# OSPF Introduction

- Open Shortest Path First (OSPF) is a link-state protocol.
- It's an open standard protocol.
- Compared to EIGRP, OSPF is a more complex protocol and supports all features such as VLSM/CIDR and more.
- Works on the concept of Areas and Autonomous systems
- Highly Scalable
- Supports VLSM/CIDR and dis-contiguous networks
- Does not have a hop count limit
- Works in multivendor environment
- Minimizes updates between neighbors.
- The administrative distance of OSPF routes is, by default, 110.
- Uses multicast addresses 224.0.0.5 and 224.0.0.6 for routing updates.

# OSPF Introduction

| Features | OSPF | EIGRP | RIPv1 | RIPv2 |
|---|---|---|---|---|
| Protocol Type | Link state | Hybrid | Distance Vector | Distance Vector |
| Classful Protocol | No | No | Yes | No |
| VLSM Support | Yes | Yes | No | Yes |
| Discontiguous Network Support | Yes | Yes | No | Yes |
| Hop count limit | None | 255 | 15 | 15 |
| Routing Updates | Event Triggered | Event Triggered | Periodic | Periodic |
| Complete Routing table shared | During new adjacencies | During new adjacencies | Periodic | Periodic |
| Mechanism for sharing updates | Multicast | Multicast and unicast | Multicast | Broadcast |
| Best Path computation | Dijkstra | DUAL | Bellman-Form | Bellman-Ford |
| Metric used | Bandwidth | Bandwidth and Delay (default) | Hop Count | Hop Count |
| Organization type | Hierarchical | Flat | Flat | Flat |
| Convergence | Fast | Very Fast | Slow | Slow |
| Auto Summarization | No | Yes | Yes | Yes |
| Manual Summarization | Yes | Yes | No | No |
| Peer authentication | Yes | Yes | Yes | No |

# OSPF Advantage and Disadvantage

The hierarchical design of OSPF provides the following benefits:

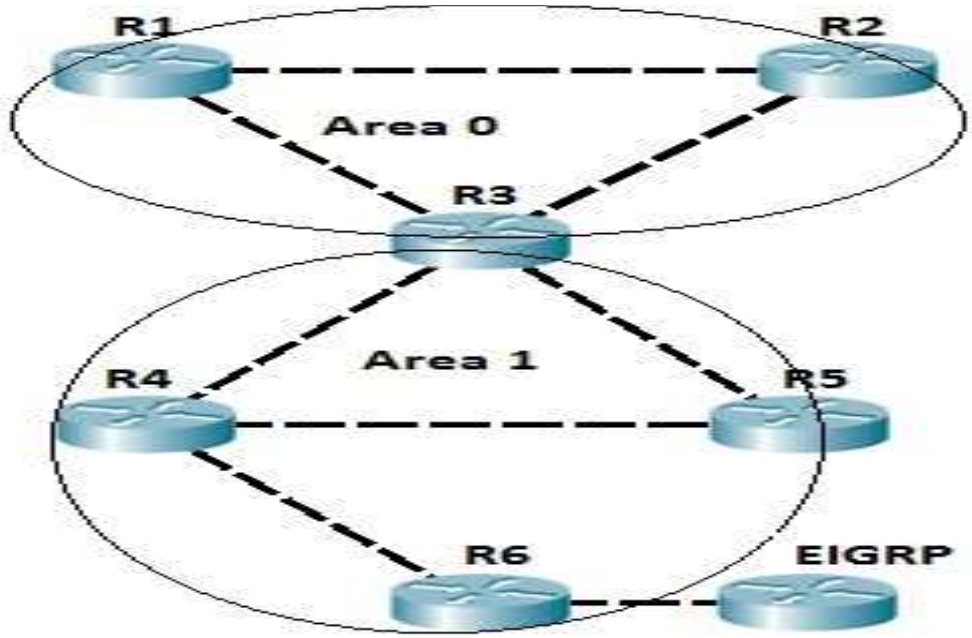
- Decrease routing overhead and flow of updates
- Limit network problems such as instability to an area
- Speed up convergence.
- One disadvantage of this is that planning and configuring OSPF is more difficult than other protocols.

# OSPF Area

- Uses AREAs to subdivide large networks, providing a hierarchical structure and limit the multicast LSAs within routers of the same area — Area 0 is called **backbone area** and all other areas connect directly to it. All OSPF networks must have a backbone area.
- Each area in the OSPF network has to connect to the backbone area (area 0). All router inside an area must have the same area ID to become OSPF neighbors. A router that has interfaces in more than one area (area 0 and area 1, for example) is called **Area Border Router (ABR)**. A router that connects an OSPF network to other routing domains (EIGRP network, for example) is called **Autonomous System Border Router (ASBR)**.

# OSPF Area

- All routers are running OSPF. Routers R1 and R2 are inside the backbone area (area 0). Router R3 is an ABR, because it has interfaces in two areas, namely area 0 and area 1. Router R4 and R5 are inside area 1. Router R6 is an ASBR, because it connects OSFP network to another routing domain (an EIGRP domain in this case). If the R1's directly connected subnet fails, router R1 sends the routing update only to R2 and R3, because all routing updates all localized inside the area.



# OSPF Topology Table

Each OSPF router stores routing and topology information in three tables:

- **Neighbor table** – stores information about OSPF neighbors
- **Topology table** – stores the topology structure of a network
- **Routing table** – stores the best routes

# OSPF Neighbor

At the Router level, when OSPF is enabled, it becomes aware of the following first:

- **Router ID** – Router ID is the IP address that will represent the router throughout the OSPF AS. Since a router may have multiple IP addresses (for its multiple interfaces), Cisco routers choose the highest loopback interface IP address. (Do not worry if you do not know what loopback interfaces are. They are covered later in the chapter). If loopback interfaces are not present, OSPF chooses the highest physical IP address configured within the active interfaces. Here highest literally means higher in number (Class C will be higher than Class A because 192 is greater than 10).
- **Links** – Simply speaking a Link is a network to which a router interface belongs. When you define the networks that OSPF will advertise, it will match interface addresses that belong to those networks. Each interface that matches is called a link. Each link has a status (up or down) and an IP address associated with it.

# OSPF Neighbor

Once a router is aware of the above two things, it will try to find more about its network by seeking out other OSPF speaking routers. At that stage the following terms come into use:

- **Hello Packets** - Similar to EIGRP hello packets, OSPF uses hello packets to discover neighbors and maintain relationships. Hello packet contains information such as area number that should match for a neighbor relation to be established. Hello packets are sent to multicast address 224.0.0.5.
- **Neighbors** - Neighbors is the term used to define two or more OSPF speaking routers connected to the same network and configured to be in the same OSPF area. Routers use hello packets to discover neighbors.
- **Neighbor Table** - OSPF will maintain a list of all neighbors from which hello packets have been received. For each neighbor various details such as RouterID and adjacency state are stored.
- **Area** - An OSPF area is a grouping of networks and routers. Every router in the area shares the same area id. Routers can belong to multiple areas; therefore, area id is linked to every interface. Routers will not exchange routing updates with routers belonging to different areas. Area 0 is called the **backbone area** and all other area must connect to it by having at least one router that belongs to both areas.

# OSPF Classifies Networks

- **Broadcast (multi-access)** - Broadcast (multi-access) networks are those that allow multiple devices to access (or connect to) the same network and also provide ability to broadcast. You will remember that when a packet is destined to all devices in a network, it is termed as a broadcast. Ethernet is an example of a broadcast multi-access network.
- **Non-Broadcast multi-access (NBMA)** - Networks that allow multi-access but do not have broadcast ability are called NBMA networks. Frame Relay networks are usually NBMA.
- **Point-to-Point** - Point-to-Point networks consist of direct connection between two routers and provide a single path of communication. When routers are connected back-to-back using serial interfaces, a point-to-point network is created. Point-to-point networks can also exist logically across geographical locations using various WAN technologies such as Frame Relay and PPP.
- **Point-to-Multipoint** - Point-to-Multipoint networks consist of multiple connections between a single interface of a router and multiple remote routers. All routers belong to the same network but have to communicate via the central router, whose interface connects the remote routers.

# OSPF Classifies Networks

Depending on the network type that OSPF discovers on the router interfaces, it will need to form **Adjacencies.** An **adjacency** is the relation between neighbors that allows direct exchange of routes.

- **Broadcast (multi-access)** – Since multiple routers can connect to such networks, OSPF elects a **Designated Router (DR)** and a **Backup Designated Router (BDR).** All routers in these networks, form adjacencies only with the DR and BDR. This also means that route updates are only shared between the routers and the DR and BDR. It is the duty of the DR to share routing updates with the rest of the routers in the network. If a DR loses connectivity to the network, the BDR will take its place. The election process is discussed later in the chapter.
- **NBMA** – Since NBMA is also a multi-access network, a DR and a BDR is elected and routers form adjacencies only with them. The problem with NBMA networks is that since broadcast capability and in turn multicast capability is not present, routers cannot discover neighbors. So NBMA networks require you to manually tell OSPF about the neighbors present in the network. Apart from this, OSPF functions as it does in a broadcast multi-access network.

# OSPF Classifies Networks

- **Point-to-Point** – Since there are only two routers present in a point-to-point network, there is no need to elect a DR and BDR. Both routers form adjacency with each other and exchange routing updates. Neighbors are discovered automatically in these networks.
- **Point-to-multipoint** – Point-to-multipoint interfaces are treat as special point-to-point interfaces by OSPF and it does a little extra work on here that is out of scope of CCNA. There is no DR/BDR election in such networks and neighbors are automatically discovered.

# OSPF Neighbor State

- Before establishing a neighbor relationship, OSPF routers need to go through several state changes. These states are explained below.
- **1. Init state** – a router has received a Hello message from the other OSFP router
  **2. 2-way state** – the neighbor has received the Hello message and replied with a Hello message of his own
  **3. Exstart state** – beginning of the LSDB exchange between both routers. Routers are starting to exchange link state information.
  **4. Exchange state** – DBD (Database Descriptor) packets are exchanged. DBDs contain LSAs headers. Routers will use this information to see what LSAs need to be exchanged.
  **5. Loading state** – one neighbor sends LSRs (Link State Requests) for every network it doesn't know about. The other neighbor replies with the LSUs (Link State Updates) which contain information about requested networks. After all the requested information have been received, other neighbor goes through the same process
  **6. Full state** – both routers have the synchronized database and are fully adjacent with each other.

# OSPF Topology Table

Once OSPF has formed adjacencies, it will start exchanging routing updates. The following two terms come to use here:

- **Link State Advertisements -** Link State Advertisements (LSAs) are OSPF packets containing link-state and routing information. These are exchanged between routers that have formed adjacencies. The packets essentially tell routers in the networks about different networks (links) that are present and how to reach them.
- **Topology Table -** The topology table contains information on every link the router learns about (via LSAs). The information is the topology table is used to compute the best path to remote networks.

# DR/BDR Election

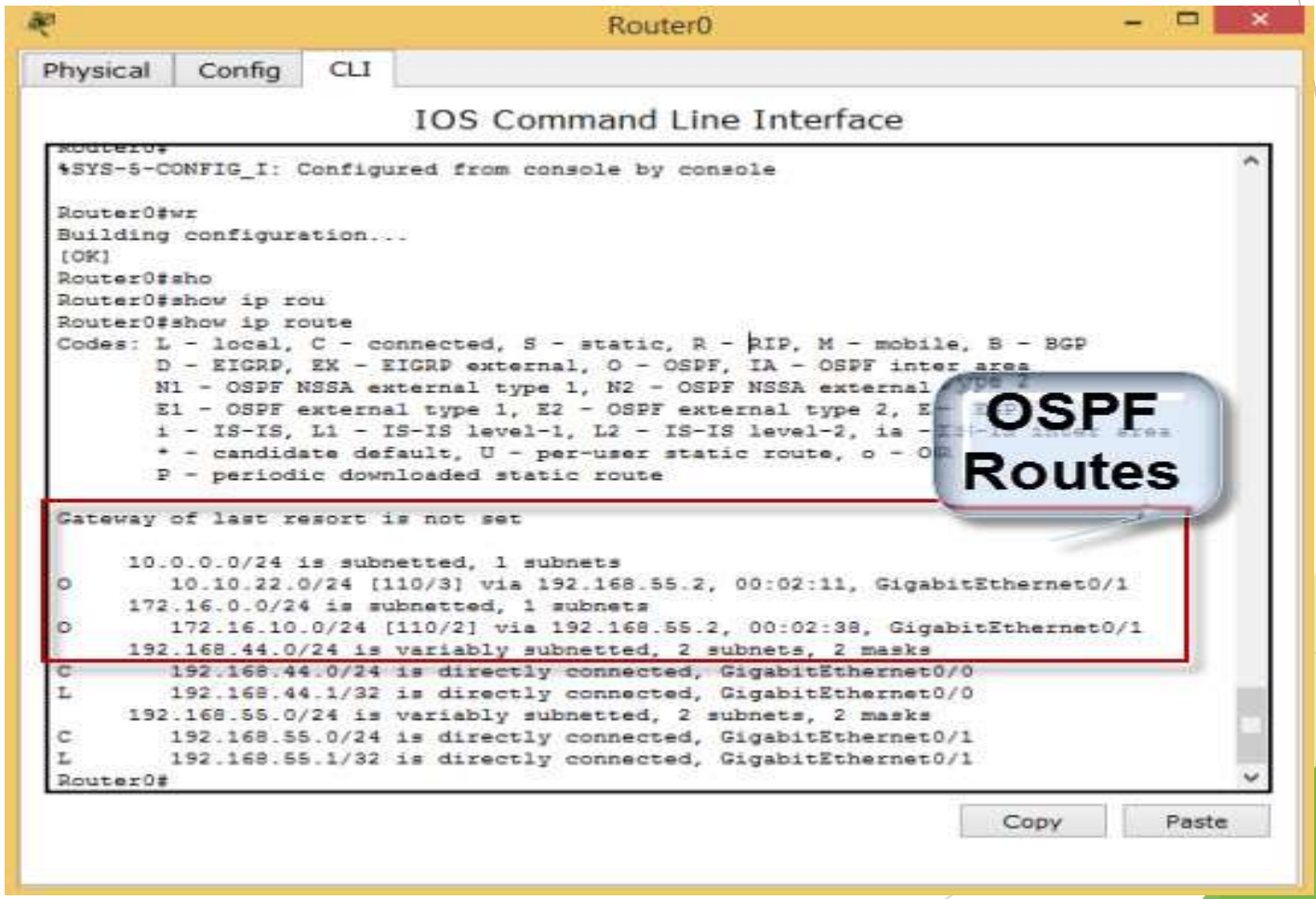
- When routers realize that they are connected to a multi-access network, they will look at each Hello packet received to find the **priority** and **Router ID** of each router. Then the priority is compared and the router with the highest priority is selected the DR. The router with the second highest priority becomes the BDR. By default the priority of each router is 1 and can be changed on a per-interface basis.
- If all routers have the default priority, then the router with the highest Router ID is elected the DR while the router with the second highest Router ID is elected the BDR. If the priority of a router is set to zero, it will not participate in the election process and will never be a DR or BDR.
- As you know, the Router ID is the highest physical IP address present on a Router. This can be overridden by using a loopback interface because a router will use the highest loopback address, if one is present.
- If you need to influence the DR/BDR election in a network segment, you can do one of the following:
- Manually increase the priority of a router interface to ensure that the router becomes the DR/BDR.
- Configure a loopback interface so that the Router ID becomes higher than that of the other routers in the network segment.

# OSPF Different Cost Values

| | |
|---|---|
| Gigabit Ethernet Interface (1 Gbps) | 1 |
| Fast Ethernet Interface (100 Mbps) | 1 |
| Ethernet Interface (10 Mbps) | 10 |
| DS1 (1.544 Mbps) | 64 |
| DSL (768 Kbps) | 133 |

# OSPF Routing Table

- Stores the best routes.



# How to Change OSPF Interface Cost

Cisco routers have three methods to change the OSPF interface cost:

1. By directly using the interface command ‘ip ospf cost <1-65535>’
Router#conf t
Router(config)#int gi0/0/0
Router(config-if)#ip ospf cost <1-65535>
We can verify this by using the ‘show ip ospf interface’ command.

2. Changing the ‘interface bandwidth’ setting (in kilobits), which changes the calculated value.
Router#conf t
Router(config)#int gi0/0/0
Router(config-if)#bandwidth <1-10000000>

3. Changing the OSPF reference bandwidth setting, which changes the calculated value.
Router#conf t Router(config)#router ospf 1
Router(config-router)#auto-cost reference-bandwidth 100000

# Loopback Interface

- Loopback interfaces are virtual, logical interfaces that exist in the software only. They are used for administrative purposes such as providing a stable OSPF interface or diagnostics. Using loopback interfaces with OSPF has the following benefits:
- Provides an interface that is always active.
- Provides an OSPF Router ID that is predictable and always same. Making is easier to troubleshoot OSPF.
- Router ID is a differentiator in DR/BDR election. Having a loopback interface with higher order IP address can influence the election.

# OSPF Configuration

- Configuration Syntax:
- Router(config)#router ospf <process id 1-65535>
- Router(config-router)network <network address> <wild card mask> area <0-4294967295>

# OSPF Troubleshooting

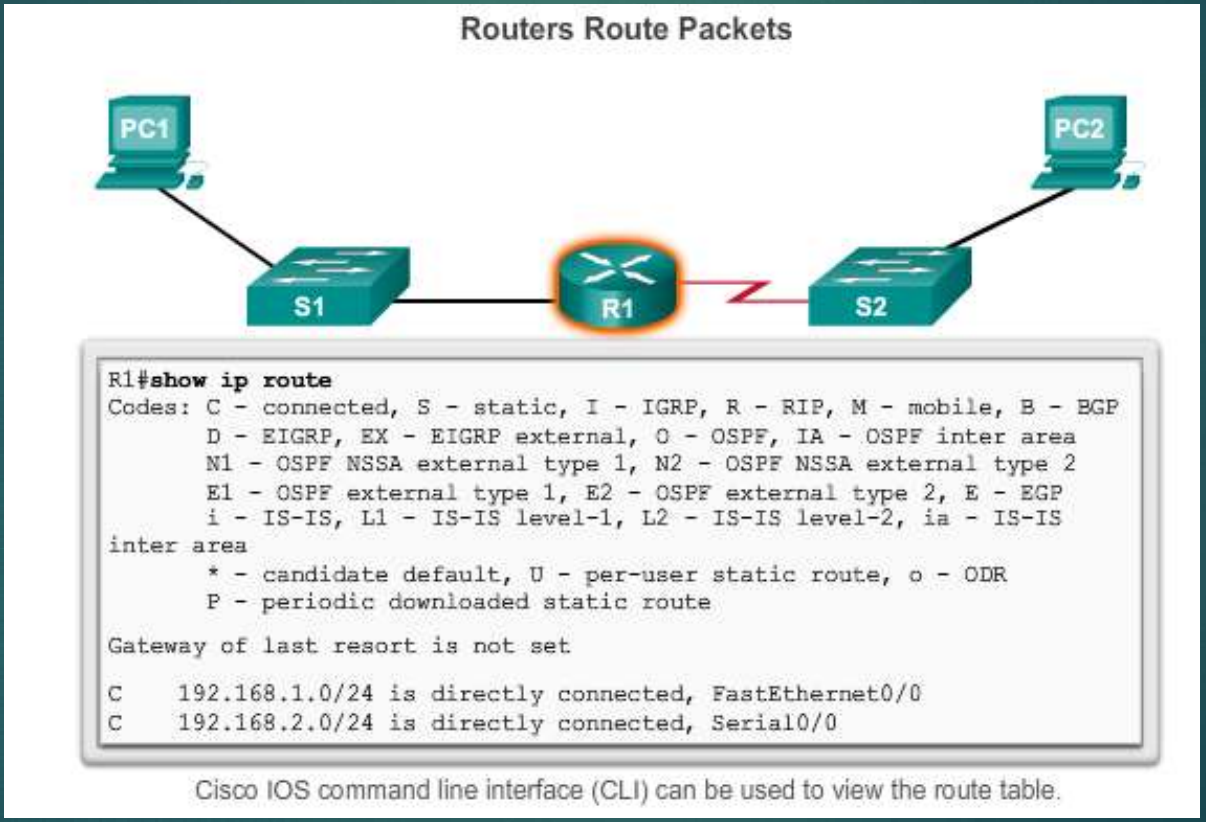
- show ip protocols
- show ip ospf
- show ip ospf interface
- show ip ospf neighbor
- show ip route

# Router and Routing Concept

Functions of a Router

# Why Routing?

The router is responsible for the routing of traffic between networks.



Functions of a Router

# Routers are Computers

Routers are specialized computers containing the following required components to operate:

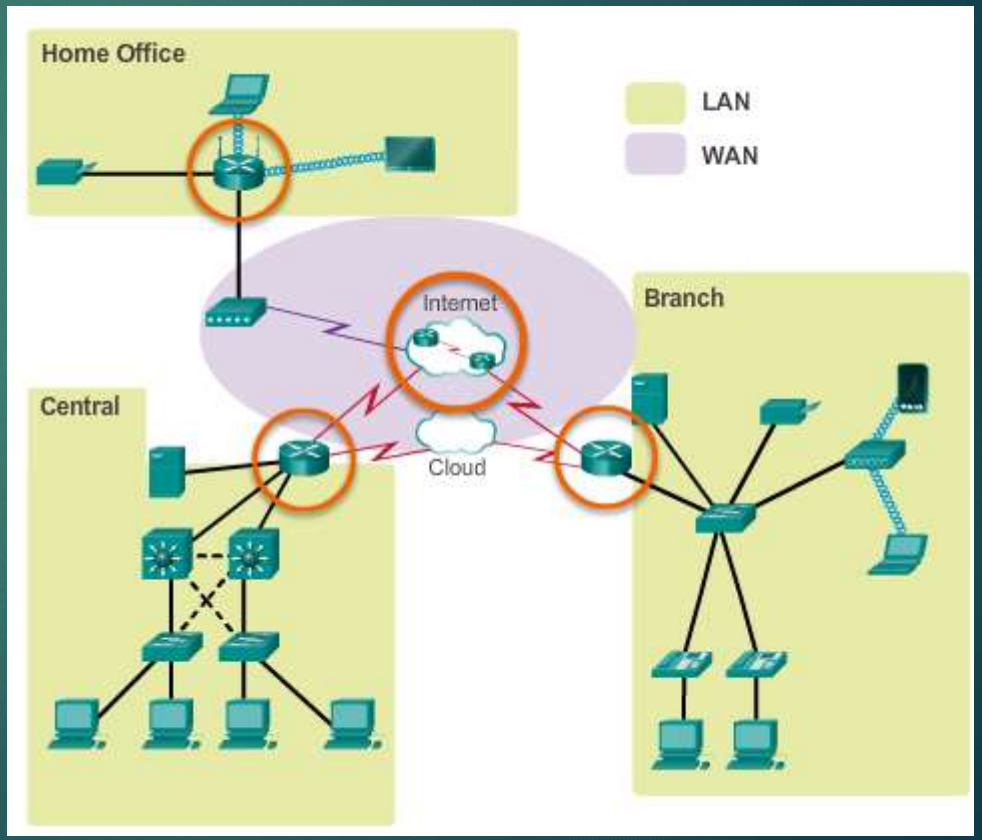
- Central processing unit (CPU)
- Operating system (OS) - Routers use Cisco IOS
- Memory and storage (RAM, ROM, NVRAM, Flash, hard drive)

| Memory | Volatile / Non-Volatile | Stores |
|---|---|---|
| RAM | Volatile | • Running IOS<br>• Running configuration file<br>• IP routing and ARP tables<br>• Packet buffer |
| ROM | Non-Volatile | • Bootup instructions<br>• Basic diagnostic software<br>• Limited IOS |
| NVRAM | Non-Volatile | • Startup configuration file |
| Flash | Non-Volatile | • IOS<br>• Other system files |

Functions of a Router

# Routers Interconnect Networks

- Routers can connect multiple networks.
- Routers have multiple interfaces, each on a different IP network.



Functions of a Router

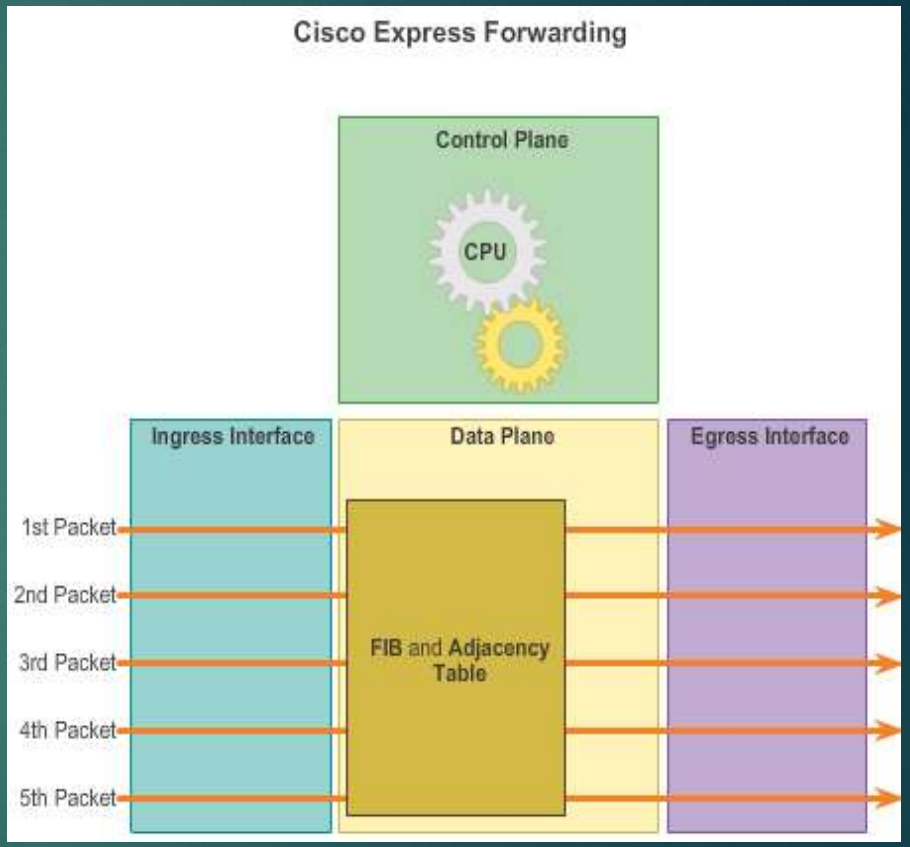
# Routers Choose Best Paths

- Routers use static routes and dynamic routing protocols to learn about remote networks and build their routing tables.
- Routers use routing tables to determine the best path to send packets.
- Routers encapsulate the packet and forward it to the interface indicated in routing table.

Functions of a Router

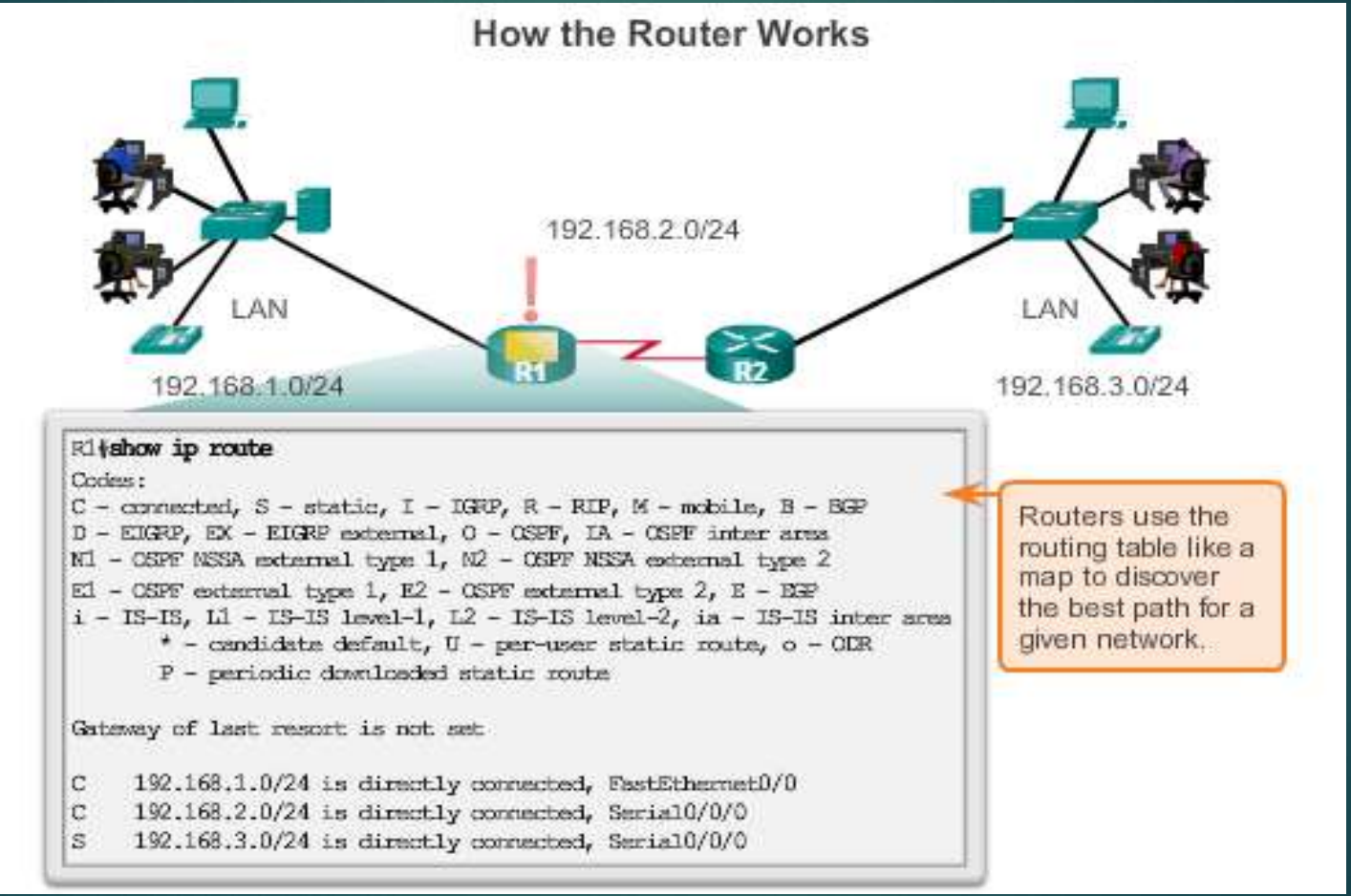
# Packet Forwarding Methods

- **Process switching** – An older packet forwarding mechanism still available for Cisco routers.
- **Fast switching** – A common packet forwarding mechanism which uses a fast-switching cache to store next hop information.
- **Cisco Express Forwarding (CEF)** – The most recent, fastest, and preferred Cisco IOS packet-forwarding mechanism. Table entries are not packet-triggered like fast switching but change-triggered.



Functions of a Router

# Routers Choose Best Paths



Path Determination

# Best Path

**Best path is selected by a routing protocol based on the value or metric it uses to determine the distance to reach a network**:

- A metric is the value used to measure the distance to a given network.
- Best path to a network is the path with the lowest metric.

**Dynamic routing protocols use their own rules and metrics to build and update routing tables**:

- Routing Information Protocol (RIP) - Hop count
- Open Shortest Path First (OSPF) - Cost based on cumulative bandwidth from source to destination
- Enhanced Interior Gateway Routing Protocol (EIGRP) - Bandwidth, delay, load, reliability

# Load Balancing

When a router has two or more paths to a destination with equal cost metrics, then the router forwards the packets using both paths equally:

- Equal cost load balancing can improve network performance.
- Equal cost load balancing can be configured to use both dynamic routing protocols and static routes.
- RIP, OSPF and EIGRP support equal cost load balancing.

Path Determination of the route

# Administrative Distance

If multiple paths to a destination are configured on a router, the path installed in the routing table is the one with the lowest Administrative Distance (AD):

- A static route with an AD of 1 is more reliable than an EIGRP-discovered route with an AD of 90.
- A directly connected route with an AD of 0 is more reliable than a static route with an AD of 1.

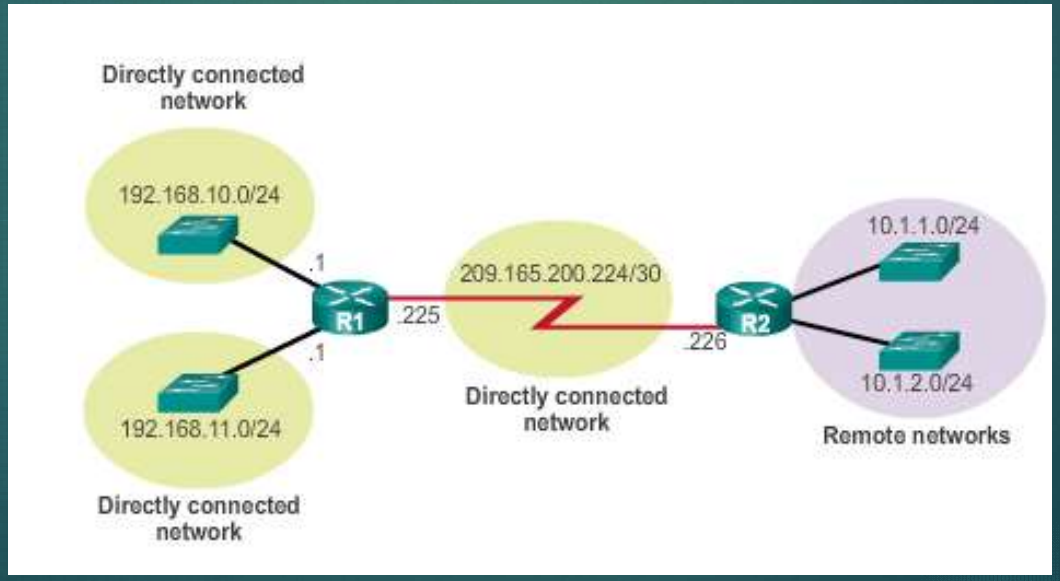
**Default Administrative Distances**

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

# The Routing Table

A routing table is a file stored in RAM that contains information about:

- Directly connected routes
- Remote routes
- Network or next hop associations



# Routing Table Sources

The **show ip route** command is used to display the contents of the routing table:

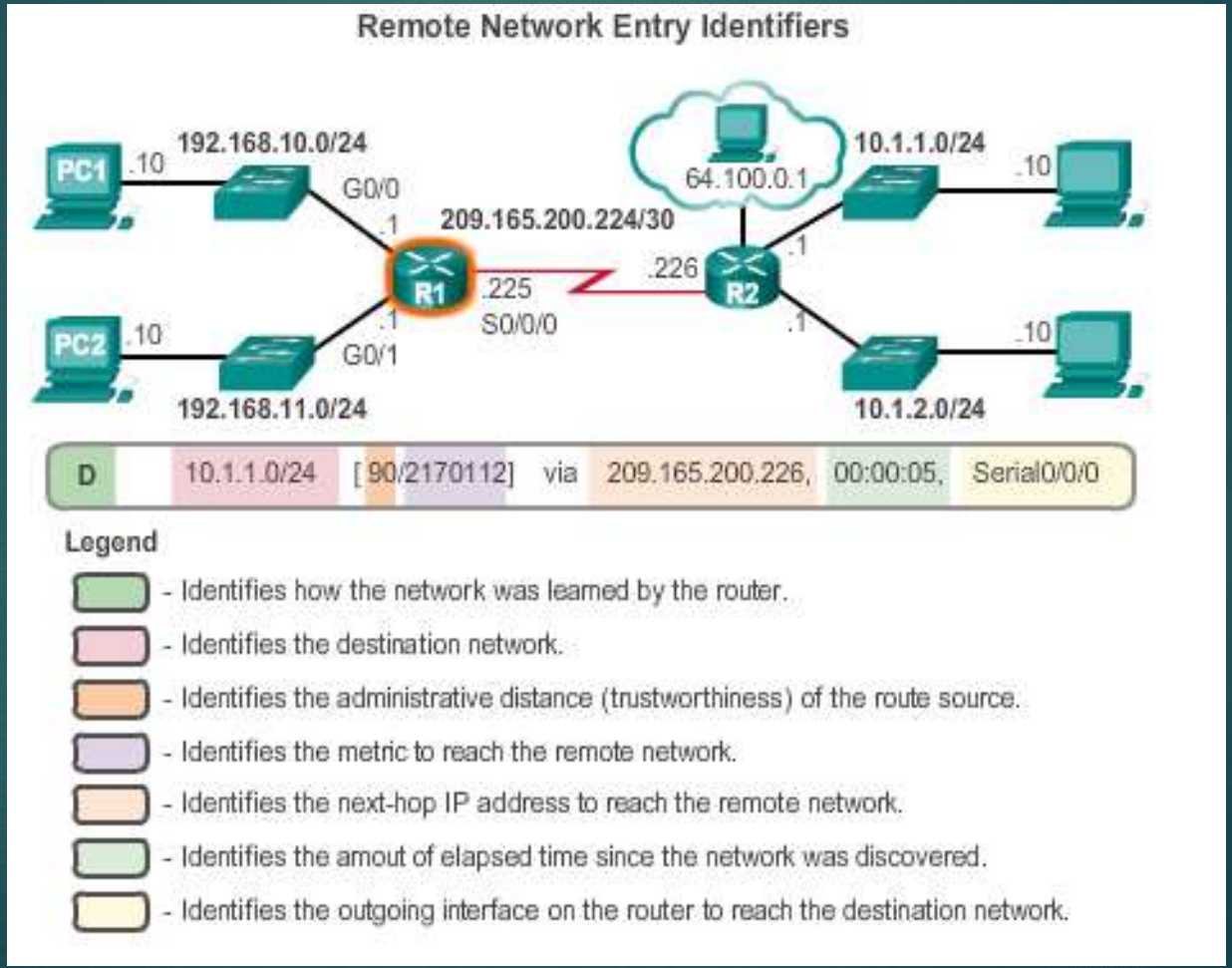
- **Local route interfaces -** Added to the routing table when an interface is configured. (displayed in IOS 15 or newer)
- **Directly connected interfaces -** Added to the routing table when an interface is configured and active.
- **Static routes -** Added when a route is manually configured and the exit interface is active.
- **Dynamic routing protocol -** Added when EIGRP or OSPF are implemented and networks are identified.

The Routing Table

# Routing Table Sources

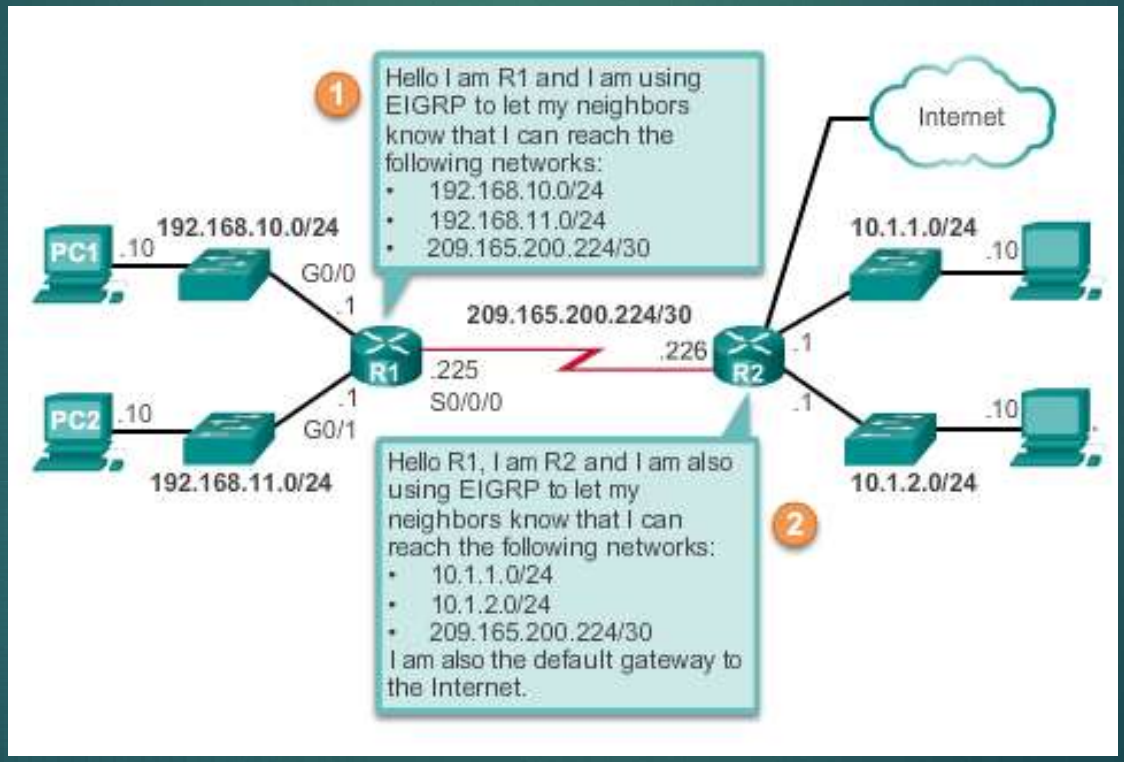

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia -
       IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.1.0/24 [90/2170112] via 209.165.200.226, 00:00:05,
```

# Interpreting the entries in the routing table.



# Dynamic Routing

Dynamic routing is used by routers to share information about the reachability and status of remote networks. It performs network discovery and maintains routing tables.



# IPv4 Routing Protocols

Cisco ISR routers can support a variety of dynamic IPv4 routing protocols including:

- **EIGRP –** Enhanced Interior Gateway Routing Protocol
- **OSPF –** Open Shortest Path First
- **IS-IS –** Intermediate System-to-Intermediate System
- **RIP –** Routing Information Protocol

# IPv6 Routing Protocols

Cisco ISR routers can support a variety of dynamic IPv6 routing protocols including:

- **RIPng -** RIP next generation
- **OSPFv3**
- **EIGRP** for IPv6
- **MP-BGP4 -** Multicast Protocol-Border Gateway Protocol

# Summary

- There are many key structures and performance-related characteristics referred to when discussing networks: topology, speed, cost, security, availability, scalability, and reliability.
- Cisco routers and Cisco switches have many similarities. They support a similar model operating system, similar command structures, and many of the same commands.
- One distinguishing feature between switches and routers is the type of interfaces supported by each.
- The main purpose of a router is to connect multiple networks and forward packets from one network to the next. This means that a router typically has multiple interfaces. Each interface is a member or host on a different IP network.

# Summary (cont.)

- The routing table is a list of networks known by the router.
- A remote network is a network that can only be reached by forwarding the packet to another router.
- Remote networks are added to the routing table in two ways: either by the network administrator manually configuring static routes or by implementing a dynamic routing protocol.
- Static routes do not have as much overhead as dynamic routing protocols; however, static routes can require more maintenance if the topology is constantly changing or is unstable.
- Dynamic routing protocols automatically adjust to changes without any intervention from the network administrator. Dynamic routing protocols require more CPU processing and also use a certain amount of link capacity for routing updates and messages.

# Summary (cont.)

- Routers make their primary forwarding decision at Layer 3, the Network layer. However, router interfaces participate in Layers 1, 2, and 3. Layer 3 IP packets are encapsulated into a Layer 2 data link frame and encoded into bits at Layer 1.
- Router interfaces participate in Layer 2 processes associated with their encapsulation. For example, an Ethernet interface on a router participates in the ARP process like other hosts on that LAN.
- Components of the IPv6 routing table are very similar to the IPv4 routing table. For instance, it is populated using directly connected interfaces, static routes and dynamically learned routes.

# OSPF LSA(Link State Advertisement) Types

- Router LSA (**Type 1**)
- Network LSA (**Type 2**)
- ABR Summary LSA (**Type 3**)
- ASBR Summary LSA (**Type 4**)
- ASBR Externel LSA (**Type 5**)
- Group Summary (**Type 6**)
- NSSA External LSA (**Type 7**)
- External Attributes LSA (**Type 8**)
- Opaque LSAs (**Type 9, 10, 11**)

**LSA Type 1** (**Router LSA**) packets are sent between routers within the same area of origin and do not leave the area. An OSPF router uses **LSA Type 1** packets to describe its own interfaces but also carries information about its neighbors to adjacent routers in the same area.

**LSA Type 2** (**Network LSA**) packets are generated by the **Designated Router** (**DR**) to describe all routers connected to its segment directly. **LSA Type 2** packets are flooded between neighbors in the same area of origin and remain within that area.

**LSA Type 3** (**Summary LSA**) packets are generated by Area Border Routers (**ABR**) to summarize its directly connected area, and advertise inter-area router information to other areas the **ABR** is connected to, with the use of a summary prefix (e.g 192.168.0.0/22). **LSA Type 3** packets are flooded to multiple areas throughout the network and help with OSPF's scalability with the use of summary prefixes.

**LSA Type 4** (**ASBR Summary LSA**) packets are the LSAs that advertise the presence of an Autonomous System Border Router (**ASBR**) to other areas. In the example below when **R2** (**ABR**) receives the **LSA Type 1** packet from **R1** it will create a **LSA Type 4** (**Summary ASBR LSA**) packet, which advertises the **ASBR** route received from **Area 1**, and inject it into **Area 0**.

**LSA Type 5** (**ASBR External LSA**) packets are generated by the **ASBR** to advertise external redistributed routes into the OSPF's **AS**. A typical example of an **LSA Type 5** would be an **external prefix** e.g **192.168.10.0/24** or **default route** (internet) as shown below:

**LSA Type 6** (**Group Membership LSA**) packets were designed for Multicast OSPF (MOSPF), a protocol that supports multicast routing through OSPF. MOSPF is not supported by Cisco and is not widely used and is expected to be retired soon.

**LSA Type 7** (**NSSA External LSA**) packets are used for some special area types that do not allow external distributed routes to go through and thus block **LSA Type 5** packets from flooding through them, **LSA Type 7** packets act as a mask for **LSA Type 5** packets to allow them to move through these special areas and reach the **ABR** that is able to translate **LSA Type 7** packets back to **LSA Type 5** packets.

**LSA Type 8** packets (**External Attributes LSA -OSPFv2-/ Link Local LSA -OSPFv3-**) in OSPFv2 (IPv4) are called **External Attribute LSAs**, and are used to transit BGP attributes through an OSPF network while BGP destinations are conveyed via **LSA Type 5** packets, however, this feature isn't supported by most routers. With OSPFv3 IPv6), **LSA Type 8** is redefined to carry **IPv6** information through OSPF network.

**LSA TYPE 9, 10 & 11**

Generally **Opaque LSAs** (**LSA Type 9, 10 & 11**) are used to extend the capabilities of OSPF allowing the protocol to carry information OSPF doesn't necessarily care about. Practical application of **Opaque LSAs** is in MPLS traffic engineering where they are used to communicate interface parameters such as maximum bandwidth, unreserved bandwidth, etc. Following is a short analysis of each of the three Opaque LSAs.

**LSA Type 9** in **OSPFv2** (**IPv4**) is defined as a **Link Scope Opaque LSA** for carrying OSPF information. For OSPFv3 it's redefined to handle a communication prefix for a special area type called **Stub Area**.

**LSA Type 10** packets are used to flood OSPF information through other area routers even if these routers do not process this information in order to extend OSPF functionality, this LSA is used for traffic engineering to advertise MPLS and other protocols.

**LSA Type 11** packets serve the same purpose as **LSA Type 10** packets but are not flooded into special area types (Stub areas).

| Device Name | Protocol configuration | IP Scheme / Interface configurations |
|---|---|---|
| **Laptop0** | — | G0/2 = 192.168.20.2\24 |
| **Laptop1** | — | G0/2 = 10.10.10.2\24 |
| **Laptop2** | | G0/2 = 192.168.21.2/24 |
| **Laptop3** | | G0/2 = 172.30.30.2/24 |
| **Laptop4** | | G0/2 = 192.168.1.2/24 |
| **Router 0** | OSPF1 | G0/0 = 20.20.20.1/30<br>G0/1 = 10.0.0.1/30 |
| **Router 1** | OSPF1 | G0/0 = 20.20.20.2/30<br>G0/1 = 30.30.30.1/30<br>G0/2 = 10.10.10.1/24 |
| **Router2** | OSPF 1 | G0/1 = 10.0.0.2/30, G0/0 = 40.40.40.1/30<br>G0/2 = 192.168.21.1/24 |
| **Router3** | OSPF 1 | G0/1 = 30.30.30.2/30, G0/2 = 172.30.30.1\24 |
| **Router4** | OSPF 1 | G0/0 = 40.40.40.2/30, G0/1 = 192.168.1.1/24 |

## Router0 Configuration:-

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#hostname Router0
Router0(config)#interface gigabitEthernet 0/0
Router0(config-if)#ip address 20.20.20.1 255.255.255.252
Router0(config-if)#no shutdown
Router0(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

Router0(config-if)#exit
Router0(config)#interface gigabitEthernet 0/1
Router0(config-if)#ip address 10.0.0.1 255.255.255.252
Router0(config-if)#no shutdown
Router0(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
Router0(config-if)#exit
Router0(config)#interface gigabitEthernet 0/2
Router0(config-if)#ip address 192.168.20.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up
Router0(config-if)#^Z
Router0#
%SYS-5-CONFIG_I: Configured from console by console
Router0#wr
Building configuration...
[OK]
Router0#
```

## Laptop0 Configuration:

IP Address-192.168.20.2

SubnetMask-255.255.255.0

Gateway-192.168.20.1

## Router1 Configuration:-

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router1
```

```
Router1(config)#interface gi0/0
Router1(config-if)#ip ad
Router1(config-if)#ip address 20.20.20.2 255.255.255.252
Router1(config-if)#no sh
Router1(config-if)#no shutdown

Router1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to
up

Router1(config-if)#exit
Router1(config)#interface gi 0/1
Router1(config-if)#ip ad
Router1(config-if)#ip address 30.30.30.1 255.255.255.252
Router1(config-if)#no sh
Router1(config-if)#no shutdown

Router1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

Router1(config-if)#exit
Router1(config)#in
Router1(config)#interface gi
Router1(config)#interface gigabitEthernet 0/2
Router1(config-if)#ip ad
Router1(config-if)#ip address 10.10.10.1 255.255.255.0
Router1(config-if)#no sh
Router1(config-if)#no shutdown

Router1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

Router1(config-if)#^Z
Router1#
%SYS-5-CONFIG_I: Configured from console by console

Router1#wr
Building configuration...
[OK]
Router1#
```

## Laptop1 Configuration:

IP Address-10.10.10.2

SubnetMask-255.255.255.0

Gateway-10.10.10.1

### Router2 Configuration:-

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router2
Router2(config)#interface gi0/1
Router2(config-if)#ip address 10.0.0.2 255.255.255.252
Router2(config-if)#no shutdown

Router2(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to
up

Router2(config-if)#exit
Router2(config)#interface gigabitEthernet 0/0
Router2(config-if)#ip address 40.40.40.1 255.255.255.252
Router2(config-if)#no shutdown
Router2(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
Router2(config-if)#exit
Router2(config)#interface gigabitEthernet 0/2
Router2(config-if)#ip address 192.168.21.1 255.255.255.0
Router2(config-if)#no shutdown
Router2(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up
Router2(config-if)#exit
Router2(config)#^Z
Router2#
%SYS-5-CONFIG_I: Configured from console by console
Router2#wr
Building configuration...
[OK]
Router2#
```

## Laptop2 Configuration:

IP Address-192.168.21.2

SubnetMask-255.255.255.0

Gateway-192.168.21.1

## Router3 Configuration:-

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router3
Router3(config)#interface gigabitEthernet 0/1
Router3(config-if)#ip address 30.30.30.2 255.255.255.252
Router3(config-if)#no shutdown
Router3(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to
up

Router3(config-if)#exit
Router3(config)#interface gigabitEthernet 0/2
Router3(config-if)#ip address 172.30.30.1 255.255.255.0
Router3(config-if)#no shutdown
Router3(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to
up
Router3(config-if)#exit
Router3(config)#exit
Router3#
%SYS-5-CONFIG_I: Configured from console by console

Router3#wr
Building configuration...
[OK]
Router3#
```

## Laptop3 Configuration:

IP Address-172.30.30.2

SubnetMask-255.255.255.0

Gateway-172.30.30.1

## Router4 Configuration:-

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router4
Router4(config)#interface gigabitEthernet 0/0
Router4(config-if)#ip address 40.40.40.1 255.255.255.252
Router4(config-if)#no shutdown
Router4(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to
up
Router4(config-if)#ip address 40.40.40.2 255.255.255.252
Router4(config-if)#no shutdown
Router4(config-if)#exit
Router4(config)#interface gigabitEthernet 0/2
Router4(config-if)#ip address 192.168.1.1 255.255.255.0
Router4(config-if)#no shutdown
Router4(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to
up

Router4(config-if)#exit
Router4(config)#exit
Router4#
%SYS-5-CONFIG_I: Configured from console by console

Router4#wr
Building configuration...
[OK]
Router4#
```

## Laptop4 Configuration:

IP Address-192.168.1.2

SubnetMask-255.255.255.0

Gateway-192.168.1.1

## OSPF Configuration

### Router0 Configuration:

```
Router0>en
Router0#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router0(config)#rou
Router0(config)#router os
Router0(config)#router ospf 1
Router0(config-router)#net
Router0(config-router)#network 20.20.20.0 0.0.0.3 are
Router0(config-router)#network 20.20.20.0 0.0.0.3 area 0
Router0(config-router)#netw
Router0(config-router)#network 10.0.0.0 0.0.0.3 are
Router0(config-router)#network 10.0.0.0 0.0.0.3 area 0
Router0(config-router)#netw
Router0(config-router)#network 192.168.20.0 0.0.0.255 are
Router0(config-router)#network 192.168.20.0 0.0.0.255 area 0
Router0(config-router)#exit
Router0(config)#exit
Router0#
%SYS-5-CONFIG_I: Configured from console by console

Router0#wr
Building configuration...
[OK]
Router0#
```

## Router1 Configuration:-

```
Router1>en
Router1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#router ospf 1
Router1(config-router)#network 20.20.20.0 0.0.0.3 area 0
Router1(config-router)#network 10.10.10.0 0.0.0.255 area 0
Router1(config-router)#network 30.30.30.0 0.0.0.3 area 1
Router1(config-router)#exit
Router1(config)#exit
Router1#
%SYS-5-CONFIG_I: Configured from console by console

Router1#
```

## Router2 Configuration:-

```
Router2>en
Router2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router2(config)#router ospf 1
Router2(config-router)#network 10.0.0.0 0.0.0.3 area 0
Router2(config-router)#network 192.168.21.0 0.0.0.255 area 0
Router2(config-router)#network 40.40.40.0 0.0.0.3 area 1
Router2(config-router)#exit
Router2(config)#exit
Router2#wr
Building configuration...
[OK]
Router2#
```

## Router3 Configuration:-

```
Router3>en
Router3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router3(config)#router ospf 1
Router3(config-router)#network 30.30.30.0 0.0.0.3 area 1
Router3(config-router)#network 172.30.30.0 0.0.0.255 area 1
Router3(config-router)#exit
Router3(config)#exit
Router3#
%SYS-5-CONFIG_I: Configured from console by console
Router3#wr
Building configuration...
[OK]
Router3#
```

## Router4 Configuration:-

```
Router4>en
Router4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router4(config)#router ospf 1
Router4(config-router)#network 40.40.40.0 0.0.0.3 area 1
Router4(config-router)#network 192.168.1.0 0.0.0.255 area 1
Router4(config-router)#exit
Router4(config)#exit
Router4#
%SYS-5-CONFIG_I: Configured from console by console
Router4#wr
Building configuration...
[OK]
Router4#
```

Run below commands on all router for checking OSPF.

1. show ip protocols
2. show ip ospf
3. show ip ospf interface
4. show ip ospf neighbor
5. show ip route

# OSPF LSA(Link State Advertisement) Types

- Router LSA (**Type 1**)
- Network LSA (**Type 2**)
- ABR Summary LSA (**Type 3**)
- ASBR Summary LSA (**Type 4**)
- ASBR Externel LSA (**Type 5**)
- Group Summary (**Type 6**)
- NSSA External LSA (**Type 7**)
- External Attributes LSA (**Type 8**)
- Opaque LSAs (**Type 9, 10, 11**)

**LSA Type 1** (**Router LSA**) packets are sent between routers within the same area of origin and do not leave the area. An OSPF router uses **LSA Type 1** packets to describe its own interfaces but also carries information about its neighbors to adjacent routers in the same area.

**LSA Type 2** (**Network LSA**) packets are generated by the **Designated Router** (**DR**) to describe all routers connected to its segment directly. **LSA Type 2** packets are flooded between neighbors in the same area of origin and remain within that area.

**LSA Type 3** (**Summary LSA**) packets are generated by Area Border Routers (**ABR**) to summarize its directly connected area, and advertise inter-area router information to other areas the **ABR** is connected to, with the use of a summary prefix (e.g 192.168.0.0/22). **LSA Type 3** packets are flooded to multiple areas throughout the network and help with OSPF's scalability with the use of summary prefixes.

**LSA Type 4** (**ASBR Summary LSA**) packets are the LSAs that advertise the presence of an Autonomous System Border Router (**ASBR**) to other areas. In the example below when **R2** (**ABR**) receives the **LSA Type 1** packet from **R1** it will create a **LSA Type 4** (**Summary ASBR LSA**) packet, which advertises the **ASBR** route received from **Area 1**, and inject it into **Area 0**.

**LSA Type 5** (**ASBR External LSA**) packets are generated by the **ASBR** to advertise external redistributed routes into the OSPF's **AS**. A typical example of an **LSA Type 5** would be an **external prefix** e.g **192.168.10.0/24** or **default route** (internet) as shown below:

**LSA Type 6** (**Group Membership LSA**) packets were designed for Multicast OSPF (MOSPF), a protocol that supports multicast routing through OSPF. MOSPF is not supported by Cisco and is not widely used and is expected to be retired soon.

**LSA Type 7** (**NSSA External LSA**) packets are used for some special area types that do not allow external distributed routes to go through and thus block **LSA Type 5** packets from flooding through them, **LSA Type 7** packets act as a mask for **LSA Type 5** packets to allow them to move through these special areas and reach the **ABR** that is able to translate **LSA Type 7** packets back to **LSA Type 5** packets.

**LSA Type 8** packets (**External Attributes LSA -OSPFv2-/ Link Local LSA -OSPFv3-**) in OSPFv2 (IPv4) are called **External Attribute LSAs**, and are used to transit BGP attributes through an OSPF network while BGP destinations are conveyed via **LSA Type 5** packets, however, this feature isn't supported by most routers. With OSPFv3 IPv6), **LSA Type 8** is redefined to carry **IPv6** information through OSPF network.

**LSA TYPE 9, 10 & 11**

Generally **Opaque LSAs** (**LSA Type 9, 10 & 11**) are used to extend the capabilities of OSPF allowing the protocol to carry information OSPF doesn't necessarily care about. Practical application of **Opaque LSAs** is in MPLS traffic engineering where they are used to communicate interface parameters such as maximum bandwidth, unreserved bandwidth, etc. Following is a short analysis of each of the three Opaque LSAs.

**LSA Type 9** in **OSPFv2** (**IPv4**) is defined as a **Link Scope Opaque LSA** for carrying OSPF information. For OSPFv3 it's redefined to handle a communication prefix for a special area type called **Stub Area**.

**LSA Type 10** packets are used to flood OSPF information through other area routers even if these routers do not process this information in order to extend OSPF functionality, this LSA is used for traffic engineering to advertise MPLS and other protocols.

**LSA Type 11** packets serve the same purpose as **LSA Type 10** packets but are not flooded into special area types (Stub areas).

### OSPF Default-Information Originate and the Default Route

The **default route** or the **Gateway of Last Resort** is used to forward packets if our destination IP address does not have a match in our routing table. In IPv4, the CIDR notation is 0.0.0.0/0, whereas, in IPv6, it is ::/0. And because the prefix length is 0, it is also the shortest possible match. Using the **'default-information originate'** command enables our default routes to be injected in our routing protocol, such as OSPF, and be propagated in our network.

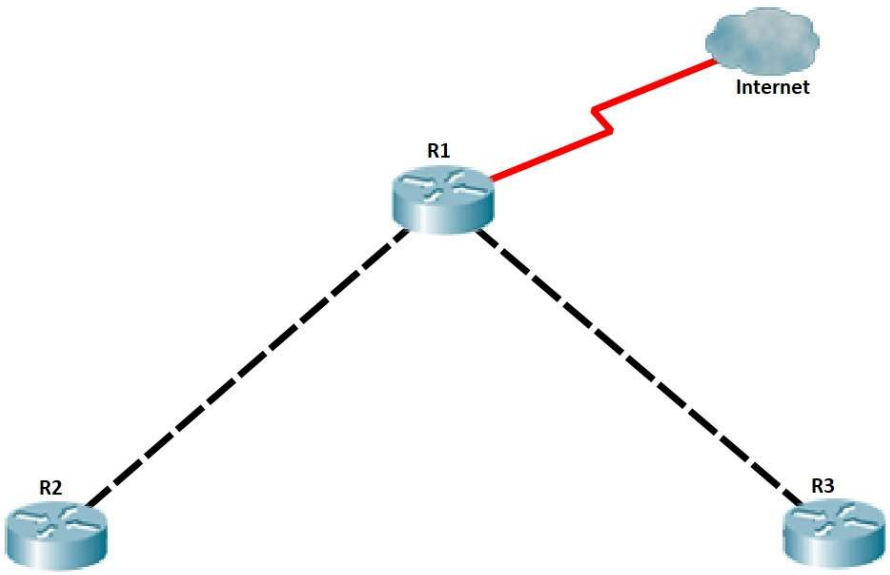
## What is a Default Route?

In networks where learning all of the more specific routes is not desirable, such as stub networks, default routes are beneficial. A default route is very useful when a router is connected to the Internet because, without it, the router must have all the routing entries of the Internet's networks, which can reach hundreds of thousands. Consider the massive CPU load; if the router is unable to handle it, it will degrade or crash.

With a static default route, the router only needs to know the destinations within the internal organization and will use the default route to forward IP packets for any other address to the Internet.

### OSPF Default-Information Originate Command

Any OSPF router can originate default routes injected into a normal area. The OSPF router does not create a default route into the OSPF domain by default. The **'default-information originate'** command is required for OSPF to generate a default route.



In our example above, router R1 is directly connected to the Internet. It's a common enterprise setup where all Internet traffic breaks out from a single site. In R1, we have a static route configured pointing to the ISP/Internet next hop device.

```
S* 0.0.0.0/0 [1/0] via 1.1.1.2
```

We are running the OSPF routing protocol, and R1 is peering with both R2 and R3.

```
R1#sh ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address
Interface

10.30.30.33       1   FULL/DR         00:00:30    10.30.30.33
GigabitEthernet0/2

10.20.20.22       1   FULL/DR         00:00:30    10.20.20.22
GigabitEthernet0/1
```

Before we proceed, let's check the routing tables of R2 and R3.

```
R2#sh ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
- BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2

       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP

       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area

       * - candidate default, U - per-user static route, o - ODR

       P - periodic downloaded static route
```

```
Gateway of last resort is not set



     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

C       10.20.20.20/30 is directly connected, GigabitEthernet0/1

L       10.20.20.22/32 is directly connected, GigabitEthernet0/1

O       10.30.30.32/30 [110/2] via 10.20.20.21, 00:05:10,
GigabitEthernet0/1

R3#sh ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
- BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2

       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP

       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area

       * - candidate default, U - per-user static route, o - ODR

       P - periodic downloaded static route



Gateway of last resort is not set



     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

O       10.20.20.20/30 [110/2] via 10.30.30.34, 00:05:20,
GigabitEthernet0/1

C       10.30.30.32/30 is directly connected, GigabitEthernet0/1

L       10.30.30.33/32 is directly connected, GigabitEthernet0/1
```

As we can see, we're only learning internal routes (directly connected networks in this case). Now, let's inject the default route into the OSPF domain.

```
R1#conf t

R1(config)#router ospf 1

R1(config-router)#default-information originate
```

As simple as that, default route exists in R2 and R3 as well.

```
R2#sh ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
- BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2

       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP

       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area

       * - candidate default, U - per-user static route, o - ODR

       P - periodic downloaded static route



Gateway of last resort is 10.20.20.21 to network 0.0.0.0



     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

C       10.20.20.20/30 is directly connected, GigabitEthernet0/1

L       10.20.20.22/32 is directly connected, GigabitEthernet0/1

O       10.30.30.32/30 [110/2] via 10.20.20.21, 00:16:22,
GigabitEthernet0/1

O*E2 0.0.0.0/0 [110/1] via 10.20.20.21, 00:00:28,
GigabitEthernet0/1
```

```
R3#sh ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B
- BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
2

       E1 - OSPF external type 1, E2 - OSPF external type 2, E -
EGP

       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-
IS inter area

       * - candidate default, U - per-user static route, o - ODR

       P - periodic downloaded static route



Gateway of last resort is 10.30.30.34 to network 0.0.0.0



     10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks

O       10.20.20.20/30 [110/2] via 10.30.30.34, 00:16:39,
GigabitEthernet0/1

C       10.30.30.32/30 is directly connected, GigabitEthernet0/1

L       10.30.30.33/32 is directly connected, GigabitEthernet0/1

O*E2 0.0.0.0/0 [110/1] via 10.30.30.34, 00:00:45,
GigabitEthernet0/1
```

The **EIGRP** metric is **calculated** by a formula using **five separate** values known as **K Values**. By default only K Values 1 and 3 are used (Bandwidth & Delay), K2, K4 and K5 are set to 0. The EIGRP metric formula and K Values are defined below;

EIGRP Metric = 256*((K1*Bw) + (K2*Bw)/(256-Load) + K3*Delay)*(K5/(Reliability + K4)))

K1 = Bandwidth

K2 = Load

K3 = Delay

K4 = Reliability

K5 = Maximum Transmission Unit (MTU)

So if you use the order of operations you can deduce the equation down to

EIGRP Metric = 256(Bandwidth + Delay)

Now keep in mind the Bandwidth and Delay have formulas in and of themselves to derive those variables. To determine **the bandwidth** you'll divide the interface bandwidth from the **max bandwidth**. To determine delay you'll divide the interface delay by 10 as the EIGRP metric uses ten's of microseconds in its calculation. View the formulas below;

Bandwidth = (10^7/Bandwidth in Kbps) Delay = 10/uSec

So if you want to determine the composite metric of a T1 link at 1.544Mbs (1544Kbps) you'll need to get the **bandwidth** and **delay variables** first then plug those into the **EIGRP metric** calculation formula as shown below; Keep in mind the delay on a T1 serial interface is 20000uSec (20,000 Microseconds)

Bandwidth = (10^7/1544) = 6476.68 == 6476 (rounded down) Delay = (10/20000) = 2000 EIGRP Metric = 256*(6476 + 2000) = 2169856

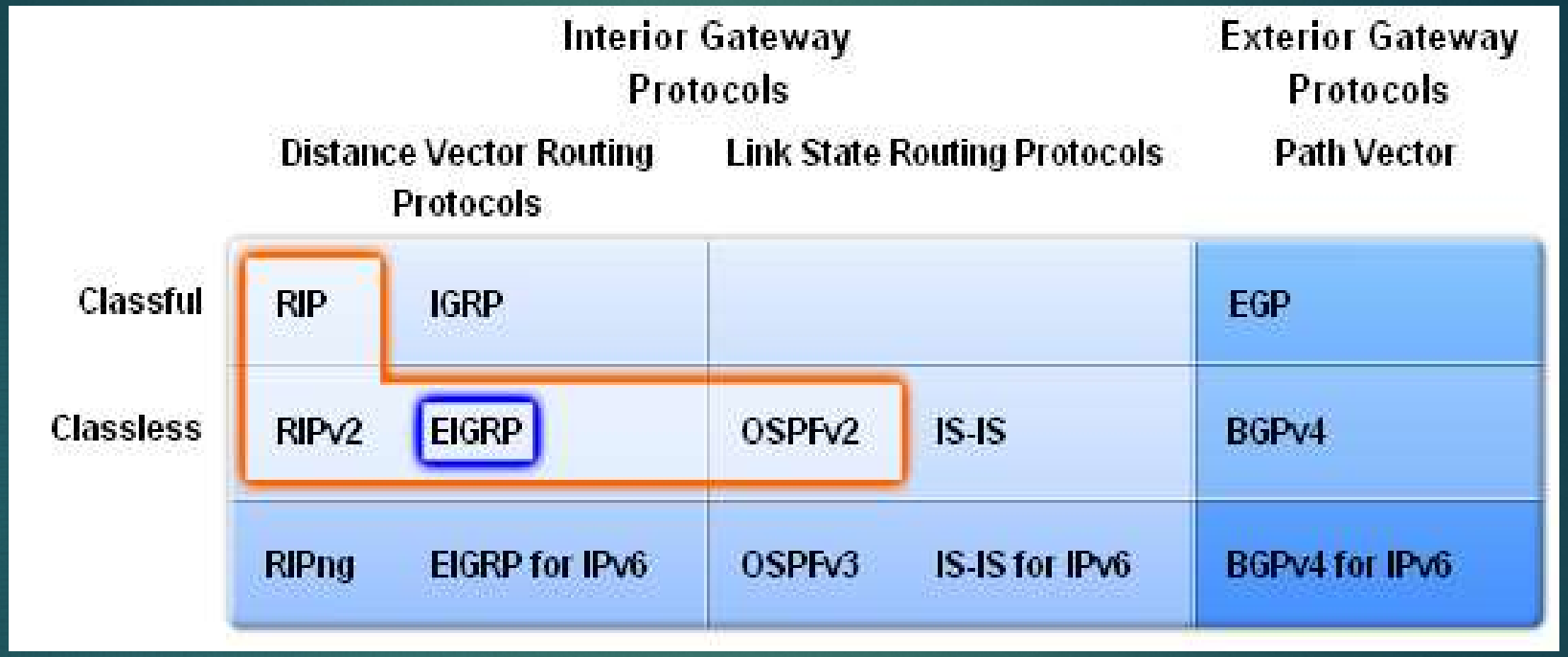
# EIGRP

ROUTING PROTOCOLS AND CONCEPTS

# Objectives

- Describe the background and history of Enhanced Interior Gateway Routing Protocol (EIGRP).
- Examine the basic EIGRP configuration commands and identify their purposes.
- Calculate the composite metric used by EIGRP.
- Describe the concepts and operation of DUAL.
- Describe the uses of additional configuration commands in EIGRP.

# Introduction



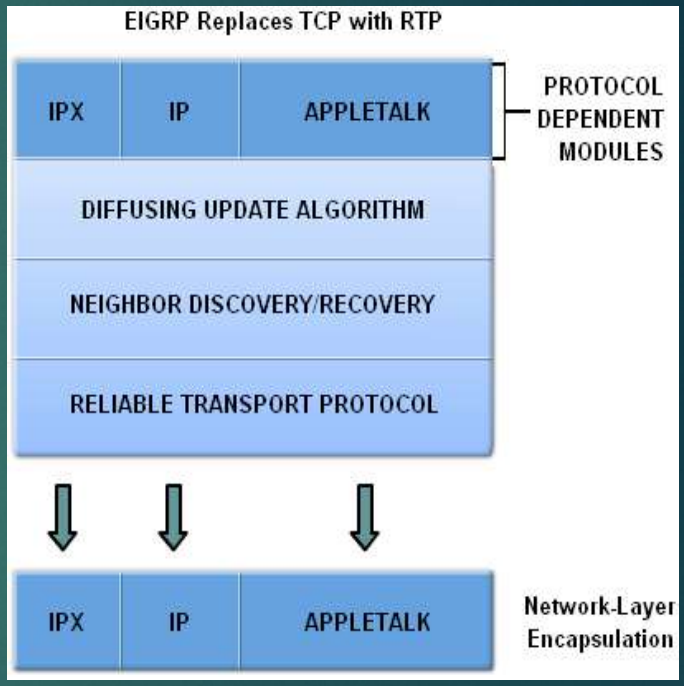
# EIGRP

- Roots of EIGRP: IGRP
  - -Developed in 1985 to overcome RIPv1's limited hop count
  - -Distance vector routing protocol
  - -Metrics used by IGRP
    - bandwidth (used by default)
    - Delay (used by default)
    - reliability
    - load
  - -Discontinued support starting with IOS 12.2(13)T & 12.2(R1s4)S

# EIGRP

**Reliable Transport Protocol (RTP)**

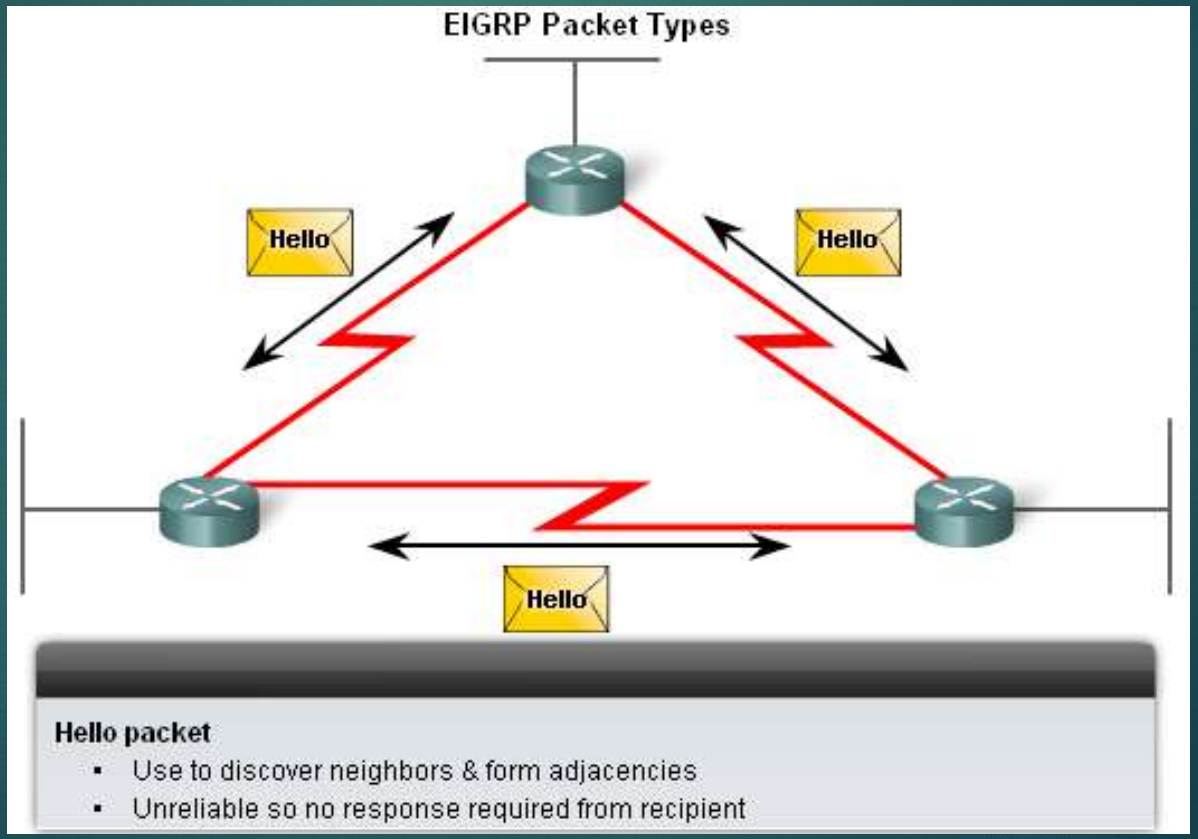
- **Purpose of RTP**
  - Used by EIGRP to transmit and receive EIGRP packets
- **Characteristics of RTP**
  - Involves both reliable & unreliable delivery of EIGRP packet
    - Reliable delivery requires acknowledgment from destination
    - Unreliable delivery does not require an acknowledgement from destination
  - Packets can be sent
    - Unicast
    - Multicast
      - Using address 224.0.0.10



# EIGRP

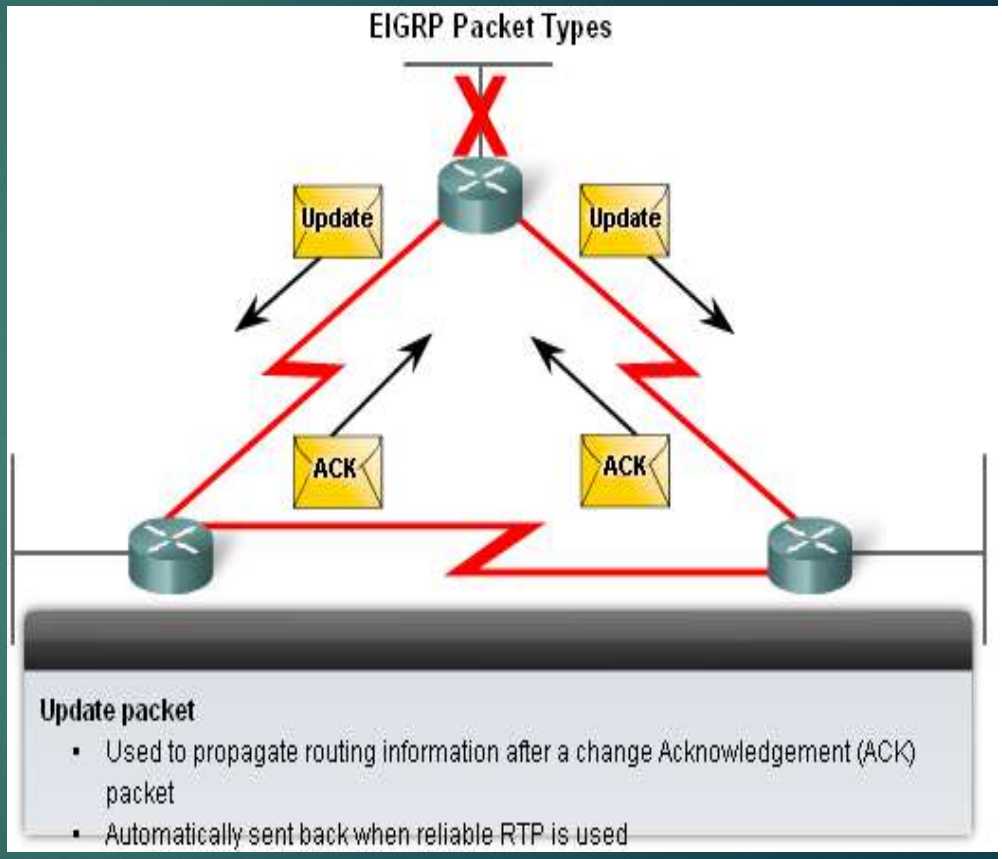
EIGRP's 5 Packet Types

- **Hello packets**
  - Used to discover & form adjacencies with neighbors



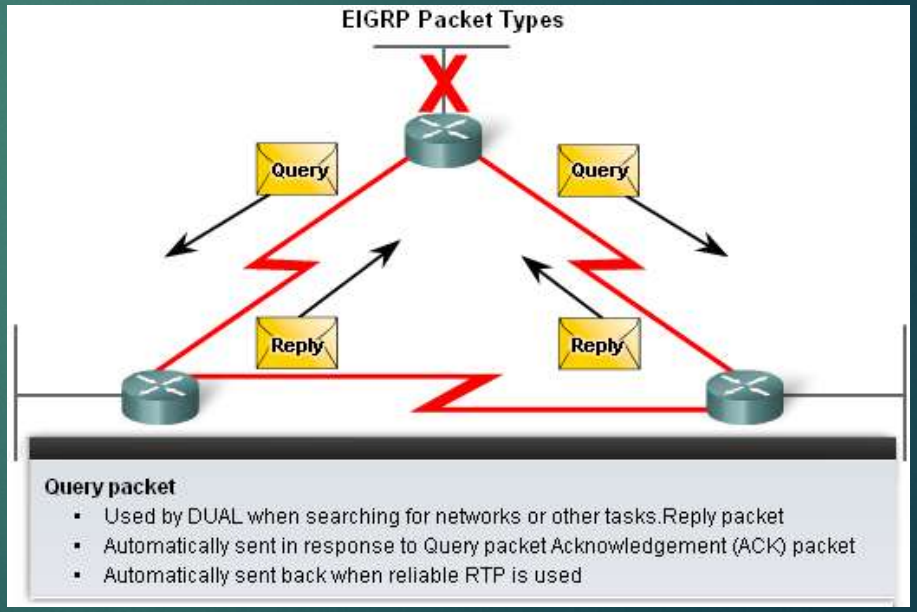
# EIGRP

- **Update packets**
  - Used to propagate routing information
- **Acknowledgement packets**
  - Used to acknowledge receipt of update, query & reply packets



# EIGRP

- **Query & Reply packets**
  - Used by DUAL for searching for networks
  - Query packets
    - -Can use
      - Unicast
      - Multicast
  - Reply packet
    - -Use only
      - unicast

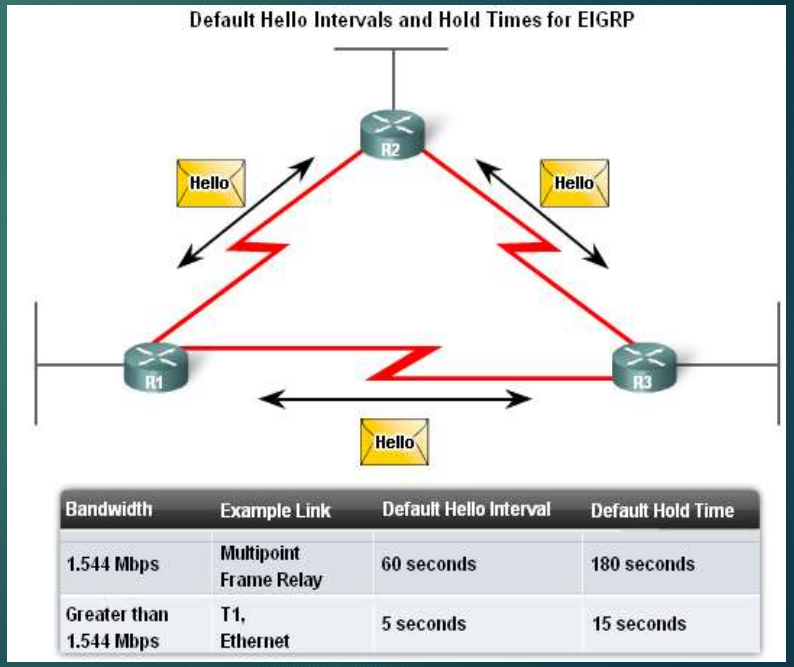


# EIGRP

- **Purpose of Hello Protocol**
  - To discover & establish adjacencies with neighbor routers
- **Characteristics of hello protocol**
  - Time interval for sending hello packet
    - Most networks it is every 5 seconds
    - Multipoint non broadcast multi-access networks
      - Unicast every 60 seconds

-Holdtime

- This is the maximum time router should wait before declaring a neighbor down
- Default holdtime
  - 3 times hello interval

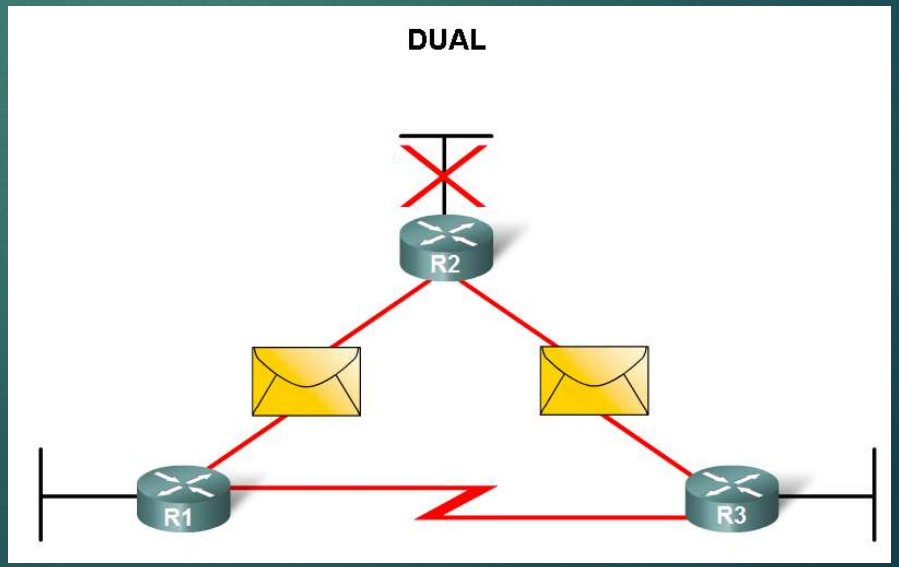



| Bandwidth | Example Link | Default Hello Interval | Default Hold Time |
|---|---|---|---|
| 1.544 Mbps | Multipoint Frame Relay | 60 seconds | 180 seconds |
| Greater than 1.544 Mbps | T1, Ethernet | 5 seconds | 15 seconds |

# EIGRP

**D**iffusing **U**pdate **Al**gorithm (DUAL)

- Purpose
  - EIGRP's primary method for preventing routing loops
- Advantage of using DUAL
  - Provides for fast convergence time by keeping a list of loop-free backup routes



# EIGRP

- Administrative Distance (AD)
  - Defined as the trustworthiness of the source route
- EIGRP default administrative distances
  - Summary routes = 5
  - Internal routes = 90
  - Imported routes = 170

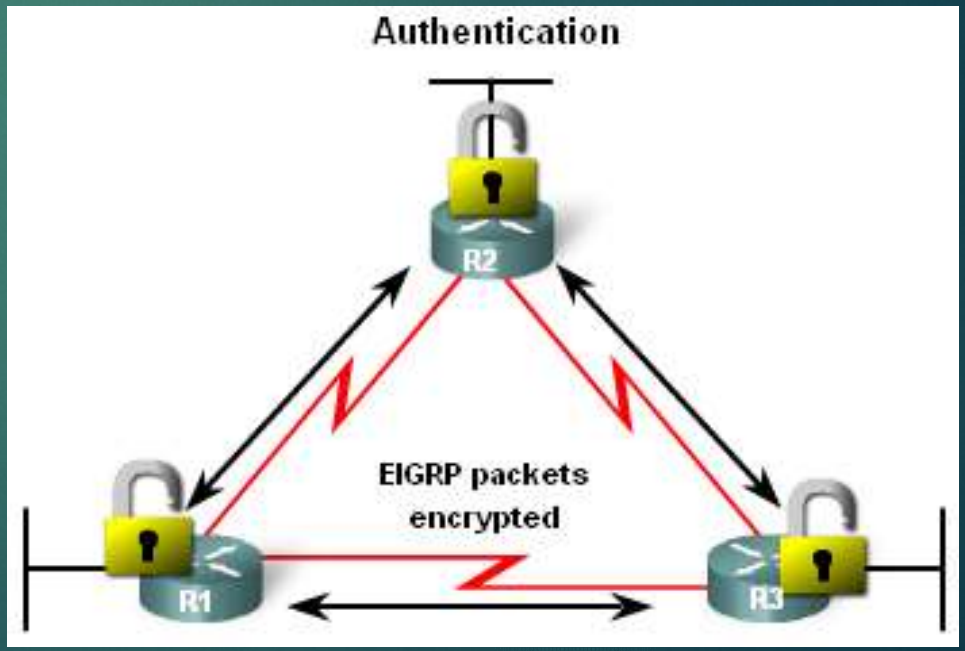
**Default Administrative Distances**

| Route Source | Administrative Distance |
|---|---|
| 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 |

# EIGRP

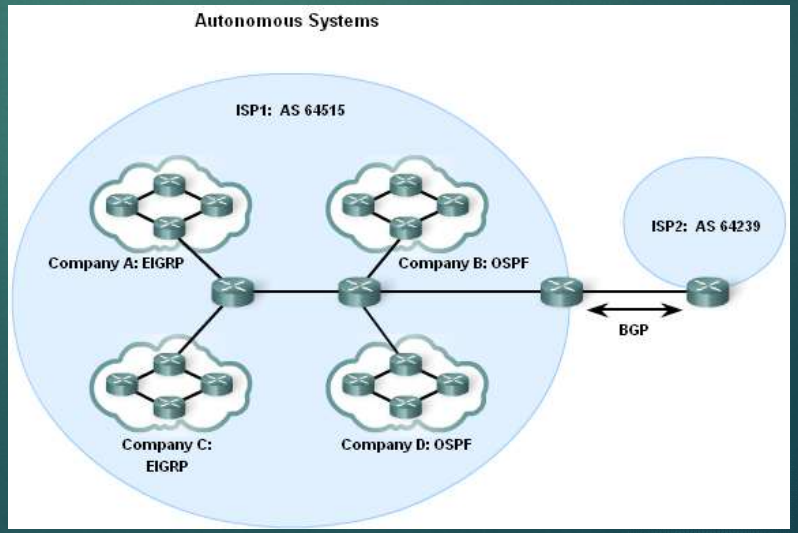
**Authentication**

- EIGRP can
  - Encrypt routing information
  - Authenticate routing information



# Basic EIGRP Configuration

- Autonomous System (AS) & Process IDs
  - This is a collection of networks under the control of a single authority (reference RFC 1930)
  - AS Numbers are assigned by IANA
  - Entities needing AS numbers
    - ISP
    - Internet Backbone prodiers
    - Institutions connecting to other institutions using AS numbers

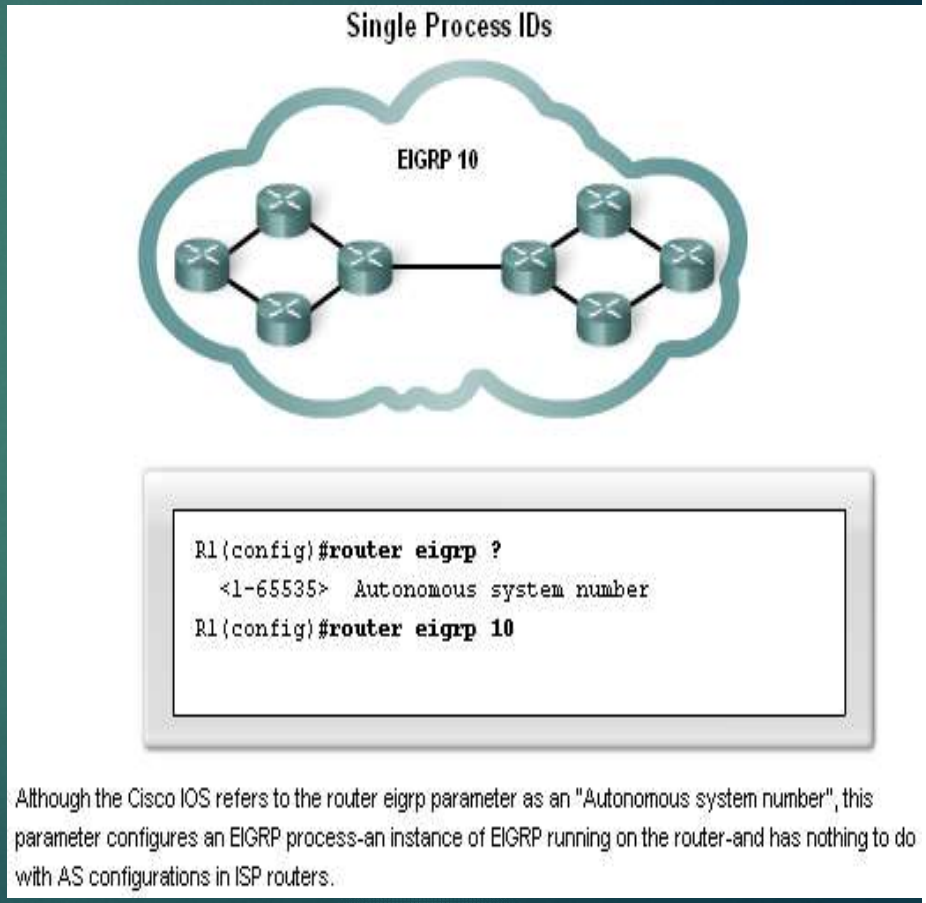


# Basic EIGRP Configuration

- EIGRP autonomous system number actually functions as a process ID
- Process ID represents an instance of the routing protocol running on a router
- Example

Router(config)#router

eigrp *autonomous-system*



# Basic EIGRP Configuration

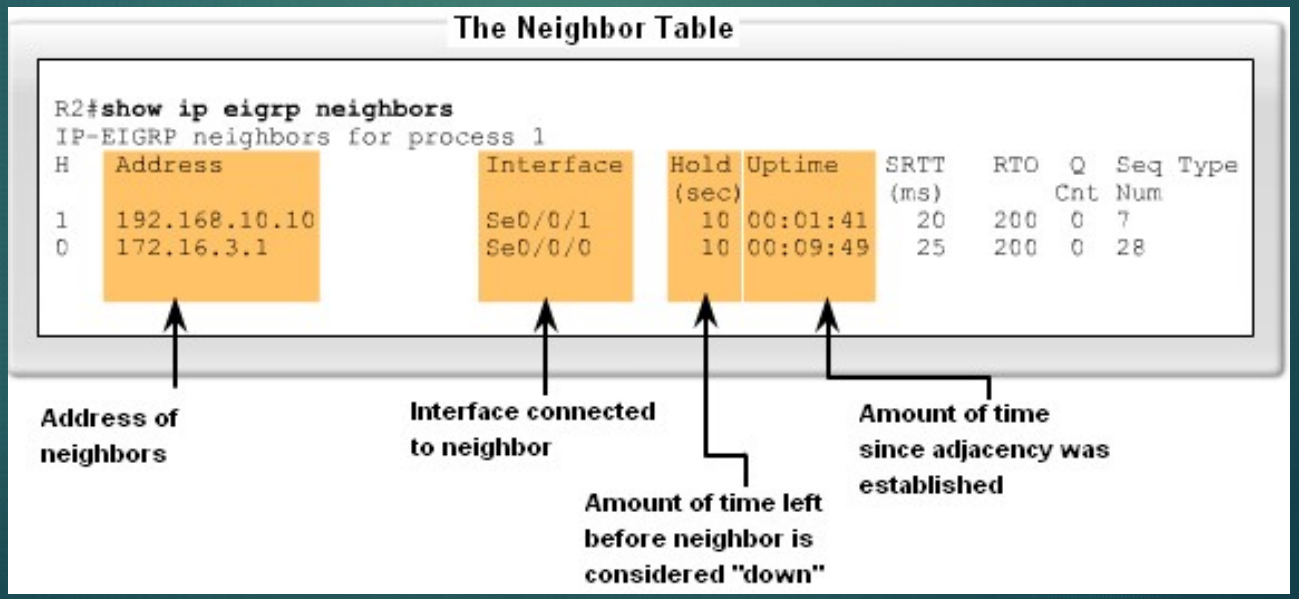
The *router eigrp* command

- The global command that enables eigrp is
  - *router eigrp autonomous-system*
  - -All routers in the EIGRP routing domain must use the same process ID number (autonomous-system number)
  - (router-eigrp)#network ip_address wildcard_mask

# Basic EIGRP Configuration

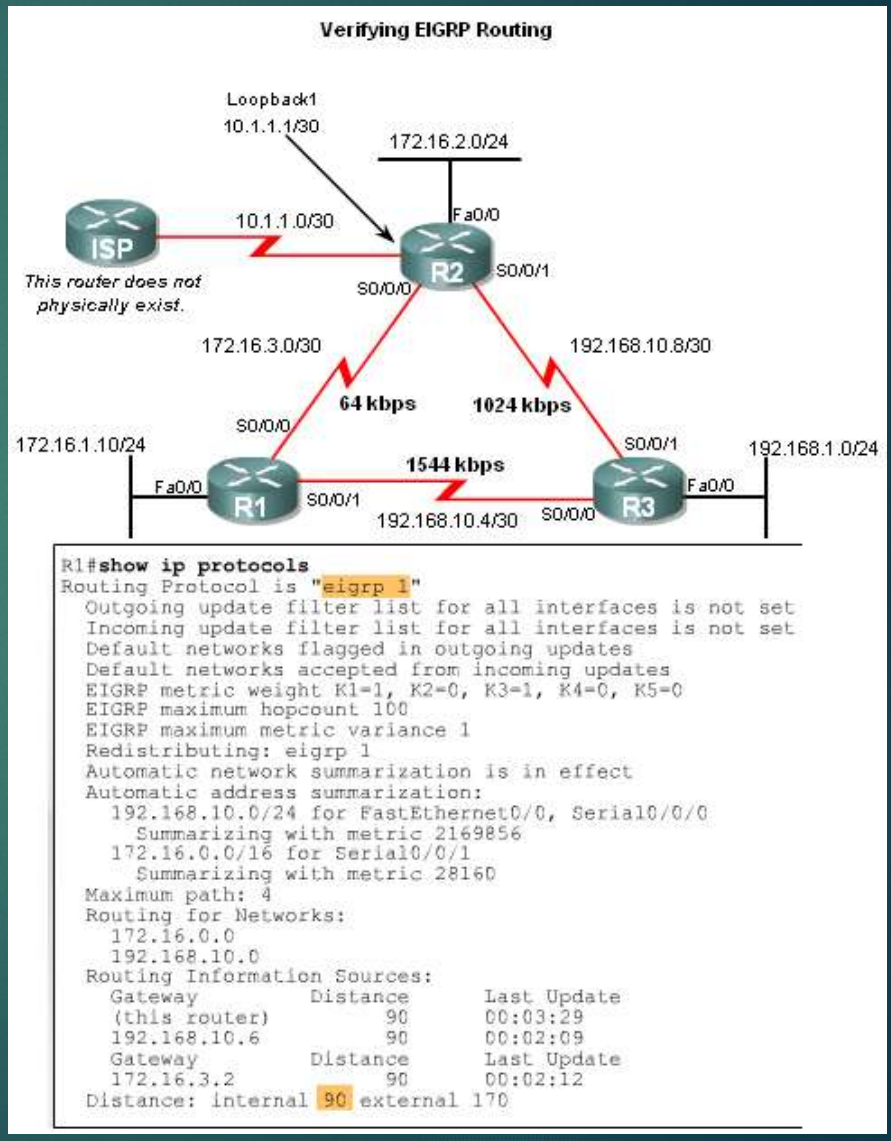
**Verifying EIGRP**

- EIGRP routers must establish adjacencies with their neighbors before any updates can be sent or received
- Command used to view neighbor table and verify that EIGRP has established adjacencies with neighbors is
  - ***show ip eigrp neighbors***



# EIGRP

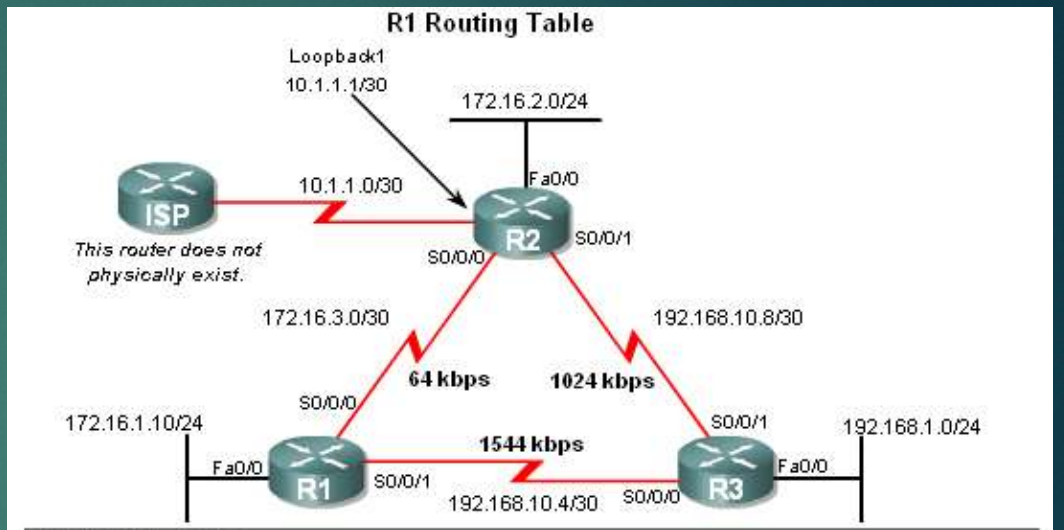
- The ***show ip protocols*** command is also used to verify that EIGRP is enabled



# Basic EIGRP Configuration

**Examining the Routing Table**

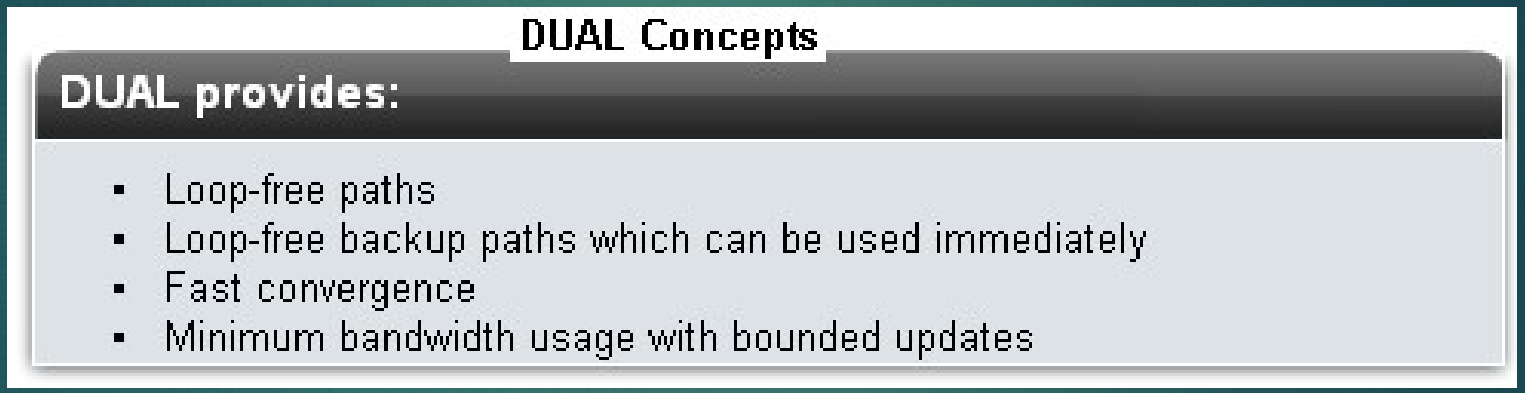
- The **show ip route** command is also used to verify EIGRP
- EIGRP routes are denoted in a routing table by the letter "**D**"
- By default , EIGRP automatically summarizes routes at major network boundary




```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
<Output omitted>

Gateway of last resort is not set

     192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.10.0/24 is a summary, 00:03:50, Null0
C       192.168.10.4/30 is directly connected, Serial0/0/1
D       192.168.10.8/30 [90/2681856] via 192.168.10.6, 00:02:43, Serial0/0/1
     172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D       172.16.0.0/16 is a summary, 00:10:52, Null0
C       172.16.1.0/24 is directly connected, FastEthernet0/0
D       172.16.2.0/24 [90/2172416] via 172.16.3.2, 00:10:47, Serial0/0/0
C       172.16.3.0/30 is directly connected, Serial0/0/0
D    192.168.1.0/24 [90/2172416] via 192.168.10.6, 00:02:31, Serial0/0/1
```

# DUAL Concepts

- The **D**iffusing **U**pdate **Al**gorithm (DUAL) is used to prevent looping



# DUAL Concepts

- Successor
  - The best least cost route to a destination found in the routing table
- Feasible distance
  - The lowest calculated metric along a path to a destination network

**Feasible Distance and Successor**

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

Gateway of last resort is not set

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

feasible distance — successor

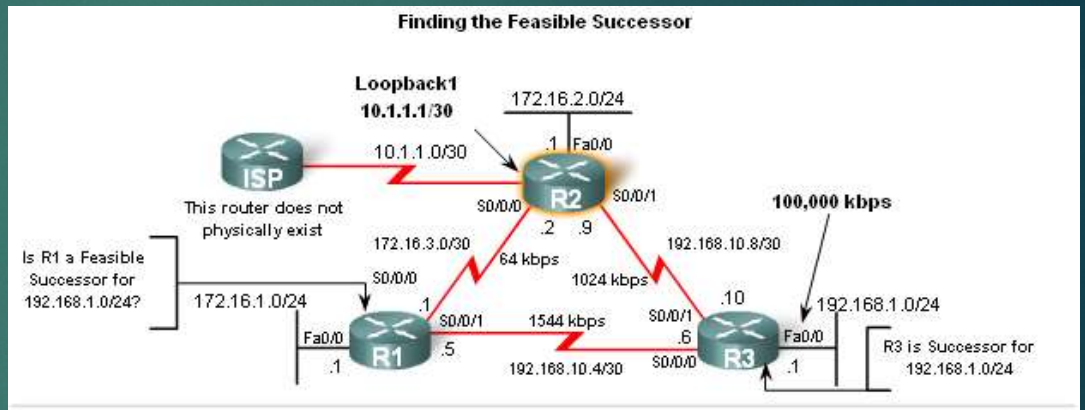
**R3 at 192.168.10.10 is the successor for network 192.168.1.0/24. This route has a feasible distance of 3014400.**

# DUAL Concepts

Feasible Successors, Feasibility Condition & Reported Distance

- Feasible Successor

  -This is a loop free backup route to same destination as successor route

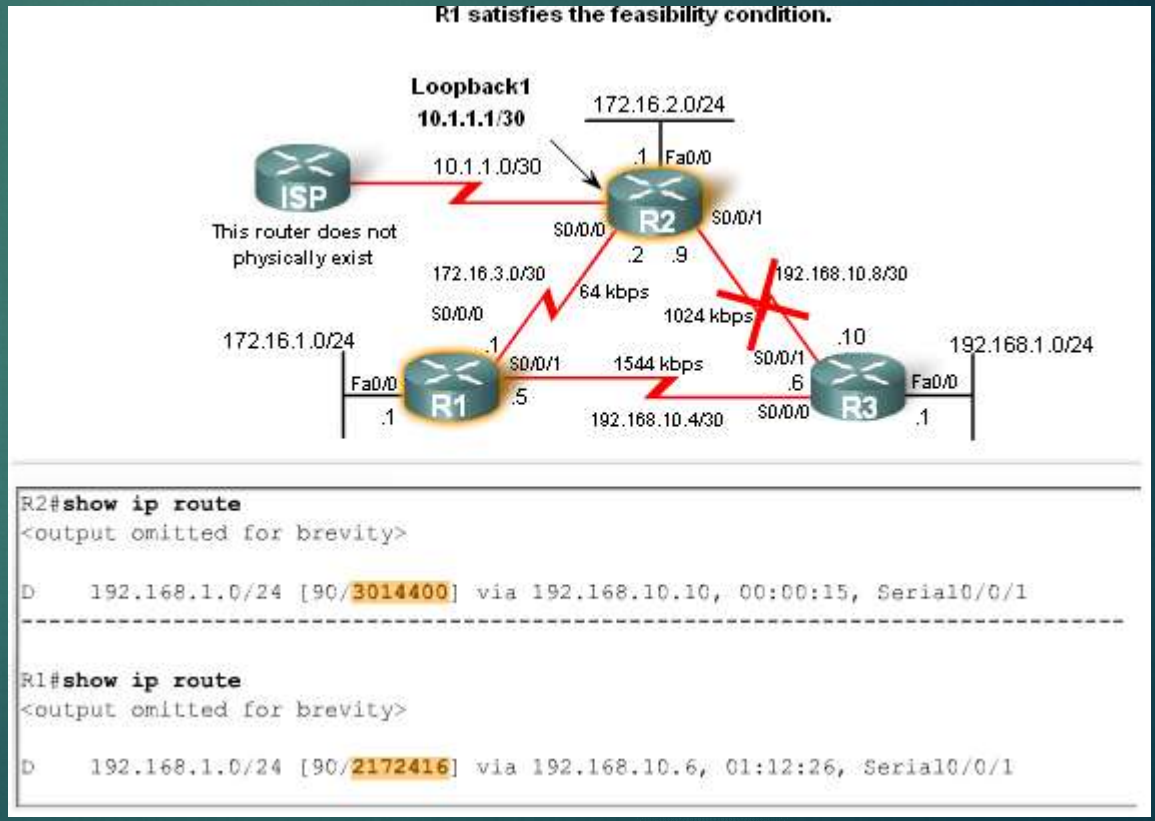



```
R2#show ip route
<code output omitted>

Gateway of last resort is not set

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

# DUAL Concepts

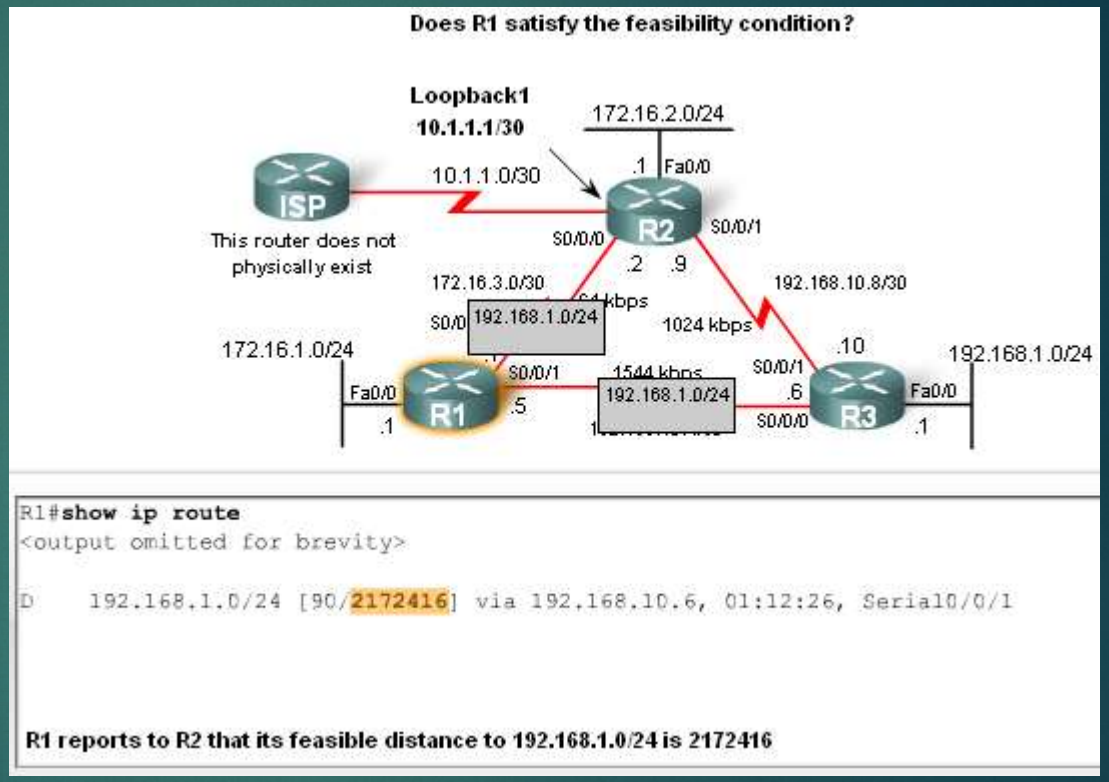
## Feasible Successors, Feasibility Condition & Reported Distance

- Reported distance (RD)
  - -The metric that a router reports to a neighbor about its own cost to that network



# DUAL Concepts

- Feasibility Condition (FC)
  - -Met when a neighbor's RD is less than the local router's FD to the same destination network



# DUAL Concepts

১. **Feasible Distance (FD):** কোন নেটওয়ার্কে যাওয়ার জন্য সর্বনিম্ন Cost ।

২. **Advertised Distance (AD):** একই নেটওয়ার্কে যাওয়ার জন্য Successor রাউটারের Cost ।

৩.**Successor (S):** কোন নেটওয়ার্কে যাওয়ার জন্য সবচেয়ে ভাল Neighbour রাউটার ।

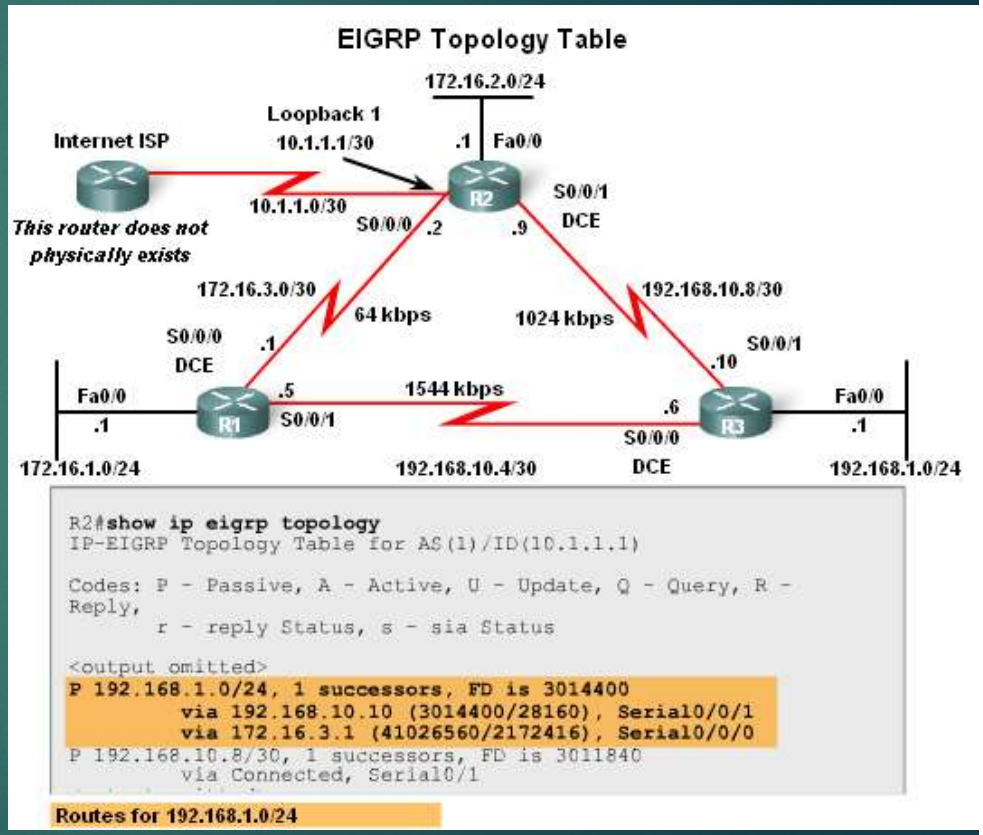
৪. **Feasible Successor (FS):** একই নেটওয়ার্কে যাওয়ার জন্য ব্যাকআপ Neighbor রাউটার ।

**Feasible Successor (FS)** হওয়ার শর্তঃ

শুধুমাত্র Advertised Distance জানানোর মাধ্যমেই কোন একটি রাউটার অন্য আরেকটি রাউটারের Feasible Successor হতে পারবে না। এ জন্য Feasible Successor এর AD ভ্যালু অবশ্যই Successor এর Feasible Distance অপেক্ষা ছোট হতে হবে। এ খানে, R4 রাউটার R1 এর Feasible Successor এজন্যই হতে পেরেছে

# DUAL Concepts

- Topology Table:  Successor & Feasible Successor
- EIGRP Topology table
  - Viewed using the *show ip eigrp topology* command
    - Contents of table include:
      - all successor routes
      - all feasible successor routes



# Summary

- It is an Interior Gateway Routing (IGP) protocol, means it can only be used to perform routing within the same autonomous system.
- It supports up to 255 hops count.
- It is used for a mid-sized network.
- It uses composite metric (bandwidth, load, reliability, delay, and MTU) to calculate the best path.
- It uses Diffusing Update Algorithm (DUAL).

# Summary

- Since it has the characteristics of both, the distance vector and link-state protocols. That's why sometimes it is referred as a hybrid protocol. However, I have seen many times that Cisco called it as an advanced distance-vector routing protocol.
- The **Administrative Distance** value of the EIGRP protocol is 90.
- Unlike IGRP, it is a classless routing protocol, hence, it supports CIDR and VLSM.

# Summary

- It also supports route summarization and discontinuous network.
- It supports up to six equal or unequal paths to provide load balancing for a single destination.
- The route update time of EIGRP protocol is 90 seconds.
- It supports trigger update which helps to reduce the size routing table.
- It uses **224.0.0.10 multicast address** to exchange routing information between (neighbors) routers.

# Summary

- **EIGRP commands**
  - The following commands are used for EIGRP configuration
    - RtrA(config)#router eigrp [autonomous-system #]
    - RtrA(config-router)#network *network-number*
  - The following commands can be used to verify EIGRP
    - Show ip protocols
    - Show ip eigrp neighbors
    - Show ip route
    - Show ip eigrp topolgy

# Summary

- **EIGRP metrics include**
  - Bandwidth (default)
  - Delay (default)
  - Reliability
  - Load

# Summary

- **DUAL**
  - Purpose of DUAL
    - To prevent routing loops
  - Successor
    - Primary route to a destination
  - Feasible successor
    - Backup route to a destination
  - Feasible distance
    - Lowest calculated metric to a destination
  - Reported distance
    - The distance towards a destination as advertised by an upstream neighbor

# Summary

- **Choosing the best route**
  - After router has received all updates from directly connected neighbors, it can calculate its DUAL
    - 1st metric is calculated for each route
    - 2nd route with lowest metric is designated successor & is placed in routing table
    - 3rd feasible successor is found
      - Criteria for feasible successor: it must have lower reported distance to the destination than the installed route's feasible distance
      - Feasible routes are maintained in topology table

# Summary

- **Automatic summarization**
  - On by default
  - Summarizes routes on classful boundary
  - Summarization can be disabled using the following command
    - RtrA(config-if)#no auto-summary

# BGP (Border Gateway Protocol)

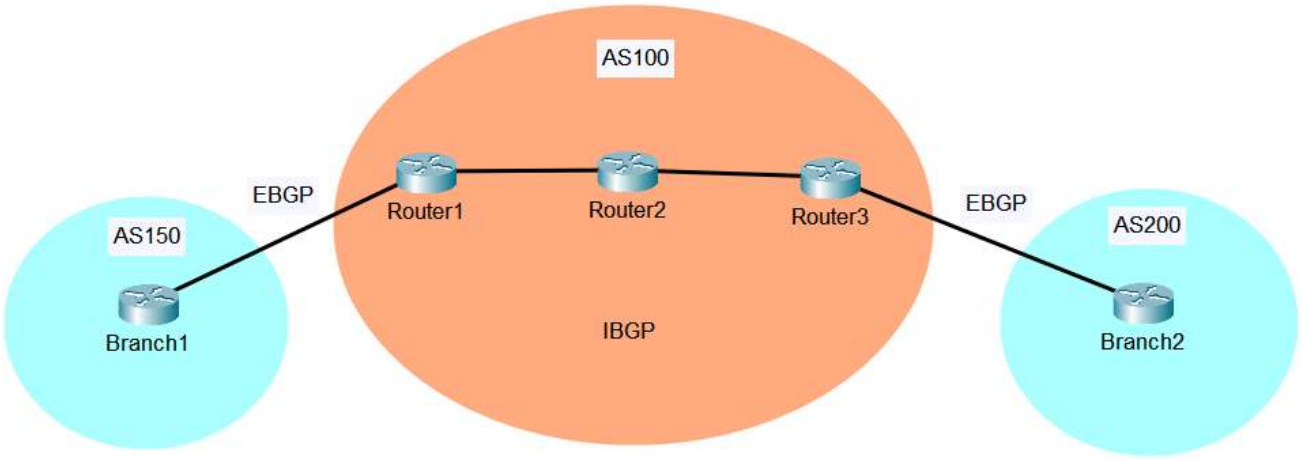
## BGP Session Types: iBGP vs eBGP

Border Gateway Protocol (BGP) has two sessions types, internal BGP (iBGP) and external BGP (eBGP). These BGP sessions are used depending on the Autonomous System of a BGP router. Let's see iBGP vs. eBGP below.

## Internal BGP (iBGP)

BGP sessions within the same Autonomous System (AS) are established with iBGP. It allows a prefix from one AS to be advertised to another AS. In the example below, you can see that there are three routers in AS100. These three routers can form an iBGP peering with one another. We can have the following iBGP peers:

- Router1 and Router2 iBGP peer
- Router2 and Router3 iBGP peer
- Router1 and Router2 iBGP peer




Also, even if the BGP routers are not directly connected, they can still become an iBGP neighbor of one another as long as they're on the same AS. With that being said, iBGP can be used when multiple routing protocols are used in the Autonomous System.

For example, Router1 and Router2 use OSPF for their connectivity, and Router2 and Router3 use EIGRP. We can establish an iBGP connectivity between Router1 and Router3 to have transit connectivity to AS150 and AS200. The downside of this setup is that when Router2 on AS100 receives traffic from either AS150 or AS200, it wouldn't know where to forward the traffic.

You might be wondering, "why don't we use IGP, such as OSPF or EIGRP, on AS100 instead, and just redistribute the prefixes?" It is possible with a few prefixes. However, if

we have a full Internet routing table, IGP lacks the scalability to handle the number of routes. BGP also uses custom routing to identify the best route, whereas IGPs use metric. The best path chosen by BGP could be longer and would be deemed as suboptimal by IGPs. Moreover, BGP path attributes are maintained by BGP only as the prefixes are advertised from AS to another AS.

Establishing iBGP sessions between all of the neighbors in the AS creates a full mesh. The routes received from iBGP neighbors are not advertised to other iBGP neighbors, which avoids loops and allows proper forwarding between Autonomous Systems.

## External BGP (eBGP)

BGP sessions that are in different Autonomous Systems are established with eBGP. In the same example topology above, eBGP is used to exchange network prefixes between AS150, AS100, and AS200. We would need the following eBGP peers:

- Branch1 and Router1 eBGP peer
- Branch2 and Router3 eBGP peer

For loop prevention, EBGP copes using the AS_Path. The advertising BGP router adds its ASN at the beginning of the existing AS_Path variable. Then, the receiving BGP router verifies if the ASN in the AS_Path variable doesn't match any of the local routers. If it does, BGP discards the Network Layer Reachability Information (NLRI), which contains the prefix.

## iBGP vs eBGP

Listed below are the difference between iBGP and eBGP.

1. Administrative Distance – upon the installation of an eBGP prefix in the routing table, it will be assigned with an Administrative Distance (AD) of 20. With iBGP, the AD will be 200.
2. Time-to-live (TTL) – eBGP packets have 1 TTL value by default. It causes packets to be dropped in transit, preventing a multi-hop BGP session. iBGP packets have a TTL value of 255, thus allowing multi-hop BGP sessions.
3. When advertising routes to an eBGP neighbor, the next hop address is changed to the IP address of the local router. With iBGP, the next hop address is unchanged.
4. The eBGP and iBGP sessions are configured similarly, except for the ASN in the remote-as statement, which is different from the ASN defined in the BGP process.

# BGP Message Types: Open, Keepalive, Update, Notification

There are four BGP message types used for communication, namely, OPEN, KEEPALIVE, UPDATE, and NOTIFICATION messages. Each message type is utilized differently by BGP. Listed below are the different BGP message types.

## Open Message

The BGP OPEN message is used to set up and establish BGP neighbor adjacency. An OPEN message includes information on the BGP router, and these must be negotiated and accepted by both BGP routers before they can exchange routing information. The BGP router information comprises the following:

**BGP Version Number** – the BGP version which the router is using. BGP version 4 is the latest version. If the two BGP routers have a version mismatch, then no BGP session will be made.

**AS Number** – the AS number must match the originating BGP router's AS number. This specifies if the BGP routers will run iBGP or eBGP as well.

**Hold Time** – it ensures that the BGP neighbor is 'alive.' By default, Cisco routers have 180 seconds hold time value. If the routers' hold time values are different, the lowest hold time value will be used. The minimum hold time value is 3 seconds and to disable KEEPALIVE messages, the hold time value is set to 0.

If the BGP router doesn't receive any UPDATE or KEEPALIVE messages from the BGP neighbor during the hold time, then it will claim that the neighbor is 'dead.' It will tear down the BGP session, the routes from the 'dead' neighbor are removed, and an UPDATE message with route withdrawal is sent to the other BGP routers for the affected prefixes. If the router does receive an UPDATE or KEEPALIVE message, then the hold timer will be reset to the initial value.

**BGP Identifier (RID)** – the BGP router ID (RID) identifies the BGP router in the advertised prefixes. It is a 32-bit unique number and it can be used to prevent loops for the routers that are advertised within the autonomous system (AS). The RID value must not be zero in order to form a neighbor adjacency. It can be set manually using the '**bgp router-id'** command. If the RID is n0t manually defined, it can dynamically use the highest loopback IP address, and if no loopback interface is configured, it will use the highest IP address on a physical interface.

**Optional Parameters** – these parameters establish the session capabilities of the BGP router. New features can be added to BGP even without having to create a new version by using this field.

## Keepalive Message

KEEPALIVE messages ensure that BGP neighbors are still alive. These messages are sent every one-third of the negotiated hold time value of the two BGP routers. By default, Cisco devices have a hold time of 180 seconds. One-third of 180 is 60, so the default KEEPALIVE message interval is 60 seconds.

If a BGP neighbor misses the three KEEPALIVE intervals, 180 seconds by default (60 x 3 = 180), the routes from that neighbor will be flushed from the other BGP router. If the hold time value is zero, no KEEPALIVE messages will be sent between the BGP peers.

## Update Message

UPDATE messages are used for advertising and exchanging routing information between BGP neighbors. The advertised prefix or the Network Layer Reachability Information (NLRI) information is included in the UPDATE message. The UPDATE message is also used in withdrawing advertised BGP routes, and it includes just the prefix only in the message. UPDATE messages also act as keepalives to lessen unnecessary traffic.

## Notification Message

The last of the BGP message types, NOTIFICATION messages will be sent if errors are detected in the BGP session. When a NOTIFICATION message is sent, the BGP neighbor adjacency will be terminated, and the BGP connection will be closed. The TCP session and the BGP table will be cleared of all entries from the BGP neighbor. Route withdrawals are done by sending UPDATE messages which will be sent to the other BGP peer/s.

## BGP States: Understanding BGP Neighbor Adjacency

Border Gateway Protocol (BGP) routers establish neighbor adjacency before being able to exchange routing information. However, unlike other routing protocols which use multicast or broadcast to discover their neighbors, BGP neighbors are manually

configured. BGP forms a BGP session through a TCP connection on TCP Port 179 with its peers or neighbor routers. The BGP Finite State Machine (FSM) is used to maintain the BGP table, which contains the peers and operational status. To establish a BGP session, the BGP FSM may take the router through the different BGP states. Listed below are six BGP states.

## Idle State

Idle is BGP's first state. If BGP detects a start event where a new BGP neighbor is configured or an established BGP peering is reset, BGP will initialize some resources and reset the ConnectRetryTimer. Then, it tries to initiate a TCP connection to the BGP peer. It will also listen for a new connection established by a BGP peer router. If BGP succeeds in this stage, it will move to Connect state.

If it fails, BGP will stay in an Idle state. The ConnectRetryTimer is then set to 60 seconds and it should decrement to zero for the connection to be initiated again. If it fails again, the previous ConnectRetryTimer will be doubled and should be decremented to zero for a new connection to be initiated again.

## Connect State

In this state, BGP waits for the three-way TCP handshake to be completed. If it succeeds, the ConnectRetryTimer will be reset by the established BGP session process. An OPEN message to the neighbor will be sent and will proceed to the OpenSent state.

If it fails, the state will continue to the Active state. If the ConnectRetry timer gets to zero and the Connect stage is not yet completed, the ConnectRetryTimer will be reset and BGP will attempt a new three-way TCP handshake. If some other things happen, such as BGP being reset, then the state will go back to Idle.

## Active State

BGP attempts a new three-way TCP handshake and establishes a connection with the BGP neighbor in this state. If it succeeds, an OPEN message will be sent to the neighbor and the hold timer will be set to 4 minutes. Then, the state will be changed to OpenSent.

If the TCP connection fails and/or the ConnectRetryTimer gets depleted, the state will return back to Connect state. The ConnectRetryTimer will be reset as well. Again, if some other things happen, such as the BGP process being reset, the state will go back to Idle.

## OpenSent State

After sending an OPEN message to the neighbor, BGP waits for an OPEN message from the BGP neighbor as well. The OPEN messages are both checked and compared for errors, such as:

- BGP version numbers should match.
- The OPEN message AS number must match the BGP neighbor's AS number.
- The OPEN message source IP address must match the BGP neighbor's IP address.
- BGP Identifiers, Router ID (RID), must exist and should be unique.
- Security Parameters

If there are no errors on the OPEN messages, BGP will send a KEEPALIVE message. The hold time is also negotiated using the lowest value between the two BGP routers. Then, the state will be moved to OpenConfirm.

If an error is found, a NOTIFICATION message will be sent, and the state will return to Idle. If the TCP session fails and gets disconnected, BGP will close the TCP connection and will reset the ConnectRetryTimer. The state will be back to an Active state. If any other event happens, the state will be moved to Idle.

## OpenConfirm State

In this state, BGP waits for KEEPALIVE messages from the BGP neighbor. If the BGP router receives a KEEPALIVE message, the state will move to an Established state and the BGP neighbor adjacency will be complete. If this happens, the hold timer will be reset as well.

If a NOTIFICATION message is received, or if the hold timer gets depleted, or if a stop event manifests, then the state will be back to Idle.

## Established State

Finally, the last of the BGP states, the established state, is where the BGP neighbor adjacency is established. The BGP peers or BGP neighbors send UPDATE messages to exchange routing information. When UPDATE and KEEPALIVE BGP messages are received, the hold timer will be reset. If a NOTIFICATION message is received or if the hold timer expires, the state will move back to Idle.

In this lab we will learn a simple eBGP (two BGP routers with different Autonomous System numbers) configuration between two routers with the topology below:


First we need to configure some interfaces on two routers as follows:

| | |
|---|---|
| R1(config)#interface fastethernet0/0<br>R1(config-if)#ip address 11.0.0.1 255.255.255.0<br>R1(config-if)#no shutdown<br>R1(config-if)#interface loopback 0<br>R1(config-if)#ip address 1.1.1.1 255.255.255.0 | R2(config)#interface fastethernet0/0<br>R2(config-if)#ip address 11.0.0.2 255.255.255.0<br>R2(config-if)#no shutdown<br>R2(config-if)#interface loopback 0<br>R2(config-if)#ip address 2.2.2.2 255.255.255.0 |

So we have just configured interface fa0/0 and loopback0 on both routers. Next we will configure the BGP configuration part on R1:

```
R1(config)#router bgp 1
R1(config-router)#neighbor 11.0.0.2 remote-as 2
```

The configuration is very simple with only two lines on R1. In the first line, BGP configuration begins with a familiar type of command: the **router bgp** command, where **AS number** is the BGP AS number used by that router (same as EIGRP, OSPF configuration).

The next command defines the IP address of the neighbor. Unlike OSPF or EIGRP, BGP cannot discover its neighbors automatically so we have to explicitly declare them. We also have to know and declare the neighbor's BGP AS number as well. In this case R1 wants to establish BGP neighbor relationship with R2 (in BGP AS 2) so it choose an interface on R2 (Fa0/0: **11.0.0.2**) and specify R2 is in **BGP AS 2** via the command "neighbor **11.0.0.2** remote-as **2**". At the other end R2 will do the same thing for R1 to set up BGP neighbor relationship.

```
R2(config)#router bgp 2
R2(config-router)#neighbor 11.0.0.1 remote-as 1
```

After a moment we should see a message (on each router) similar to the following, letting us know that an adjacency has been formed:

On R1:

```
*Aug 17 00:09:38.453: %BGP-5-ADJCHANGE: neighbor 11.0.0.2 Up
```

On R2:

```
*Aug 17 00:09:38.453: %BGP-5-ADJCHANGE: neighbor 11.0.0.1 Up
```

So after forming BGP neighbor relationship we can verify by using the "show ip bgp summary" command on both routers:

```
R1#show ip bgp summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
```

```
11.0.0.2        4     2      19      19        1    0    0 00:16:21        0
R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 2
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
11.0.0.1        4     1      20      20        1    0    0 00:17:13        0
```

Please pay attention to the "State/PfxRcd" column of the output. It indicates the number of prefixes that have been received from a neighbor. If this value is a number (including "0", which means BGP neighbor does not advertise any route) then the BGP neighbor relationship is good. If this value is a word (including "Idle", "Connect", "Active", "OpenSent", "OpenConfirm") then the BGP neighbor relationship is not good.

In the outputs above we see the BGP neighbor relationship between R1 & R2 is good with zero Prefix Received (PfxRcd) because they have not advertised any routes yet.

How about the BGP routing table? We can check with the "show ip bgp" command but currently this table is empty! This is because although they formed BGP neighbor relationship but they have not exchanged any routes. Let's try advertising the loopback 0 interface on R1 to R2:

```
R1(config-router)#network 1.1.1.0 mask 255.255.255.0
```

As you see, unlike other routing protocols like OSPF or EIGRP, we have to use subnet mask (255.255.255.0 in this case), not wildcard mask, to advertise the routes in the "network" command.

Note: With BGP, you must advertise the correct network and subnet mask in the "network" command ( in this case network 1.1.1.0/24). BGP is very strict in the routing advertisements. In other words, BGP only advertises the network which exists exactly in the routing table (in this case network 1.1.1.0/24 exists in the routing table as the loopback 0 interface). If you put the command "network 1.1.0.0 mask 255.255.0.0" or "network 1.0.0.0 mask 255.0.0.0" or "network 1.1.1.1 mask 255.255.255.255" then BGP will not advertise anything.

Now the BGP routing tables on these two routers contain this route:

```
R1#sh ip bgp
BGP table version is 4, local router ID is 11.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf  Weight   Path
*> 1.1.1.0/24       0.0.0.0                  0          32768   i
R2#sh ip bgp
BGP table version is 2, local router ID is 11.0.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf  Weight   Path
*> 1.1.1.0/24       11.0.0.1                 0              0   1 i
```

An asterisk (*) in the first column means that the route has a valid next hop. A greater-than sign (>) indicates the route has been selected as the best path to that network.

# What is NAT?

- NAT (Network Address Translation) is a process of changing the source and destination IP addresses and ports. Address translation reduces the need for IPv4 public addresses and hides private network address ranges. The process is usually done by routers or firewalls.

- There are three types of address translation:

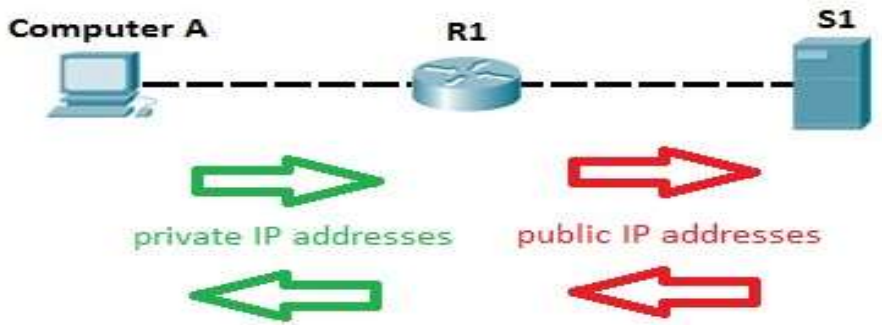
  1. Static NAT – translates one private IP address to a public one. The public IP address is always the same.

  2. Dynamic NAT – private IP addresses are mapped to the pool of public IP addresses.

  3. Port Address Translation (PAT) – one public IP address is used for all internal devices, but a different port is assigned to each private IP address. Also known as NAT Overload.

# How NAT Worked?

- For example on how NAT are worked, consider the following network topology



# How NAT Worked?

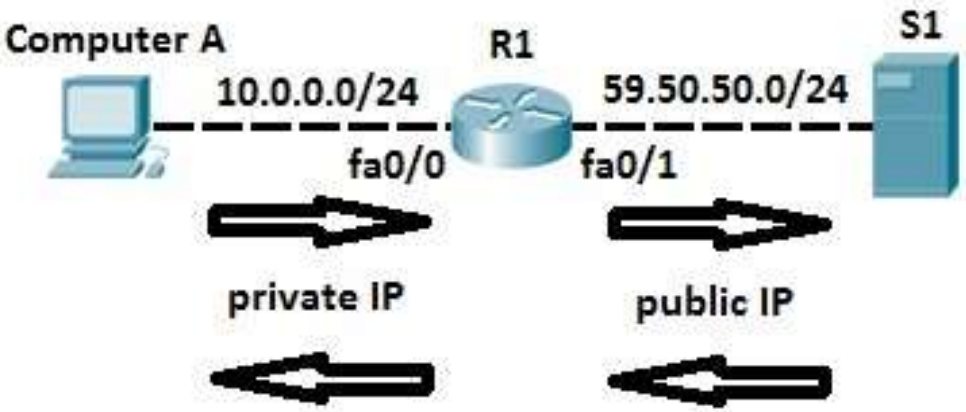
- Computer A request a web page from an Internet server. Because Computer A uses private IP addressing, the source address of the request has to be changed by the router because private IP addresses are not routable on the Internet. Router R1 receives the request, changes the source IP address to its public IP address and sends the packet to server S1. Server S1 receives the packet and replies to router R1. Router R1 receives the packet, changes the destination IP addresses to the private IP address of Computer A and sends the packet to Computer A.

# Static NAT

- With static NAT, routers or firewalls translate one private IP address to a single public IP address. Each private IP address is mapped to a single public IP address. Static NAT is not often used because it requires one public IP address for each private IP address.
- To configure static NAT, three steps are required:

1. configure private/public IP address mapping by using the *ip nat inside source static PRIVATE_IP PUBLIC_IP*command
2. configure the router's inside interface using the *ip nat inside* command
3. configure the router's outside interface using the *ip nat outside* command

# Static NAT Example



# Static NAT Example

- Computer A requests a web resource from S1. Computer A uses its private IP address when sending the request to router R1. Router R1 receives the request, changes the private IP address to the public one and sends the request to S1. S1 responds to R1. R1 receives the response, looks up in its NAT table and changes the destination IP address to the private IP address of Computer A.
- In the example above, we need to configure static NAT. To do that, the following commands are required on R1:

```
R1(config)#ip nat inside source static 10.0.0.2 59.50.50.1
R1(config)#int fa0/0
R1(config-if)#ip nat inside
R1(config-if)#int fa0/1
R1(config-if)#ip nat outside
```

# Static NAT Example

- Using the commands above, we have configured a static mapping between Computer A's private IP address of 10.0.0.2 and router's R1 public IP address of 59.50.50.1. To check NAT, you can use the *show ip nat translations* command:

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 59.50.50.1:9      10.0.0.2:9         59.50.50.2:9       59.50.50.2:9
---  59.50.50.1        10.0.0.2           ---                ---
```

# Dynamic NAT

- With dynamic NAT, We can specify two sets of addresses on Cisco router:

1. Inside addresses that will be translated.
2. A pool of global addresses.

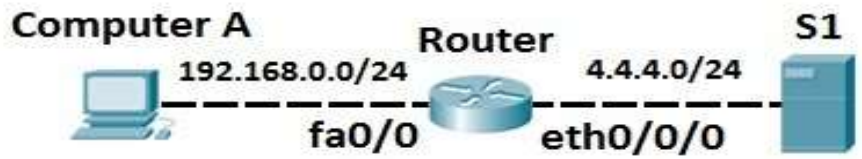
- Unlike with static NAT, where we had to manually define a static mapping between a private and a public address, with dynamic NAT the mapping of a local address to a global address happens dynamically. This means that the router dynamically picks an address from the global address pool that is not currently assigned. It can be any address from the pool of global addresses. The dynamic entry stays in the NAT translations table as long as the traffic is exchanged. The entry times out after a period of inactivity and the global IP address can be used for new translations.

# Dynamic NAT

- To configure dynamic NAT, the following steps are required:

  1. configure the router's inside interface using the *ip nat inside* command
  2. configure the router's outside interface using the *ip nat outside* command
  3. configure an ACL that has a list of the inside source addresses that will be translated
  4. configure the pool of global IP addresses using the *ip nat pool NAME FIRST_IP_ADDRESS LAST_IP_ADDRESS netmask SUBNET_MASK* command
  5. enable dynamic NAT with the *ip nat inside source list ACL_NUMBER pool NAME* global configuration command

# Dynamic NAT Example



# Dynamic NAT Example

- Computer A requests a web resource from S1. Computer A uses its private IP address when sending the request to router R1. Router R1 receives the request, changes the private IP address to one of the available global addresses in the pool and sends the request to S1. S1 responds to R1. R1 receives the response, looks up in its NAT table and changes the destination IP address to the private IP address of Computer A.

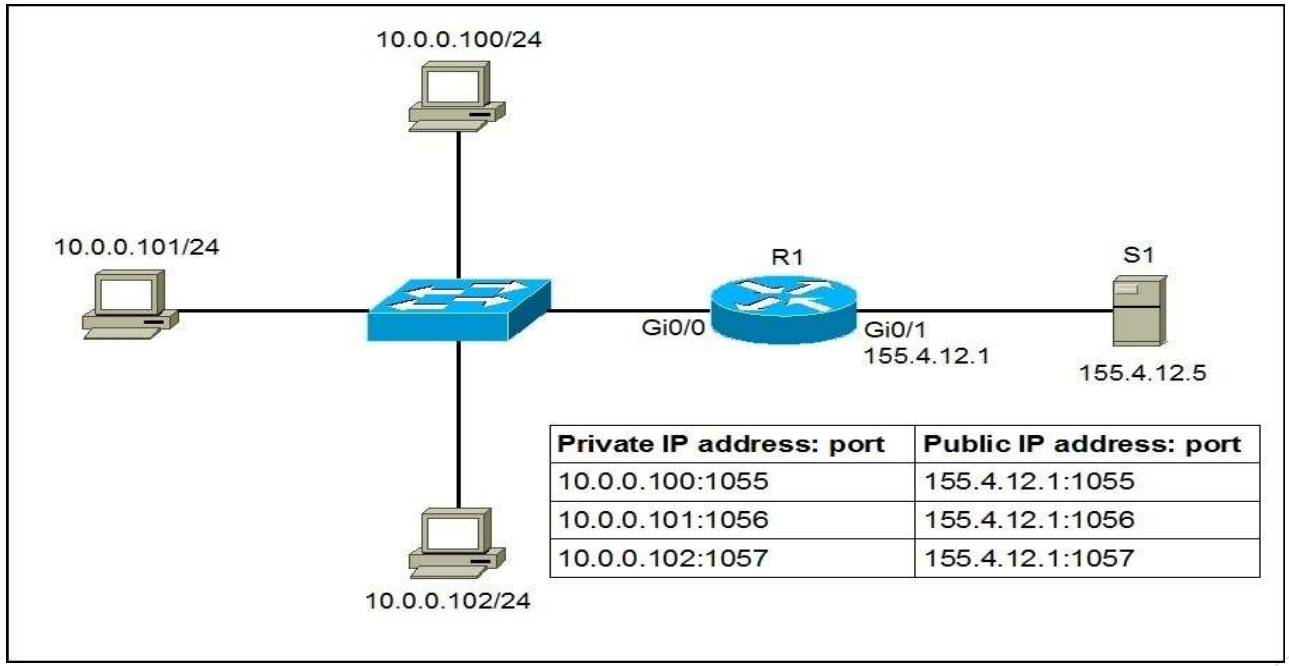
# Port Address Translation (PAT)

- With Port Address Translation (PAT), a single public IP address is used for all internal private IP addresses, but a different port is assigned to each private IP address. This type of NAT is also known as NAT Overload and is the typical form of NAT used in today's networks. It is even supported by most consumer-grade routers.

- PAT allows you to support many hosts with only few public IP addresses. It works by creating dynamic NAT mapping, in which a global (public) IP address and a unique port number are selected. The router keeps a NAT table entry for every unique combination of the private IP address and port, with translation to the global address and a unique port number.

# PAT Example

- We will use the following example network to explain the benefits of using PAT:




| Private IP address: port | Public IP address: port |
|---|---|
| 10.0.0.100:1055 | 155.4.12.1:1055 |
| 10.0.0.101:1056 | 155.4.12.1:1056 |
| 10.0.0.102:1057 | 155.4.12.1:1057 |

# PAT Example

- As you can see in the picture above, PAT uses unique source port numbers on the inside global (public) IP address to distinguish between translations. For example, if the host with the IP address of 10.0.0.101 wants to access the server S1 on the Internet, the host's private IP address will be translated by R1 to 155.4.12.1:1056 and the request will be sent to S1. S1 will respond to 155.4.12.1:1056. R1 will receive that response, look up in its NAT translation table, and forward the request to the host.
- To configure PAT, the following commands are required:

1. Configure the router's inside interface using the *"ip nat inside"* command.
2. Configure the router's outside interface using the *"ip nat outside"* command.
3. Configure an access list that includes a list of the inside source addresses that should be translated.
4. enable PAT with the *ip nat inside source list ACL_NUMBER interface TYPE overload* global configuration command.

Any Question?