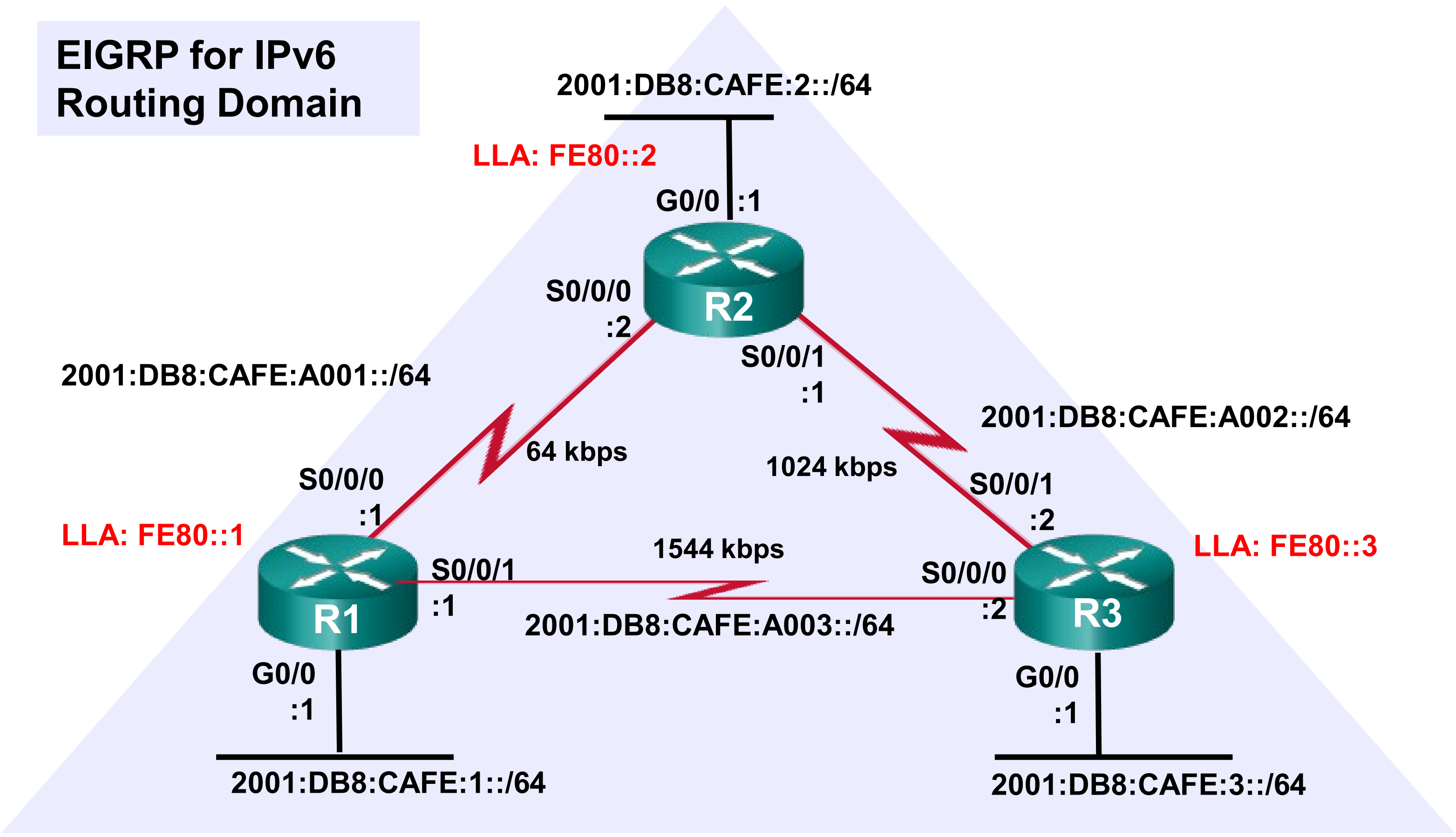
```
router# show ip route
     ....
     D   192.168.32.0/26 [90/25789217] via 10.1.1.1
     R   192.168.32.0/24 [120/4] via 10.1.1.2
     O   192.168.32.0/19 [110/229840] via 10.1.1.3
     ....
```

# IPv6 Routing Tables

```
R1# show ipv6 route

<Output omitted>

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#
```

- Components of the IPv6 routing table are very similar to the IPv4 routing table (directly connected interfaces, static routes, and dynamically learned routes).
- IPv6 is classless by design, all routes are effectively level 1 ultimate routes.
  - There is no level 1 parent of level 2 child routes.

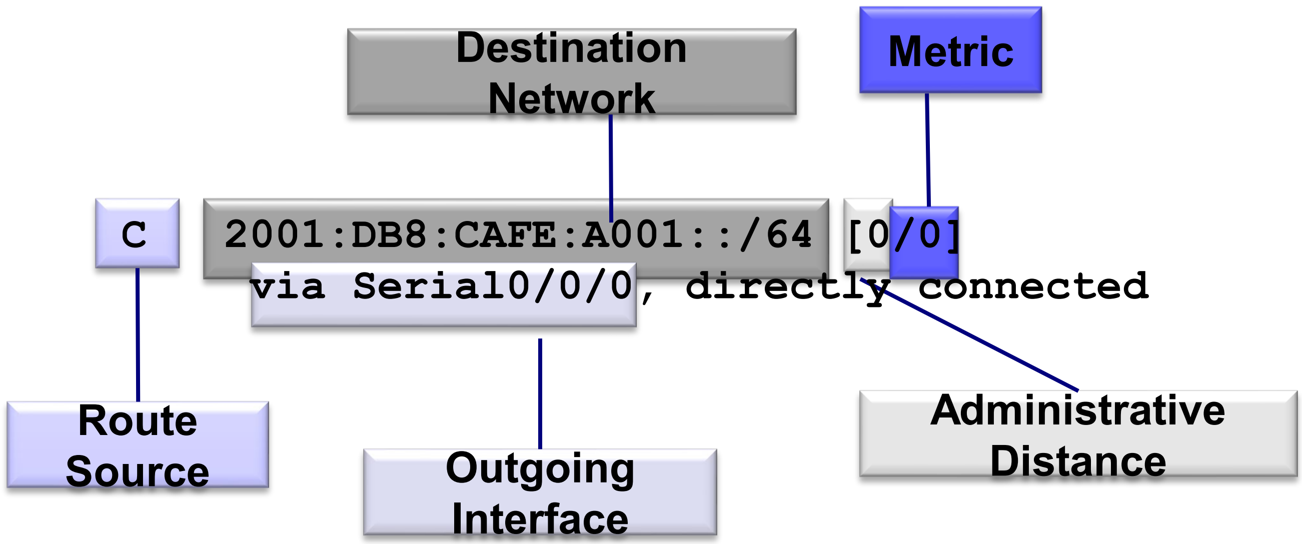
# Reference IPv6 Topology



# Directly Connected Networks of R1

```
R1# show ipv6 route

<Output omitted>

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#
```

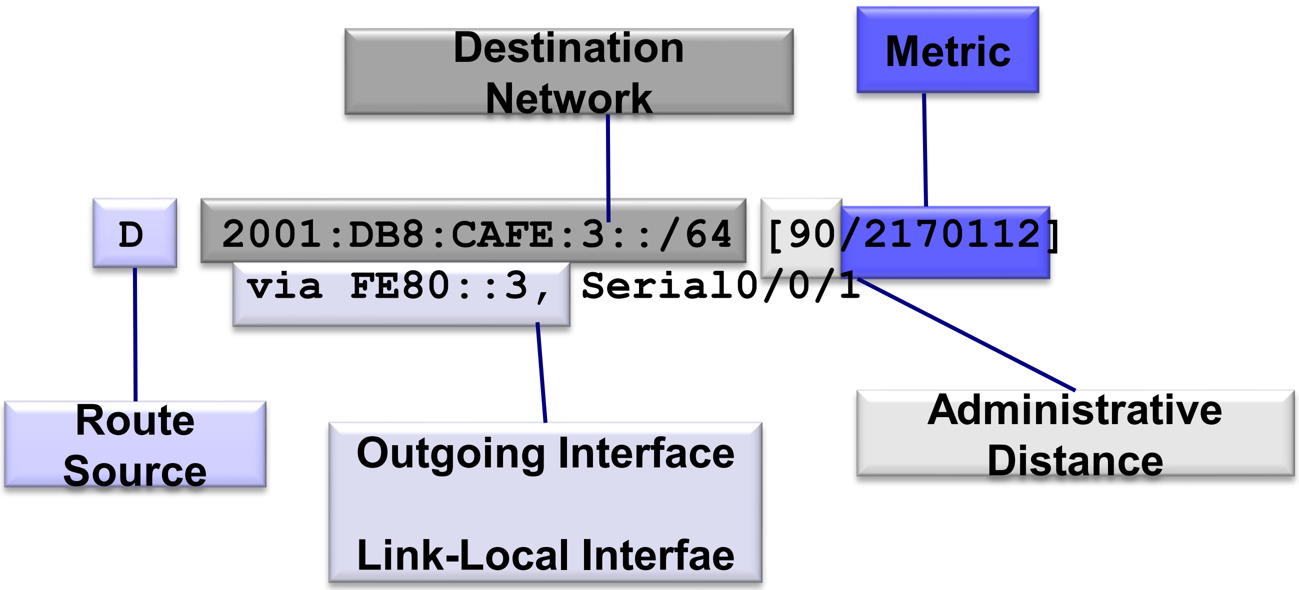
# Directly Connected Networks of R1



# Remote Network Entries on R1

```
R1# show ipv6 route

<Output omitted>

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#
```

# Remote Networks of R1



# Chapter 7: Summary

- Dynamic routing protocols:
- Used by routers to automatically learn about remote networks from other routers
- Purpose includes: discovery of remote networks, maintaining up-to-date routing information, choosing the best path to destination networks, and ability to find a new best path if the current path is no longer available
- Best choice for large networks but static routing is better for stub networks.
- Function to inform other routers about changes
- Can be classified as either classful or classless, distance-vector or link-state, and an interior or an exterior gateway protocol

# Chapter 7: Summary (cont.)

- Dynamic routing protocols:
- A link-state routing protocol can create a complete view or topology of the network by gathering information from all of the other routers
- Metrics are used to determine the best path or shortest path to reach a destination network
- Different routing protocols may use different  (hops, bandwidth, delay, reliability, and load)
- Show ip protocols command displays the IPv4 routing protocol settings currently configured on the router,  for IPv6, use show ipv6 protocols

# Chapter 7: Summary (cont.)

- Dynamic routing protocols:
- Cisco routers use the administrative distance value to determine which routing source to use
- Each dynamic routing protocol has a unique administrative value, along with static routes and directly connected networks, lower is preferred the route
- Directly connected networks are preferred source, followed by static routes and then various dynamic routing protocols
- An OSPF link is an interface on a router, information about the state of the links is known as link-states
- Link-state routing protocols apply Dijkstra's algorithm to calculate the best path route which uses accumulated costs along each path, from source to destination, to determine the total cost of a route