IPRC KARONGI

Integrated Polytechnic Regional College

P.O. Box85 KARONGI- RWANDA
Tel: +250 788871075
Email: info@iprckarongi.rp.ac.rw
Website: www.iprckarongi.rp.ac.rw

Cisco | Networking Academy®
Mind Wide Open™

Department: Information and Communication Technology

Option: Information Technology

Class: Level 2

Academic Year: 2020-2021

Semester: 2

Module Title: ADVANCED COMPUTER NETWORKING

Module code: ICT210

Credits: 15

Students Syllabus

Module Leader

Names: SAFARI N. Cyprianho

Phone: +250788392430

Email: sacyprianho@yahoo.co.uk

CISCO IT Instructor

CONTENTS

CHAPTER 1: ROUTING CONCEPTS

Networks allow people to communicate, collaborate, and interact in many ways. Networks are used to access web pages, talk using IP telephones, participate in video conferences, compete in interactive gaming, shop using the Internet, complete online coursework, and more.

Ethernet switches function at the data link layer, Layer 2, and are used to forward Ethernet frames between devices within the same network.

However, when the source IP and destination IP addresses are on different networks, the Ethernet frame must be sent to a router.

A router connects one network to another network. The router is responsible for the delivery of packets across different networks. The destination of the IP packet might be a web server in another country or an email server on the local area network.

The router uses its routing table to determine the best path to use to forward a packet. It is the responsibility of the routers to deliver those packets in a timely manner. The effectiveness of internetwork communications depends, to a large degree, on the ability of routers to forward packets in the most efficient way possible.

When a host sends a packet to a device on a different IP network, the packet is forwarded to the default gateway because a host device cannot communicate directly with devices outside of the local network. The default gateway is the destination that routes traffic from the local network to devices on remote networks. It is often used to connect a local network to the Internet.

Because the router can route packets between networks, devices on different networks can communicate. This chapter will introduce the router, its role in networks, its main hardware and software components, and the routing process. Exercises which demonstrate how to access the router, configure basic router settings, and verify settings will be provided.

1.1 Router Initial Configuration

1.1.1 Router Function

1.1.1.1 Characteristics of a Network

Networks have had a significant impact on our lives. They have changed the way we live, work, and play.

Networks allow us to communicate, collaborate, and interact in ways we never did before. We use the network in a variety of ways, including web applications, IP telephony, video conferencing, interactive gaming, electronic commerce, education, and more.

There are many key structures and performance-related characteristics referred to when discussing networks:

- **Topology** - There are physical and logical topologies. The physical topology is the arrangement of the cables, network devices, and end systems. It describes how the network devices are actually interconnected with wires and cables. The logical topology is the path over which the data is transferred in a network. It describes how the network devices appear connected to network users.
- **Speed** - Speed is a measure of the data rate in bits per second (b/s) of a given link in the network.
- **Cost** - Cost indicates the general expense for purchasing of network components, and installation and maintenance of the network.
- **Security** - Security indicates how protected the network is, including the information that is transmitted over the network. The subject of security is important, and techniques and practices are constantly evolving. Consider security whenever actions are taken that affect the network.
- **Availability** - Availability is the likelihood that the network is available for use when it is required.

- **Scalability** - Scalability indicates how easily the network can accommodate more users and data transmission requirements. If a network design is optimized to only meet current requirements, it can be very difficult and expensive to meet new needs when the network grows.
- **Reliability** - Reliability indicates the dependability of the components that make up the network, such as the routers, switches, PCs, and servers. Reliability is often measured as a probability of failure or as the mean time between failures (MTBF).

These characteristics and attributes provide a means to compare different networking solutions.

Note: While the term "speed" is commonly used when referring to the network bandwidth, it is not technically accurate. The actual speed that the bits are transmitted does not vary over the same medium. The difference in bandwidth is due to the number of bits transmitted per second, not how fast they travel over wire or wireless medium.

1.1.1.2 Why Routing?

How does clicking a link in a web browser return the desired information in mere seconds? Although there are many devices and technologies collaboratively working together to enable this, the primary device is the router. Stated simply, a router connects one network to another network.

Communication between networks would not be possible without a router determining the best path to the destination and forwarding traffic to the next router along that path. The router is responsible for the routing of traffic between networks.

The routers interconnect the networks at the different sites. When a packet arrives on a router interface, the router uses its routing table to determine how to reach the destination network. The destination of the IP packet might be a web server in another country or an email server on the local area network. It is the responsibility of routers to deliver those packets efficiently. The effectiveness of internetwork communications depends, to a large degree, on the ability of routers to forward packets in the most efficient way possible.

1.1.1.3 Routers are Computers

Most network capable devices (e.g.., computers, tablets, and smartphones) require the following components to operate:

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

A router is essentially a specialized computer. It requires a CPU and memory to temporarily and permanently store data to execute operating system instructions, such as system initialization, routing functions, and switching functions.

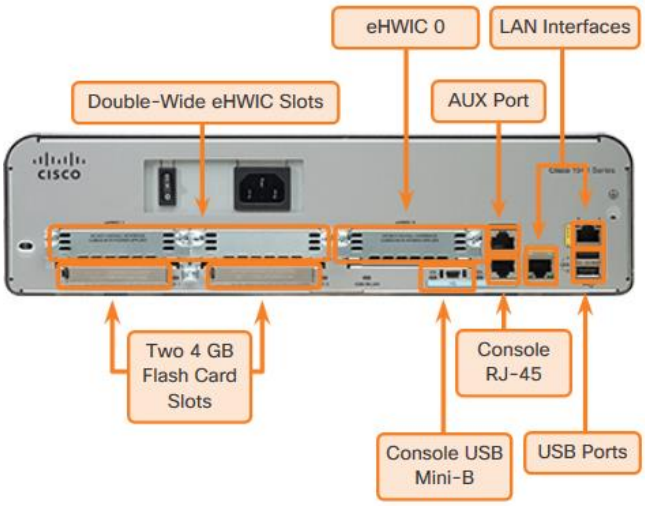
Note: Cisco devices use the Cisco Internetwork Operating System (IOS) as the system software.

Router memory is classified as volatile or non-volatile. Volatile memory loses its content when the power is turned off, while non-volatile memory does not lose its content when the power is turned off.

The table below summarizes the types of router memory, the volatility, and examples of what is stored in each.

Memory	Description
Random Access Memory (RAM)	Volatile memory that provides temporary storage for various applications and processes including: • Running IOS • Running configuration file • IP routing and ARP tables • Packet buffer
Read-Only Memory (ROM)	Non-volatile memory that provides permanent storage for: • Bootup instructions • Basic diagnostic software • Limited IOS in case the router cannot load the full featured IOS
Non-Volatile Random Access Memory (NVRAM)	Non-volatile memory that provides permanent storage for the: • Startup configuration file
Flash	Non-volatile memory that provides permament storage for: • IOS • Other system-related files

Unlike a computer, a router does not have video adapters or sound card adapters. Instead, routers have specialized ports and network interface cards to interconnect devices to other networks. Figure below identifies some of these ports and interfaces.



1.1.1.4 Routers Interconnect Networks

Most users are unaware of the presence of numerous routers on their own network or on the Internet. Users expect to be able to access web pages, send emails, and download music, regardless of whether the server accessed is on their own network or on another network. Networking professionals know that it is the router that is responsible for forwarding packets from network to network, from the original source to the final destination.

A router connects multiple networks, which means that it has multiple interfaces that each belong to a different IP network. When a router receives an IP packet on one interface, it determines which interface to use to forward the packet to the destination. The interface that the router uses to forward the packet may be the final destination, or it may be a network connected to another router that is used to reach the destination network.

Each network that a router connects to typically requires a separate interface. These interfaces are used to connect a combination of both local-area networks (LANs) and wide-area networks (WANs). LANs are commonly Ethernet networks that contain devices, such as PCs, printers, and servers. WANs are used to connect networks over a large geographical area. For example, a WAN connection is commonly used to connect a LAN to the Internet service provider (ISP) network.

1.1.1.5 Routers Choose Best Paths

The primary functions of a router are to:

- Determine the best path to send packets
- Forward packets toward their destination

The router uses its routing table to determine the best path to use to forward a packet. When the router receives a packet, it examines the destination address of the packet and uses the routing table to search for the best path to that network. The routing table also includes the interface to be used to forward packets for each known network. When a match is found, the router encapsulates the packet into the data link frame of the outgoing or exit interface, and the packet is forwarded toward its destination.

It is possible for a router to receive a packet that is encapsulated in one type of data link frame, and to forward the packet out of an interface that uses a different type of data link frame. For example, a router may receive a packet on an Ethernet interface, but must forward the packet out of an interface configured with the Point-to-Point Protocol (PPP). The data link encapsulation depends on the type of interface on the router and the type of medium to which it connects. The different data link technologies that a router can connect to include Ethernet, PPP, Frame Relay, DSL, cable, and wireless (802.11, Bluetooth, etc.).

Note: Routers use static routes and dynamic routing protocols to learn about remote networks and build their routing tables.

1.1.1.6 Packet Forwarding Mechanisms

Routers support three packet-forwarding mechanisms:

- **Process switching** - An older packet forwarding mechanism still available for Cisco routers. When a packet arrives on an interface, it is forwarded to the control plane where the CPU matches the destination address with an entry in its routing table, and then determines the exit interface and forwards the packet. It is important to understand that the router does this for every packet, even if the destination is the same for a stream of packets. This process-switching mechanism is very slow and rarely implemented in modern networks.
- **Fast switching** - This is a common packet forwarding mechanism which uses a fast-switching cache to store next-hop information. When a packet arrives on an interface, it is forwarded to the control plane where the CPU searches for a match in the fast-switching cache. If it is not there, it is process-switched and forwarded to the exit interface. The flow information for the packet is also stored in the fast-switching cache. If another packet going to the same destination arrives on an interface, the next-hop information in the cache is re-used without CPU intervention.
- **Cisco Express Forwarding (CEF)** - CEF is the most recent and preferred Cisco IOS packet-forwarding mechanism. Like fast switching, CEF builds a Forwarding Information Base (FIB), and an adjacency table. However, the table entries are not packet-triggered like fast switching but change-triggered such as when something changes in the network topology. Therefore, when a network has converged, the FIB and adjacency tables contain all the information a router would have to consider when forwarding a packet. The FIB contains pre-computed reverse lookups, next hop information for routes including the interface and Layer 2 information. Cisco Express Forwarding is the fastest forwarding mechanism and the preferred choice on Cisco routers.

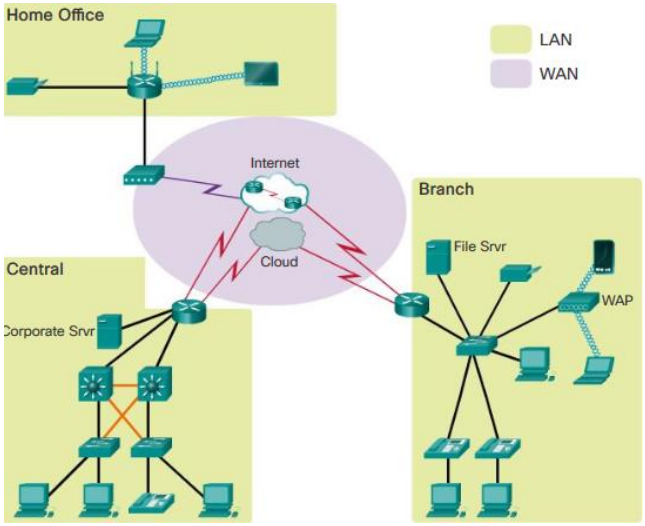
A common analogy used to describe the three packet-forwarding mechanisms is as follows:

- Process switching solves a problem by doing math long hand, even if it is the identical problem.
- Fast switching solves a problem by doing math long hand one time and remembering the answer for subsequent identical problems.
- CEF solves every possible problem ahead of time in a spreadsheet.

1.1.2 Connect Devices

1.1.2.1 Connect to a Network

Network devices and end users typically connect to a network using a wired Ethernet or wireless connection. Refer to the figure as a sample reference topology. The LANs in the figure serve as an example of how users and network devices could connect to networks.



Home Office devices can connect as follows:

- Laptops and tablets connect wirelessly to a home router.
- A network printer connects using an Ethernet cable to the switch port on the home router.
- The home router connects to the service provider cable modem using an Ethernet cable.
- The cable modem connects to the Internet service provider (ISP) network.

The Branch site devices connect as follows:

- Corporate resources (i.e., file servers and printers) connect to Layer 2 switches using Ethernet cables.
- Desktop PCs and voice over IP (VoIP) phones connect to Layer 2 switches using Ethernet cables.
- Laptops and smartphones connect wirelessly to wireless access points (WAPs).
- The WAPs connect to switches using Ethernet cables.
- Layer 2 switches connect to an Ethernet interface on the edge router using Ethernet cables. An edge router is a device that sits at the edge or boundary of a network and routes between that network and another, such as between a LAN and a WAN.
- The edge router connects to a WAN service provider (SP).
- The edge router also connects to an ISP for backup purposes.

The Central site devices connect as follows:

- Desktop PCs and VoIP phones connect to Layer 2 switches using Ethernet cables.
- Layer 2 switches connect redundantly to multilayer Layer 3 switches using Ethernet fiber-optic cables (orange connections).
- Layer 3 multilayer switches connect to an Ethernet interface on the edge router using Ethernet cables.
- The corporate website server is connected using an Ethernet cable to the edge router interface.
- The edge router connects to a WAN SP.
- The edge router also connects to an ISP for backup purposes.

In the Branch and Central LANs, hosts are connected either directly or indirectly (via WAPs) to the network infrastructure using a Layer 2 switch.

1.1.2.2 Default Gateways

To enable network access, devices must be configured with IP address information to identify the appropriate:

- **IP address** - Identifies a unique host on a local network.
- **Subnet mask** - Identifies with which network subnet the host can communicate.
- **Default gateway** - Identifies the IP address of the router to send a packet to when the destination is not on the same local network subnet.

When a host sends a packet to a device that is on the same IP network, the packet is simply forwarded out of the host interface to the destination device.

When a host sends a packet to a device on a different IP network, then the packet is forwarded to the default gateway, because a host device cannot communicate directly with devices outside of the local network. The default gateway is the destination that routes traffic from the local network to devices on remote networks. It is often used to connect a local network to the Internet.

The default gateway is usually the address of the interface on the router connected to the local network. The router maintains routing table entries of all connected networks as well as entries of remote networks, and determines the best path to reach those destinations.

Note: A router is also usually configured with its own default gateway. This is known as the Gateway of Last Resort.

1.1.2.3 Document Network Addressing

When designing a new network or mapping an existing network, document the network. At a minimum, the documentation should identify:

- Device names
- Interfaces used in the design
- IP addresses and subnet masks
- Default gateway addresses

This information is captured by creating two useful network documents:

- **Topology diagram** - Provides a visual reference that indicates the physical connectivity and logical Layer 3 addressing. Often created using software, such as Microsoft Visio.
- **An addressing table** - A table that captures device names, interfaces, IPv4 addresses, subnet masks, and default gateway addresses.

1.1.2.4 Enable IP on a Host

A host can be assigned IP address information either:

- **Statically** - The host is manually assigned the correct IP address, subnet mask, and default gateway. The DNS server IP address can also be configured.
- **Dynamically** - IP address information is provided by a server using the Dynamic Host Configuration Protocol (DHCP). The DHCP server provides a valid IP address, subnet mask, and default gateway for end devices. Other information may be provided by the server.

Statically assigned addresses are commonly used to identify specific network resources, such as network servers and printers. They can also be used in smaller networks with few hosts. However, most host devices acquire their IPv4 address information by accessing a DHCPv4 server. In large enterprises, dedicated DHCPv4 servers providing services to many LANs are implemented. In a smaller branch or small office setting, DHCPv4 services can be provided by a Cisco Catalyst switch or a Cisco ISR.

1.1.2.5 Device LEDs

Host computers connect to a wired network using a network interface and RJ-45 Ethernet cable. Most network interfaces have one or two LED link indicators next to the interface. Typically, a green LED means a good connection while a blinking green LED indicates network activity.

If the link light is not on, then there may be a problem with either the network cable or the network itself. The switch port where the connection terminates would also have an LED indicator lit. If one or both ends are not lit, try a different network cable.

Note: The actual function of the LEDs varies between computer manufacturers.

Similarly, network infrastructure devices commonly use multiple LED indicators to provide a quick status view. For example, a Cisco Catalyst 2960 switch has several status LEDs to help monitor system activity and performance. These LEDs are generally lit green when the switch is functioning normally and lit amber when there is a malfunction.

Cisco ISRs use various LED indicators to provide status information. The LEDs on the router help the network administrator conduct some basic troubleshooting. Each device has a unique set of LEDs. Consult the device-specific documentation for an accurate description of the LEDs.

1.1.2.6 Console Access

In a production environment, infrastructure devices are commonly accessed remotely using Secure Shell (SSH) or HyperText Transfer Protocol Secure (HTTPS). Console access is really only required when initially configuring a device, or if remote access fails.

Console access requires:

- **Console cable** - RJ-45-to-DB-9 serial cable or a USB serial cable
- **Terminal emulation software** - Tera Term, PuTTY, HyperTerminal

The cable is connected between the serial port of the host and the console port on the device. Most computers and notebooks no longer include built-in serial ports. If the host does not have a serial port, the USB port can be used to establish a console connection. A special USB-to-RS-232 compatible serial port adapter is required when using the USB port.

The Cisco ISR G2 supports a USB serial console connection. To establish connectivity, a USB Type-A to USB Type-B (mini-B USB) is required, as well as an operating system device driver. This device driver is available from www.cisco.com. Although these routers have two console ports, only one console port can be active at a time. When a cable is plugged into the USB console port, the RJ-45 port becomes inactive. When the USB cable is removed from the USB port, the RJ-45 port becomes active.

1.1.2.7 Enable IP on a Switch

Network infrastructure devices require IP addresses to enable remote management. Using the device IP address, the network administrator can remotely connect to the device using Telnet, SSH, HTTP, or HTTPS.

A switch does not have a dedicated interface to which an IP address can be assigned. Instead, the IP address information is configured on a virtual interface called a switched virtual interface (SVI).

1.1.3 Router Basic Settings

1.1.3.1 Configure Basic Router Settings

Cisco routers and Cisco switches have many similarities. They support a similar modal operating system, similar command structures, and many of the same commands. In addition, both devices have similar initial configuration steps.

For instance, the following configuration tasks should always be performed:

- **Name the device** – Distinguishes it from other routers.
- **Secure management access** – Secures privileged EXEC, user EXEC, and remote access.
- **Configure a banner** – Provides legal notification of unauthorized access.

Always save the changes on a router and verify the basic configuration and router operations.

1.1.3.2 Configure an IPv4 Router Interface

One distinguishing feature between switches and routers is the type of interfaces supported by each. For example, Layer 2 switches support LANs and, therefore, have multiple FastEthernet or Gigabit Ethernet ports.

Routers support LANs and WANs and can interconnect different types of networks; therefore, they support many types of interfaces. For example, G2 ISRs have one or two integrated Gigabit Ethernet interfaces and High-Speed WAN Interface Card (HWIC) slots to accommodate other types of network interfaces, including serial, DSL, and cable interfaces.

To be available, an interface must be:

- **Configured with an IP address and a subnet mask** - Use the **ip address** *ip-address subnet-mask* interface configuration command.
- **Activated** - By default, LAN and WAN interfaces are not activated (**shutdown**). To enable an interface, it must be activated using the **no shutdown** command. (This is similar to powering on the interface.) The interface must also be connected to another device (a hub, a switch, or another router) for the physical layer to be active.

Optionally, the interface could also be configured with a short description of up to 240 characters. It is good practice to configure a description on each interface. On production networks, the benefits of interface descriptions are quickly realized as they are helpful in troubleshooting and to identify a third party connection and contact information.

Depending on the type of interface, additional parameters may be required. For example, in the lab environment, the serial interface connecting to the serial cable end labeled DCE must be configured with the **clock rate** command.

Note: Accidentally using the **clock rate** command on a DTE interface generates a "%Error: This command applies only to DCE interface" informational message.

1.1.3.3 Configure an IPv6 Router Interface

Configuring an IPv6 interface is similar to configuring an interface for IPv4. Most IPv6 configuration and verification commands in the Cisco IOS are very similar to their IPv4 counterparts. In many cases, the only difference is the use of **ipv6** in place of **ip** in commands.

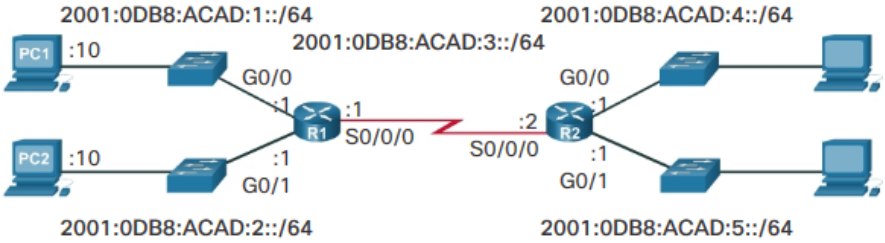
An IPv6 interface must be:

- **Configured with IPv6 address and subnet mask** - Use the **ipv6 address** *ipv6-address*/*prefix-length* **[link-local | eui-64]** interface configuration command.
- **Activated** - The interface must be activated using the **no shutdown** command.

Note: An interface can generate its own IPv6 link-local address without having a global unicast address by using the **ipv6 enable** interface configuration command.

Unlike IPv4, IPv6 interfaces will typically have more than one IPv6 address. At a minimum, an IPv6 device must have an IPv6 link-local address but will most likely also have an IPv6 global unicast address. IPv6 also supports the ability for an interface to have multiple IPv6 global unicast addresses from the same subnet. The following commands can be used to statically create a global unicast or link-local IPv6 address:

- **ipv6 address** *ipv6-address/prefix-length* - Creates a global unicast IPv6 address as specified.
- **ipv6 address** *ipv6-address/prefix-length* **eui-64** - Configures a global unicast IPv6 address with an interface identifier (ID) in the low-order 64 bits of the IPv6 address using the EUI-64 process.
- **ipv6 address** *ipv6-address/prefix-length* **link-local** - Configures a static link-local address on the interface that is used instead of the link-local address that is automatically configured when the global unicast IPv6 address is assigned to the interface or enabled using the **ipv6 enable** interface command. Recall, the **ipv6 enable** interface command is used to automatically create an IPv6 link-local address whether or not an IPv6 global unicast address has been assigned.



In the example topology shown in Figure above, R1 must be configured to support the following IPv6 network addresses:

- 2001:0DB8:ACAD:0001:/64 or equivalently 2001:DB8:ACAD:1::/64
- 2001:0DB8:ACAD:0002:/64 or equivalently 2001:DB8:ACAD:2::/64
- 2001:0DB8:ACAD:0003:/64 or equivalently 2001:DB8:ACAD:3::/64

When the router is configured using the **ipv6 unicast-routing** global configuration command, the router begins sending ICMPv6 Router Advertisement messages out the interface. This enables a PC connected to the interface to automatically configure an IPv6 address and to set a default gateway without needing the services of a DHCPv6 server. Alternatively, a PC connected to the IPv6 network can have an IPv6 address manually configured. Notice that the default gateway address configured for PC1 is the IPv6 global unicast address of the R1 GigabitEthernet 0/0 interface.

1.1.3.4 Configure an IPv4 Loopback Interface

Another common configuration of Cisco IOS routers is enabling a loopback interface.

The loopback interface is a logical interface internal to the router. It is not assigned to a physical port and can therefore never be connected to any other device. It is considered a software interface that is automatically placed in an "up" state, as long as the router is functioning.

The loopback interface is useful in testing and managing a Cisco IOS device because it ensures that at least one interface will always be available. For example, it can be used for testing purposes, such as testing internal routing processes, by emulating networks behind the router.

Additionally, the IPv4 address assigned to the loopback interface can be significant to processes on the router that use an interface IPv4 address for identification purposes, such as the Open Shortest Path First (OSPF) routing process. By enabling a loopback interface, the router will use the always available loopback interface address for identification, rather than an IP address assigned to a physical port that may go down.

```
Enabling and assigning a loopback address is simple:
Router(config)# interface loopback number
Router(config-if)# ip address ip-address subnet-mask
Router(config-if)# exit
```

Multiple loopback interfaces can be enabled on a router. The IPv4 address for each loopback interface must be unique and unused by any other interface.

1.1.4 Verify Connectivity of Directly Connected Networks

1.1.4.1 Verify Interface Settings

There are several **show** commands that can be used to verify the operation and configuration of an interface. The following three commands are especially useful to quickly identify an interface status:

- **show ip interface brief** - Displays a summary for all interfaces including the IPv4 address of the interface and current operational status.
- **show ip route** - Displays the contents of the IPv4 routing table stored in RAM. In Cisco IOS 15, active interfaces should appear in the routing table with two related entries identified by the code '**C**' (Connected) or '**L**' (Local). In previous IOS versions, only a single entry with the code '**C**' will appear.
- **show running-config interface** *interface-id* - Displays the commands configured on the specified interface.

The following two commands are used to gather more detailed interface information:

- **show interfaces** - Displays interface information and packet flow count for all interfaces on the device.
- **show ip interface** - Displays the IPv4 related information for all interfaces on a router.

1.1.4.2 Verify IPv6 Interface Settings

The commands to verify the IPv6 interface configuration are similar to the commands used for IPv4.

The **show ipv6 interface brief** command displays a summary for each of the interfaces. The "up/up" output on the same line as the interface name indicates the Layer 1/Layer 2 interface state. This is the same as the Status and Protocol columns in the equivalent IPv4 command.

The output displays two configured IPv6 addresses per interface. One address is the IPv6 global unicast address that was manually entered. The other address, which begins with FE80, is the link-local unicast address for the interface. A link-local address is automatically added to an interface whenever a global unicast address is assigned. An IPv6 network interface is required to have a link-local address, but not necessarily a global unicast address.

The **show ipv6 interface gigabitethernet 0/0 c**ommand output shown displays the interface status and all of the IPv6 addresses belonging to the interface. Along with the link local address and global unicast address, the output includes the multicast addresses assigned to the interface, beginning with prefix FF02.

The **show ipv6 route** command can be used to verify that IPv6 networks and specific IPv6 interface addresses have been installed in the IPv6 routing table. The **showipv6route** command will only display IPv6 networks, not IPv4 networks.

Within the routing table, a '**C**' next to a route indicates that this is a directly connected network. When the router interface is configured with a global unicast address and is in the "up/up" state, the IPv6 prefix and prefix length is added to the IPv6 routing table as a connected route.

The IPv6 global unicast address configured on the interface is also installed in the routing table as a local route. The local route has a /128 prefix. Local routes are used by the routing table to efficiently process packets with the interface address of the router as the destination.

1.1.4.3 Filter Show Command Output

Commands that generate multiple screens of output are, by default, paused after 24 lines. At the end of the paused output, the --More-- text displays. Pressing **Enter** displays the next line and pressing the spacebar displays the next set of lines. Use the **terminal length** command to specify the number of lines to be displayed. A value of 0 (zero) prevents the router from pausing between screens of output.

Another very useful feature that improves the user experience in the command-line interface (CLI) is the filtering of **show** output. Filtering commands can be used to display specific sections of output. To enable

the filtering command, enter a pipe (|) character after the **show** command and then enter a filtering parameter and a filtering expression.

The filtering parameters that can be configured after the pipe include:

- **section** - Shows entire section that starts with the filtering expression
- **include** - Includes all output lines that match the filtering expression
- **exclude** - Excludes all output lines that match the filtering expression
- **begin** - Shows all the output lines from a certain point, starting with the line that matches the filtering expression

Note: Output filters can be used in combination with any **show** command.

1.1.4.4 Command History Feature

The command history feature is useful, because it temporarily stores the list of executed commands to be recalled.

To recall commands in the history buffer, press **Ctrl**+**P** or the **Up Arrow** key. The command output begins with the most recent command. Repeat the key sequence to recall successively older commands. To return to more recent commands in the history buffer, press **Ctrl**+**N** or the **Down Arrow** key. Repeat the key sequence to recall successively more recent commands.

By default, command history is enabled and the system captures the last 10 command lines in its history buffer. Use the **show history** privileged EXEC command to display the contents of the buffer.

It is also practical to increase the number of command lines that the history buffer records during the current terminal session only. Use the **terminal history size** user EXEC command to increase or decrease the size of the buffer.

1.2 Routing Decisions

1.2.1 Switching Packets between Networks

1.2.1.1 Router Switching Function

A primary function of a router is to forward packets toward their destination. This is accomplished by using a switching function, which is the process used by a router to accept a packet on one interface and forward it out of another interface. A key responsibility of the switching function is to encapsulate packets in the appropriate data link frame type for the outgoing data link.

Note: In this context, the term "switching" literally means moving packets from source to destination and should not be confused with the function of a Layer 2 switch.

After the router has determined the exit interface using the path determination function, the router must encapsulate the packet into the data link frame of the outgoing interface.

What does a router do with a packet received from one network and destined for another network? The router performs the following three major steps:
Step 1. De-encapsulates the Layer 2 frame header and trailer to expose the Layer 3 packet.
Step 2. Examines the destination IP address of the IP packet to find the best path in the routing table.
Step 3. If the router finds a path to the destination, it encapsulates the Layer 3 packet into a new Layer 2 frame and forwards the frame out the exit interface.

It is common for packets to require encapsulation into a different type of Layer 2 frame than the one which was received. For example, a router might receive an Ethernet encapsulated frame on a FastEthernet interface, and then process that frame to be forwarded out of a serial interface.

1.2.2 Path Determination

1.2.2.1 Routing Decisions

A primary function of a router is to determine the best path to use to send packets. To determine the best path, the router searches its routing table for a network address that matches the destination IP address of the packet.

The routing table search results in one of three path determinations:

- **Directly connected network** - If the destination IP address of the packet belongs to a device on a network that is directly connected to one of the interfaces of the router, that packet is forwarded directly to the destination device. This means that the destination IP address of the packet is a host address on the same network as the interface of the router.
- **Remote network** - If the destination IP address of the packet belongs to a remote network, then the packet is forwarded to another router. Remote networks can only be reached by forwarding packets to another router.
- **No route determined** - If the destination IP address of the packet does not belong to either a connected or remote network, the router determines if there is a Gateway of Last Resort available. A Gateway of Last Resort is set when a default route is configured or learned on a router. If there is a default route, the packet is forwarded to the Gateway of Last Resort. If the router does not have a default route, then the packet is discarded.

1.2.2.2 Best Path

Determining the best path involves the evaluation of multiple paths to the same destination network and selecting the optimum or shortest path to reach that network. Whenever multiple paths to the same network exist, each path uses a different exit interface on the router to reach that network.

The 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 quantitative value used to measure the distance to a given network. The best path to a network is the path with the lowest metric.

Dynamic routing protocols typically use their own rules and metrics to build and update routing tables. The routing algorithm generates a value, or a metric, for each path through the network. Metrics can be based on either a single characteristic or several characteristics of a path. Some routing protocols can base route selection on multiple metrics, combining them into a single metric.

The following lists some dynamic protocols and the metrics they use:

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

1.2.2.3 Load Balancing

What happens if a routing table has two or more paths with identical metrics to the same destination network?

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. This is called equal cost load balancing. The routing table contains the single destination network, but has multiple exit interfaces, one for each equal cost path. The router forwards packets using the multiple exit interfaces listed in the routing table.

If configured correctly, load balancing can increase the effectiveness and performance of the network. Equal cost load balancing can be configured to use both dynamic routing protocols and static routes.

Note: Only EIGRP supports unequal cost load balancing.

1.2.2.4 Administrative Distance

It is possible for a router to be configured with multiple routing protocols and static routes. If this occurs, the routing table may have more than one route source for the same destination network. For example, if both RIP and EIGRP are configured on a router, both routing protocols may learn of the same destination network. However, each routing protocol may decide on a different path to reach the destination based on that routing protocol's metrics. RIP chooses a path based on hop count, whereas EIGRP chooses a path based on its composite metric. How does the router know which route to use?

Cisco IOS uses what is known as the administrative distance (AD) to determine the route to install into the IP routing table. The AD represents the "trustworthiness" of the route; the lower the AD, the more trustworthy the route source. For example, a static route has an AD of 1, whereas an EIGRP-discovered route has an AD of 90. Given two separate routes to the same destination, the router chooses the route with the lowest AD. When a router has the choice of a static route and an EIGRP route, the static route takes precedence. Similarly, a directly connected route with an AD of 0 takes precedence over a static route with an AD of 1.

The figure lists various routing protocols and their associated ADs.

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

1.3 Router Operation

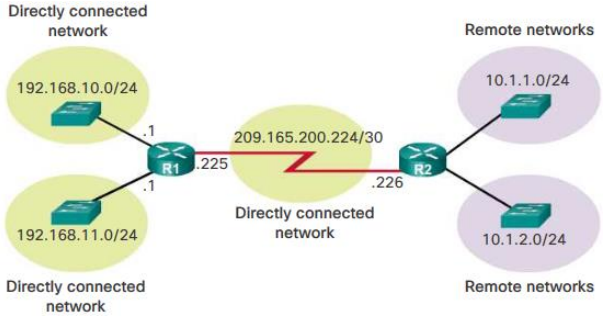
1.3.1 Analyze the Routing Table

1.3.1.1 The Routing Table

The routing table of a router stores information about:

- **Directly connected routes** - These routes come from the active router interfaces. Routers add a directly connected route when an interface is configured with an IP address and is activated.
- **Remote routes** - These are remote networks connected to other routers. Routes to these networks can either be statically configured or dynamically learned through dynamic routing protocols.

Specifically, a routing table is a data file in RAM that is used to store route information about directly connected and remote networks. The routing table contains network or next hop associations. These associations tell a router that a particular destination can be optimally reached by sending the packet to a specific router that represents the next hop on the way to the final destination. The next hop association can also be the outgoing or exit interface to the next destination.



The figure above identifies the directly connected networks and remote networks of router R1.

1.3.1.2 Routing Table Sources

On a Cisco router, the **show ip route** command can be used to display the IPv4 routing table of a router. A router provides additional route information, including how the route was learned, how long the route has been in the table, and which specific interface to use to get to a predefined destination.

Entries in the routing table can be added as:

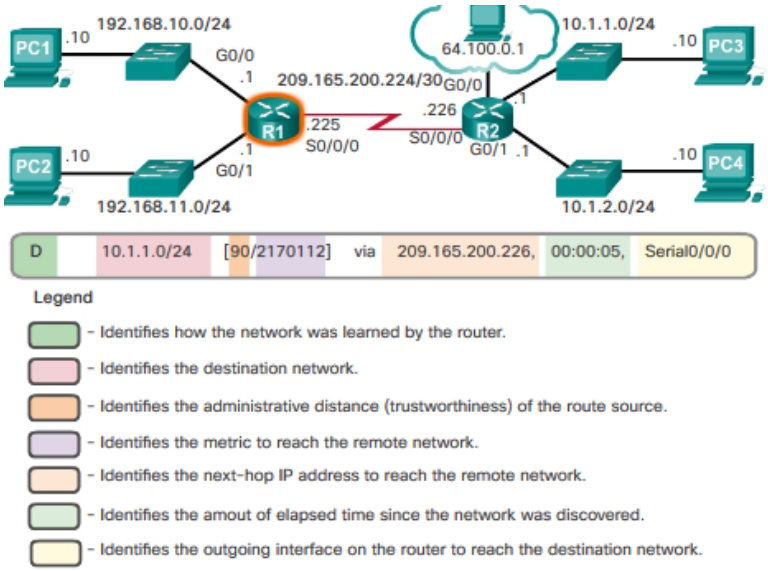
- **Local Route interfaces** - Added when an interface is configured and active. This entry is only displayed in IOS 15 or newer for IPv4 routes and all IOS releases for IPv6 routes.
- **Directly connected interfaces** - Added to the routing table when an interface is configured and active.
- **Static routes** - Added when a route is manually configured and the exit interface is active.
- **Dynamic routing protocol** - Added when routing protocols that dynamically learn about the network, such as EIGRP or OSPF, are implemented and networks are identified.

The sources of the routing table entries are identified by a code. The code identifies how the route was learned. For instance, common codes include:

- **L** - Identifies the address assigned to a router's interface. This allows the router to efficiently determine when it receives a packet for the interface instead of being forwarded.
- **C** - Identifies a directly connected network.
- **S** - Identifies a static route created to reach a specific network.
- **D** - Identifies a dynamically learned network from another router using EIGRP.
- **O** - Identifies a dynamically learned network from another router using the OSPF routing protocol.

1.3.1.3 Remote Network Routing Entries

As a network administrator, it is imperative to know how to interpret the content of IPv4 and IPv6 routing tables. The figure displays an IPv4 routing table entry on R1 for the route to remote network 10.1.1.0.



The entry identifies the following information:

- **Route source** - Identifies how the route was learned.
- **Destination network** - Identifies the address of the remote network.
- **Administrative distance** - Identifies the trustworthiness of the route source. Lower values indicate preferred route source.
- **Metric** - Identifies the value assigned to reach the remote network. Lower values indicate preferred routes.
- **Next-hop** - Identifies the IPv4 address of the next router to forward the packet to.
- **Route timestamp** - Identifies how much time has passed since the route was learned.

- **Outgoing interface** - Identifies the exit interface to use to forward a packet toward the final destination.

1.3.2 Directly Connected Routes

1.3.2.1 Directly Connected Interfaces

A newly deployed router, without any configured interfaces, has an empty routing table.

Before the interface state is considered up/up and added to the IPv4 routing table, the interface must:

- Be assigned a valid IPv4 or IPv6 address
- Be activated with the **no shutdown** command
- Receive a carrier signal from another device (router, switch, host, etc.)

When the interface is up, the network of that interface is added to the routing table as a directly connected network.

1.3.2.2 Directly Connected Routing Table Entries

An active, properly configured, directly connected interface actually creates two routing table entries.

The routing table entry for directly connected interfaces is simpler than the entries for remote networks. The entries contain the following information:

- **Route source** - Identifies how the route was learned. Directly connected interfaces have two route source codes. '**C**' identifies a directly connected network. '**L**' identifies the IPv4 address assigned to the router's interface.
- **Destination network** - The address of the remote network.
- **Outgoing interface** - Identifies the exit interface to use when forwarding packets to the destination network.

Note: Prior to IOS 15, local route routing table entries (**L**) were not displayed in the IPv4 routing table. Local route (**L**) entries have always been a part of the IPv6 routing table.

1.3.3. Statically Learned Routes

1.3.3.1 Static Routes

After directly connected interfaces are configured and added to the routing table, then static or dynamic routing can be implemented.

Static routes are manually configured. They define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include improved security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols, and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

There are two common types of static routes in the routing table:

- Static route to a specific network
- Default static route

A static route can be configured to reach a specific remote network. IPv4 static routes are configured using the following command:

```
Router(config)# ip route network mask { next-hop-ip | exit-intf }
```

A static route is identified in the routing table with the code '**S**'.

A default static route is similar to a default gateway on a host. The default static route specifies the exit point to use when the routing table does not contain a path for the destination network. A default static

route is useful when a router has only one exit point to another router, such as when the router connects to a central router or service provider.

To configure an IPv4 default static route, use the following command:

```
Router(config)# ip route 0.0.0.0 0.0.0.0 { exit-intf | next-hop-ip }
```

1.3.4 Dynamic Routing Protocols

1.3.4.1 Dynamic Routing

Dynamic routing protocols are used by routers to share information about the reachability and status of remote networks. Dynamic routing protocols perform several activities, including network discovery and maintaining routing tables.

Network discovery is the ability of a routing protocol to share information about the networks that it knows about with other routers that are also using the same routing protocol. Instead of depending on manually configured static routes to remote networks on every router, a dynamic routing protocol allows the routers to automatically learn about these networks from other routers. These networks, and the best path to each, are added to the routing table of the router, and identified as a network learned by a specific dynamic routing protocol.

During network discovery, routers exchange routes and update their routing tables. Routers have converged after they have finished exchanging and updating their routing tables. Routers then maintain the networks in their routing tables.

1.3.4.2 IPv4 Routing Protocols

A router running a dynamic routing protocol does not only make a best path determination to a network, it also determines a new best path if the initial path becomes unusable (or if the topology changes). For these reasons, dynamic routing protocols have an advantage over static routes. Routers that use dynamic routing protocols automatically share routing information with other routers and compensate for any topology changes without involving the network administrator.

Cisco 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

To determine which routing protocols are supported by the IOS, use the **router ?** command in global configuration mode.

1.3.4.3 IPv4 Dynamic Routing Examples

In this dynamic routing example, assume that R1 and R2 have been configured to support the dynamic routing protocol EIGRP. The routers also advertise directly connected networks. R2 advertises that it is the default gateway to other networks.

The output displays the routing table of R1 after the routers have exchanged updates and converged. Along with the connected and link local interfaces, there are three '**D**' entries in the routing table.

- The entry beginning with '**D*EX**' identifies that the source of this entry was EIGRP ('**D**'). The route is a candidate to be a default route ('*****'), and the route is an external route ('***EX**') forwarded by EIGRP.
- The other two '**D**' entries are routes installed in the routing table based on the update from R2 advertising its LANs.

1.3.4.4 IPv6 Routing Protocols

ISR devices support dynamic IPv6 routing protocols including:

- RIPng (RIP next generation)
- OSPFv3
- EIGRP for IPv6

Support for dynamic IPv6 routing protocols is dependent on hardware and IOS version. Most of the modifications in the routing protocols are to support the longer IPv6 addresses and different header structures.

To enable IPv6 routers to forward traffic, you must configure the **ipv6 unicast-routing** global configuration command.

1.3.4.5 IPv6 Dynamic Routing Examples

Routers R1 and R2 have been configured with the dynamic routing protocol EIGRP for IPv6. (This is the IPv6 equivalent of EIGRP for IPv4.)

To view the routing table on R1, enter the **show ipv6 route** command. The output displays the routing table of R1 after the routers have exchanged updates and converged. Along with the connected and local routes, there are two '**D**' entries (EIGRP routes) in the routing table.

CHAPTER 2: STATIC ROUTING

Routing is at the core of every data network, moving information across an internetwork from source to destination. Routers are the devices responsible for the transfer of packets from one network to the next.

Routers learn about remote networks either dynamically, using routing protocols, or manually, or using static routes. In many cases, routers use a combination of both dynamic routing protocols and static routes. This chapter focuses on static routing.

Static routes are very common and do not require the same amount of processing and overhead as dynamic routing protocols.

2.1 Implement Static Routes

2.1.1 Static Routing

2.1.1.1 Reach Remote Networks

A router can learn about remote networks in one of two ways:

- **Manually** - Remote networks are manually entered into the route table using static routes.
- **Dynamically** - Remote routes are automatically learned using a dynamic routing protocol.

A network administrator can manually configure a static route to reach a specific network. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured any time the network topology changes.

2.1.1.2 Why Use Static Routing?

Static routing provides some advantages over dynamic routing, including:

- Static routes are not advertised over the network, resulting in better security.
- Static routes use less bandwidth than dynamic routing protocols, no CPU cycles are used to calculate and communicate routes.
- The path a static route uses to send data is known.

Static routing has the following disadvantages:

- Initial configuration and maintenance is time-consuming.
- Configuration is error-prone, especially in large networks.
- Administrator intervention is required to maintain changing route information.
- Does not scale well with growing networks; maintenance becomes cumbersome.
- Requires complete knowledge of the whole network for proper implementation.

In the figure, dynamic and static routing features are compared. Notice that the advantages of one method are the disadvantages of the other.

	Dynamic Routing	Static Routing
Configuration Complexity	Generally independent of the network size	Increases with network size
Topology Changes	Automatically adapts to topology changes	Administrator intervention required
Scaling	Suitable for simple and complex topologies	Suitable for simple topologies
Security	Less secure	More secure
Resource Usage	Uses CPU, memory, link bandwith	No extra resources needed
Predictability	Route depends on the current topology	Route to destination is always the same

Static routes are useful for smaller networks with only one path to an outside network. They also provide security in a larger network for certain types of traffic or links to other networks that need more control. It is important to understand that static and dynamic routing are not mutually exclusive. Rather, most networks use a combination of dynamic routing protocols and static routes. This may result in the router having multiple paths to a destination network via static routes and dynamically learned routes. However,

recall that the administrative distance (AD) value is a measure of the preference of route sources. Route sources with low AD values are preferred over routes sources with higher AD values. The AD value for a static route is 1. Therefore, a static route will take precedence over all dynamically learned routes, which will have higher AD values.

2.1.1.3 When to Use Static Routes

Static routing has three primary uses:

- Providing ease of routing table maintenance in smaller networks that are not expected to grow significantly.
- Routing to and from stub networks. A stub network is a network accessed by a single route, and the router has only one neighbor.
- Using a single default route to represent a path to any network that does not have a more specific match with another route in the routing table. Default routes are used to send traffic to any destination beyond the next upstream router.

2.1.2 Types of Static Routes

2.1.2.1 Static Route Applications

Static routes are most often used to connect to a specific network or to provide a Gateway of Last Resort for a stub network. They can also be used to:

- Reduce the number of routes advertised by summarizing several contiguous networks as one static route
- Create a backup route in case a primary route link fails

The following types of IPv4 and IPv6 static routes will be discussed:

- Standard static route
- Default static route
- Summary static route
- Floating static route

2.1.2.2 Standard Static Route

Both IPv4 and IPv6 support the configuration of static routes. Static routes are useful when connecting to a specific remote network.

2.1.2.3 Default Static Route

A default route is a route that matches all packets and is used by the router if a packet does not match any other, more specific route in the routing table. A default route can be dynamically learned or statically configured. A default static route is simply a static route with 0.0.0.0/0 as the destination IPv4 address. Configuring a default static route creates a Gateway of Last Resort.

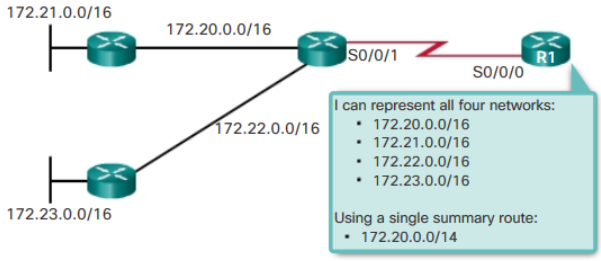
Default static routes are used:

- When no other routes in the routing table match the packet destination IP address. In other words, when a more specific match does not exist. A common use is when connecting a company's edge router to the ISP network.
- When a router has only one other router to which it is connected. In this situation, the router is known as a stub router.

2.1.2.4 Summary Static Route

To reduce the number of routing table entries, multiple static routes can be summarized into a single static route if:

- The destination networks are contiguous and can be summarized into a single network address.
- The multiple static routes all use the same exit interface or next-hop IP address.



In the figure, R1 would require four separate static routes to reach the 172.20.0.0/16 to 172.23.0.0/16 networks. Instead, one summary static route can be configured and still provide connectivity to those networks.

2.1.2.5 Floating Static Route

Another type of static route is a floating static route. Floating static routes are static routes that are used to provide a backup path to a primary static or dynamic route, in the event of a link failure. The floating static route is only used when the primary route is not available.

To accomplish this, the floating static route is configured with a higher administrative distance than the primary route. The administrative distance represents the trustworthiness of a route. If multiple paths to the destination exist, the router will choose the path with the lowest administrative distance.

For example, assume that an administrator wants to create a floating static route as a backup to an EIGRP-learned route. The floating static route must be configured with a higher administrative distance than EIGRP. EIGRP has an administrative distance of 90. If the floating static route is configured with an administrative distance of 95, the dynamic route learned through EIGRP is preferred to the floating static route. If the EIGRP-learned route is lost, the floating static route is used in its place.

2.2 Configure Static and Default Routes

2.2.1 Configure IPv4 Static Routes

2.2.1.1 ip route Command

Static routes are configured using the **ip route** global configuration command.

The following parameters are required to configure static routing:

- *network-address* - Destination network address of the remote network to be added to the routing table, often this is referred to as the prefix.
- *subnet-mask* - Subnet mask, or just mask, of the remote network to be added to the routing table. The subnet mask can be modified to summarize a group of networks.

One or both of the following parameters must also be used:

- *ip-address* - The IP address of the connecting router to use to forward the packet to the remote destination network. Commonly referred to as the next hop.
- *exit-intf* - The outgoing interface to use to forward the packet to the next hop.

The *distance* parameter is used to create a floating static route by setting an administrative distance that is higher than a dynamically learned route.

2.2.1.2 Next-Hop Options

The next hop can be identified by an IP address, exit interface, or both. How the destination is specified creates one of the three following route types:

- **Next-hop route** - Only the next-hop IP address is specified
- **Directly connected static route** - Only the router exit interface is specified
- **Fully specified static route** - The next-hop IP address and exit interface are specified

2.2.1.3 Configure a Next-Hop Static Route

In a next-hop static route, only the next-hop IP address is specified. The exit interface is derived from the next hop.

Before any packet is forwarded by a router, the routing table process must determine the exit interface to use to forward the packet. This is known as route resolvability.

When the router performs multiple lookups in the routing table before forwarding a packet, it is performing a process known as a recursive lookup. Because recursive lookups consume router resources, they should be avoided when possible.

A recursive static route is valid (that is, it is a candidate for insertion in the routing table) only when the specified next hop resolves, either directly or indirectly, to a valid exit interface. If the exit interface is "down" or "administratively down", then the static route will not be installed in the routing table.

2.2.1.4 Configure a Directly Connected Static Route

When configuring a static route, another option is to use the exit interface to specify the next-hop address.

Notice how the routing table looks different for the route configured with an exit interface than for the route configured with a recursive entry.

Configuring a directly connected static route with an exit interface allows the routing table to resolve the exit interface in a single search, instead of two searches. Although the routing table entry indicates "directly connected", the administrative distance of the static route is still 1. Only a directly connected interface can have an administrative distance of 0.

Note: For point-to-point interfaces, you can use static routes that point to the exit interface or to the next-hop address. For multipoint/broadcast interfaces, it is more suitable to use static routes that point to a next-hop address.

Note: CEF (Cisco Express Forwarding) is the default behavior on most platforms running IOS 12.0 or later. CEF provides optimized lookup for efficient packet forwarding by using two main data structures stored in the data plane: a FIB (Forwarding Information Base), which is a copy of the routing table, and an adjacency table that includes Layer 2 addressing information. The information combined in both of these tables work together so there is no recursive lookup needed for next-hop IP address lookups. In other words, a static route using a next-hop IP requires only a single lookup when CEF is enabled on the router. Although static routes that use only an exit interface on point-to-point networks are common, the use of the default CEF forwarding mechanism makes this practice unnecessary. CEF is discussed in more detail later in the course.

2.2.1.5 Configure a Fully Specified Static Route

In a fully specified static route, both the exit interface and the next-hop IP address are specified. This is another type of static route that is used in older IOSs, prior to CEF. This form of static route is used when the exit interface is a multi-access interface and it is necessary to explicitly identify the next hop. The next hop must be directly connected to the specified exit interface.

Suppose that the network link between R1 and R2 is an Ethernet link and that the GigabitEthernet 0/1 interface of R1 is connected to that network. CEF is not enabled. To eliminate the recursive lookup, a directly connected static route can be implemented using the following command:

```
R1(config)# ip route 192.168.2.0 255.255.255.0 GigabitEthernet 0/1
```

However, this may cause unexpected or inconsistent results. The difference between an Ethernet multi-access network and a point-to-point serial network is that a point-to-point serial network has only one other device on that network, the router at the other end of the link. With Ethernet networks, there may

be many different devices sharing the same multi-access network, including hosts and even multiple routers. By only designating the Ethernet exit interface in the static route, the router will not have sufficient information to determine which device is the next-hop device.

R1 knows that the packet needs to be encapsulated in an Ethernet frame and sent out the GigabitEthernet 0/1 interface. However, R1 does not know the next-hop IPv4 address; therefore, it cannot determine the destination MAC address for the Ethernet frame.

Depending upon the topology and the configurations on other routers, this static route may or may not work. It is recommended that when the exit interface is an Ethernet network, that a fully specified static route is used, including both the exit interface and the next-hop address.

Note: With the use of CEF, a fully specified static route is no longer necessary. A static route using a next-hop address should be used.

2.2.1.6 Verify a Static Route

Along with **ping** and **traceroute**, useful commands to verify static routes include:

- show ip route
- show ip route static
- show ip route *network*

2.2.2 Configure IPv4 Default Routes

2.2.2.1 Default Static Route

Routers commonly use default routes that are either configured locally or learned from another router, using a dynamic routing protocol. A default route does not require any left-most bits to match between the default route and the destination IPv4 address. A default route is used when no other routes in the routing table match the destination IP address of the packet. In other words, if a more specific match does not exist, then the default route is used as the Gateway of Last Resort.

Default static routes are commonly used when connecting:

- An edge router to a service provider network
- A stub router (a router with only one upstream neighbor router)

The command syntax for a default static route is similar to any other static route, except that the network address is **0.0.0.0** and the subnet mask is **0.0.0.0**.

Note: An IPv4 default static route is commonly referred to as a quad-zero route.

2.2.2.2 Configure a Default Static Route

R1 can be configured with three static routes to reach all of the remote networks in the example topology. However, R1 is a stub router because it is only connected to R2. Therefore, it would be more efficient to configure a default static route.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 172.16.2.2
R1(config)#
```

The example in the figure configures a default static route on R1. With the configuration shown in the example, any packets not matching more specific route entries are forwarded to 172.16.2.2.

2.2.2.3 Verify a Default Static Route

The show ip route static command output displays the contents of the static routes in the routing table. Note the asterisk (*)next to the route with code ‘S’. As displayed in the Codes table, the asterisk indicates that this static route is a candidate default route, which is why it is selected as the Gateway of Last Resort.

The key to this configuration is the /0 mask. The subnet mask in a routing table determines how many bits must match between the destination IP address of the packet and the route in the routing table. A

binary 1 indicates that the bits must match. A binary 0 indicates that the bits do not have to match. A /0 mask in this route entry indicates that none of the bits are required to match. The default static route matches all packets for which a more specific match does not exist.

2.2.3 Configure Floating Static Routes

2.2.3.1 Floating Static Routes

Floating static routes are static routes that have an administrative distance greater than the administrative distance of another static route or dynamic routes. They are very useful when providing a backup to a primary link.

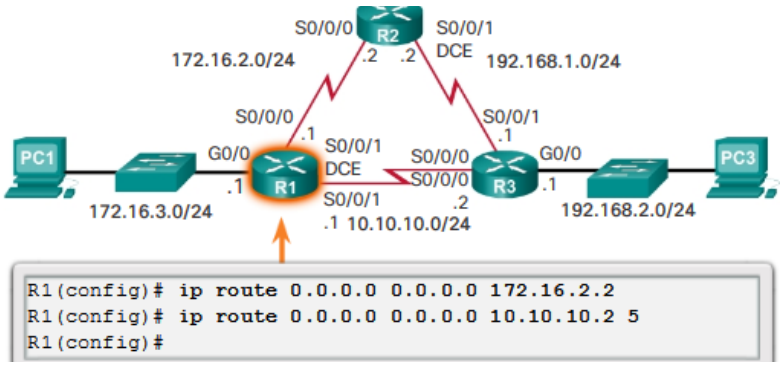
By default, static routes have an administrative distance of 1, making them preferable to routes learned from dynamic routing protocols. For example, the administrative distances of some common dynamic routing protocols are:

- EIGRP = 90
- IGRP = 100
- OSPF = 110
- IS-IS = 115
- RIP = 120

The administrative distance of a static route can be increased to make the route less desirable than that of another static route or a route learned through a dynamic routing protocol. In this way, the static route "floats" and is not used when the route with the better administrative distance is active. However, if the preferred route is lost, the floating static route can take over, and traffic can be sent through this alternate route.

2.2.3.2 Configure an IPv4 Floating Static Route

IPv4 floating static routes are configured using the **ip route** global configuration command and specifying an administrative distance. If no administrative distance is configured, the default value (1) is used.



Refer to the topology in Figure above. In this scenario, the preferred default route from R1 is to R2. The connection to R3 should be used for backup only.

R1 is configured with a default static route pointing to R2. Because no administrative distance is configured, the default value (1) is used for this static route. R1 is also configured with a floating static default pointing to R3 with an administrative distance of 5. This value is greater than the default value of 1 and therefore; this route floats and is not present in the routing table, unless the preferred route fails.

2.2.3.3 Test the IPv4 Floating Static Route

Because the default static route on R1 to R2 has an administrative distance of 1, traffic from R1 to R3 should go through R2. The output in Figure confirms that traffic between R1 and R3 flows through R2.

```
R1# traceroute 192.168.2.1
Type escape sequence to abort.
Tracing the route to 192.168.2.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.2.2 4 msec 4 msec 8 msec
  2 192.168.1.1 12 msec * 12 msec
```

What would happen if R2 failed? To simulate this failure both serial interfaces of R2 are shut down.

R1 automatically generates messages indicating that the serial interface to R2 is down. A look at the routing table verifies that the default route is now pointing to R3 using the floating static default route configured with an AD value of 5 and a next-hop of 10.10.10.2.

```
R1# traceroute 192.168.2.1
Type escape sequence to abort.
Tracing the route to 192.168.2.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.10.2 4 msec 4 msec *
```

The output in Figure above confirms that traffic now flows directly between R1 and R3.

2.2.4 Configure Static Host Routes

2.2.4.1 Automatically Installed Host Routes

A host route is an IPv4 address with a 32-bit mask or an IPv6 address with a 128-bit mask. There are three ways a host route can be added to the routing table:

- Automatically installed when an IP address is configured on the router
- Configured as a static host route
- Host route automatically obtained through other methods (discussed in later courses)

Cisco IOS automatically installs a host route, also known as a local host route, when an interface address is configured on the router. A host route allows for a more efficient process for packets that are directed to the router itself, rather than for packet forwarding. This is in addition to the connected route, designated with a C in the routing table for the network address of the interface.

When an active interface on a router is configured with an IP address, a local host route is automatically added to the routing table. The local routes are marked with “L” in the output of the routing table. The IP addresses assigned to the Branch Serial0/0/0 interface are 198.51.100.1/30 for IPv4 and 2001:DB8:ACAD:1::1/64 for IPv6. The local routes for the interface are installed by the IOS in the routing table

Note: For IPv4, the local routes marked with “L” were introduced with IOS version 15.

2.2.4.2 Configure IPv4 and IPv6 Static Host Routes

A host route can be a manually configured static route to direct traffic to a specific destination device, such as an authentication server. The static route uses a destination IP address and a 255.255.255.255 (/32) mask for IPv4 host routes and a /128 prefix length for IPv6 host routes. Static routes are marked with “S” in the output of the routing table.

For IPv6 static routes, the next-hop address can be the link-local address of the adjacent router. However, you must specify an interface type and an interface number when using a link-local address as the next hop.

CHAPTER 3: DYNAMIC ROUTING

The data networks that we use in our everyday lives to learn, play, and work range from small, local networks to large, global internetworks. At home, a user may have a router and two or more computers. At work, an organization may have multiple routers and switches servicing the data communication needs of hundreds or even thousands of PCs.

Routers forward packets by using information in the routing table. Routes to remote networks can be learned by the router in two ways: static routes and dynamic routes.

In a large network with numerous networks and subnets, configuring and maintaining static routes between these networks requires a great deal of administrative and operational overhead. This operational overhead is especially cumbersome when changes to the network occur, such as a down link or implementing a new subnet. Implementing dynamic routing protocols can ease the burden of configuration and maintenance tasks and give the network scalability.

3.1 Dynamic Routing Protocols

3.1.1 Dynamic Routing Protocol Overview

3.1.1.1 Dynamic Routing Protocol Evolution

Dynamic routing protocols have been used in networks since the late 1980s. One of the first routing protocols was RIP. RIPv1 was released in 1988, but some of the basic algorithms within the protocol were used on the Advanced Research Projects Agency Network (ARPANET) as early as 1969.

As networks evolved and became more complex, new routing protocols emerged. The RIP protocol was updated to RIPv2 to accommodate growth in the network environment. However, RIPv2 still does not scale to the larger network implementations of today. To address the needs of larger networks, two advanced routing protocols were developed: Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS). Cisco developed the Interior Gateway Routing Protocol (IGRP) and Enhanced IGRP (EIGRP), which also scales well in larger network implementations.

Additionally, there was the need to connect different internetworks and provide routing between them. The Border Gateway Protocol (BGP) is now used between Internet service providers (ISPs). BGP is also used between ISPs and their larger private clients to exchange routing information.

With the advent of numerous consumer devices using IP, the IPv4 addressing space is nearly exhausted; thus, IPv6 has emerged. To support the communication based on IPv6, newer versions of the IP routing protocols have been developed.

3.1.1.2 Dynamic Routing Protocol Components

Routing protocols are used to facilitate the exchange of routing information between routers. A routing protocol is a set of processes, algorithms, and messages that are used to exchange routing information and populate the routing table with the routing protocol's choice of best paths. The purpose of dynamic routing protocols includes:

- Discovery of remote networks
- Maintaining up-to-date routing information
- Choosing the best path to destination networks
- Ability to find a new best path if the current path is no longer available

The main components of dynamic routing protocols include:

- **Data structures -** Routing protocols typically use tables or databases for its operations. This information is kept in RAM.

- **Routing protocol messages -** Routing protocols use various types of messages to discover neighboring routers, exchange routing information, and other tasks to learn and maintain accurate information about the network.
- **Algorithm -** An algorithm is a finite list of steps used to accomplish a task. Routing protocols use algorithms for facilitating routing information and for best path determination.

Routing protocols allow routers to dynamically share information about remote networks and automatically offer this information to their own routing tables. Click Play in the figure to see an animation of this process.

Routing protocols determine the best path, or route, to each network. That route is then offered to the routing table. The route will be installed in the routing table is there is not another routing source with a lower administrative distance. For example, a static route with an administrative distance of 1 will have precedence over the same network learned by a dynamic routing protocol. A primary benefit of dynamic routing protocols is that routers exchange routing information when there is a topology change. This exchange allows routers to automatically learn about new networks and also to find alternate paths when there is a link failure to a current network.

3.1.2 Dynamic versus Static Routing

3.1.2.1 Static Routing Uses

Before identifying the benefits of dynamic routing protocols, consider the reasons why network professionals use static routing. Dynamic routing certainly has several advantages over static routing; however, static routing is still used in networks today. In fact, networks typically use a combination of both static and dynamic routing.

Static routing has several primary uses, including:

- Providing ease of routing table maintenance in smaller networks that are not expected to grow significantly.
- Routing to and from a stub network, which is a network with only one default route out and no knowledge of any remote networks.
- Accessing a single default route (which is used to represent a path to any network that does not have a more specific match with another route in the routing table).

3.1.2.2 Static Routing Advantages and Disadvantages

The table in the figure highlights the advantages and disadvantages of static routing. Static routing is easy to implement in a small network. Static routes stay the same, which makes them fairly easy to troubleshoot. Static routes do not send update messages; therefore, they require very little overhead.

Advantages	Disadvantages
Easy to implement in a small network.	Suitable only for simple topologies or for special purposes such as a default static route.
Very secure. No advertisements are sent as compared to dynamic routing protocols.	Configuration complexity increases dramatically as network grows.
Route to destination is always the same.	Manual intervention required to re-route traffic.
No routing algorithm or update mechanism required; therefore, extra resources (CPU or RAM) are not required.	

The disadvantages of static routing include:

- They are not easy to implement in a large network.
- Managing the static configurations can become time consuming.
- If a link fails, a static route cannot reroute traffic.

3.1.2.3 Dynamic Routing Protocols Uses

Dynamic routing protocols help the network administrator manage the time-consuming and exacting process of configuring and maintaining static routes.

Imagine maintaining the static routing configurations for the seven routers. What if the company grew and now had four regions and 28 routers to manage? What happens when a link goes down? How do you ensure that redundant paths are available? Dynamic routing is the best choice for large networks.

3.1.2.4 Dynamic Routing Advantages and Disadvantages

The table in the figure highlights the advantages and disadvantages of dynamic routing. Dynamic routing protocols work well in any type of network consisting of several routers. They are scalable and automatically determine better routes if there is a change in the topology. Although there is more to the configuration of dynamic routing protocols, they are simpler to configure than static routing in a large network.

There are disadvantages to dynamic routing. Dynamic routing requires knowledge of additional commands. It is also less secure than static routing because the interfaces identified by the routing protocol send routing updates out. Routes taken may differ between packets. The routing algorithm uses additional CPU, RAM, and link bandwidth.

Notice how dynamic routing addresses the disadvantages of static routing.

3.1.3 Types of Routing Protocols

3.1.3.1 Classifying Routing Protocols

Dynamic routing protocols are used to facilitate the exchange of routing information between routers. A routing protocol is a set of processes, algorithms, and messages that are used to exchange routing information and populate the routing table with the routing protocol's choice of best paths. The purpose of dynamic routing protocols includes:

- Discovery of remote networks
- Maintaining up-to-date routing information
- Choosing the best path to destination networks
- Ability to find a new best path if the current path is no longer available

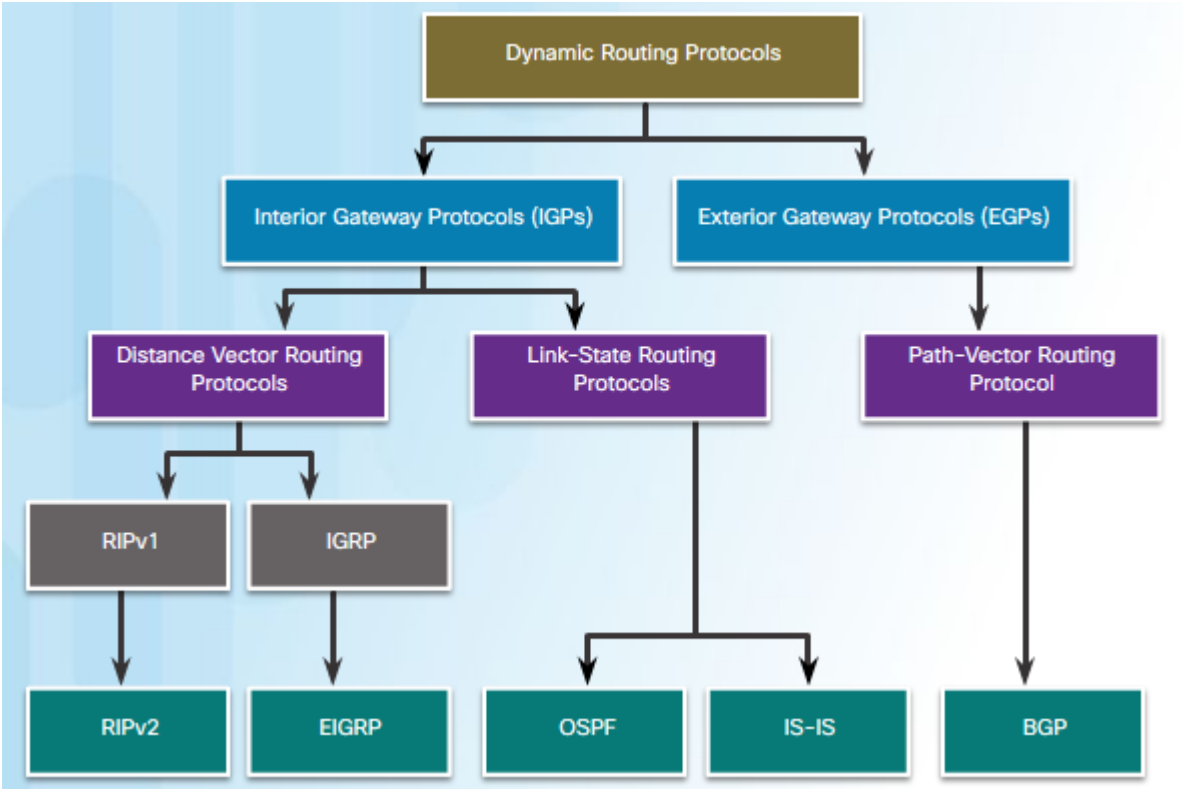
Routing protocols can be classified into different groups according to their characteristics. Specifically, routing protocols can be classified by their:

- **Purpose** - Interior Gateway Protocol (IGP) or Exterior Gateway Protocol (EGP)
- **Operation** - Distance vector protocol, link-state protocol, or path-vector protocol
- **Behavior** - Classful (legacy) or classless protocol

For example, IPv4 routing protocols are classified as follows:

- **RIPv1 (legacy)** - IGP, distance vector, classful protocol
- **IGRP (legacy)** - IGP, distance vector, classful protocol developed by Cisco (deprecated from 12.2 IOS and later)
- **RIPv2** - IGP, distance vector, classless protocol
- **EIGRP** - IGP, distance vector, classless protocol developed by Cisco
- **OSPF** - IGP, link-state, classless protocol
- **IS-IS** - IGP, link-state, classless protocol
- **BGP** - EGP, path-vector, classless protocol

The classful routing protocols, RIPv1 and IGRP, are legacy protocols and are only used in older networks. These routing protocols have evolved into the classless routing protocols, RIPv2 and EIGRP, respectively. Link-state routing protocols are classless by nature.



The figure above displays a hierarchical view of dynamic routing protocol classification.

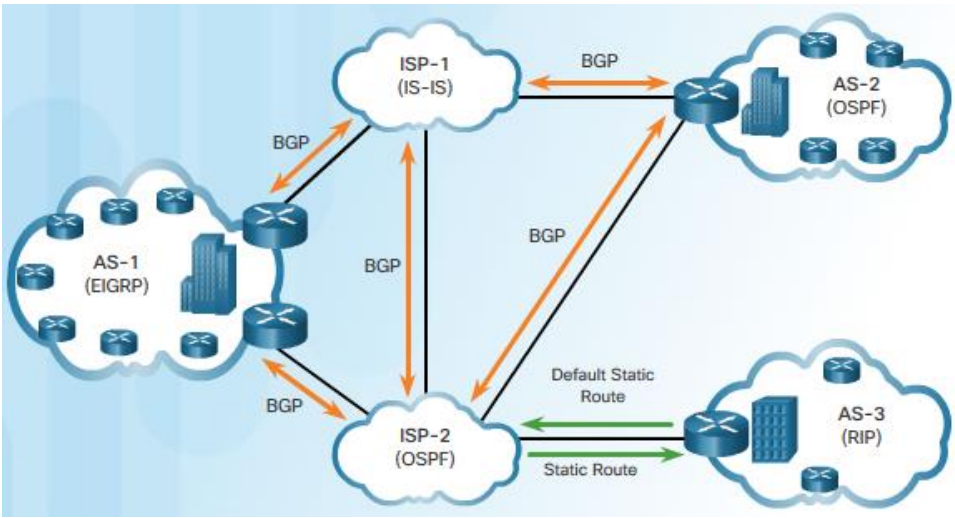
3.1.3.2 IGP and EGP Routing Protocols

An autonomous system (AS) is a collection of routers under a common administration such as a company or an organization. An AS is also known as a routing domain. Typical examples of an AS are a company's internal network and an ISP's network.

The Internet is based on the AS concept; therefore, two types of routing protocols are required:

- **Interior Gateway Protocols (IGP)** - Used for routing within an AS. It is also referred to as intra-AS routing. Companies, organizations, and even service providers use an IGP on their internal networks. IGPs include RIP, EIGRP, OSPF, and IS-IS.
- **Exterior Gateway Protocols (EGP)** - Used for routing between ASes. It is also referred to as inter-AS routing. Service providers and large companies may interconnect using an EGP. The Border Gateway Protocol (BGP) is the only currently-viable EGP and is the official routing protocol used on the Internet.

Note: Because BGP is the only EGP available, the term EGP is rarely used; instead, most engineers simply refer to BGP.



The example in the figure above provides simple scenarios highlighting the deployment of IGPs, BGP, and static routing:

- **ISP-1** - This is an AS and it uses IS-IS as the IGP. It interconnects with other autonomous systems and service providers using BGP to explicitly control how traffic is routed.
- **ISP-2** - This is an AS and it uses OSPF as the IGP. It interconnects with other autonomous systems and service providers using BGP to explicitly control how traffic is routed.
- **AS-1** - This is a large organization and it uses EIGRP as the IGP. Because it is multihomed (i.e., connects to two different service providers), it uses BGP to explicitly control how traffic enters and leaves the AS.
- **AS-2** - This is a medium-sized organization and it uses OSPF as the IGP. It is also multihomed; therefore, it uses BGP to explicitly control how traffic enters and leaves the AS.
- **AS-3** - This is a small organization with older routers within the AS; it uses RIP as the IGP. BGP is not required because it is single-homed (i.e., connects to one service provider). Instead, static routing is implemented between the AS and the service provider.

Note: BGP is beyond the scope of this course and is not discussed in detail.

3.1.3.3 Distance Vector Routing Protocols

Distance vector means that routes are advertised by providing two characteristics:

- **Distance** - Identifies how far it is to the destination network and is based on a metric such as the hop count, cost, bandwidth, delay, and more.
- **Vector** - Specifies the direction of the next-hop router or exit interface to reach the destination.

A router using a distance vector routing protocol does not have the knowledge of the entire path to a destination network. Distance vector protocols use routers as sign posts along the path to the final destination. The only information a router knows about a remote network is the distance or metric to reach that network and which path or interface to use to get there. Distance vector routing protocols do not have a map of the network topology like other types of routing protocols do.

There are four distance vector IPv4 IGPs:

- **RIPv1** - First generation legacy protocol
- **RIPv2** - Simple distance vector routing protocol
- **IGRP** - First generation Cisco proprietary protocol (obsolete and replaced by EIGRP)
- **EIGRP** - Advanced version of distance vector routing

3.1.3.4 Link-State Routing Protocols

In contrast to distance vector routing protocol operation, a router configured with a link-state routing protocol can create a complete view or topology of the network by gathering information from all of the other routers.

To continue our analogy of sign posts, using a link-state routing protocol is like having a complete map of the network topology. The sign posts along the way from source to destination are not necessary, because all link-state routers are using an identical map of the network. A link-state router uses the link-state information to create a topology map and to select the best path to all destination networks in the topology.

Link-state routing protocols do not use periodic updates. In contrast, RIP-enabled routers send periodic updates of their routing information to their neighbors. After the routers have learned about all the required networks (achieved convergence), a link-state update is only sent when there is a change in the topology.

Link-state protocols work best in situations where:

- The network design is hierarchical, usually occurring in large networks

- Fast adaptation to network changes is crucial
- The administrators are knowledgeable about the implementation and maintenance of a link-state routing protocol

There are two link-state IPv4 IGPs:

- **OSPF** - Popular standards-based routing protocol
- **IS-IS** - Popular in provider networks

3.1.3.5 Classful Routing Protocols

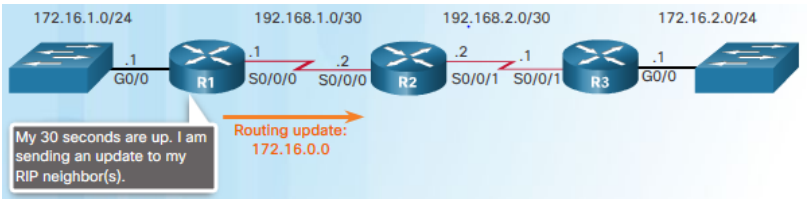
The biggest distinction between classful and classless routing protocols is that classful routing protocols do not send subnet mask information in routing updates. Classless routing protocols include subnet mask information in the routing updates.

The two original IPv4 routing protocols developed were RIPv1 and IGRP. They were created when network addresses were allocated based on classes (i.e., class A, B, or C). At that time, a routing protocol did not need to include the subnet mask in the routing update, because the network mask could be determined based on the first octet of the network address.

Note: Only RIPv1 and IGRP are classful. All other IPv4 and IPv6 routing protocols are classless. Classful addressing has never been a part of IPv6.

The fact that RIPv1 and IGRP do not include subnet mask information in their updates means that they cannot provide variable-length subnet masks (VLSMs) and classless interdomain routing (CIDR).

Classful routing protocols also create problems in discontiguous networks. A discontiguous network is when subnets from the same classful major network address are separated by a different classful network address.



To illustrate the shortcoming of classful routing, refer to the topology in Figure above. Notice that the LANs of R1 (172.16.1.0/24) and R3 (172.16.2.0/24) are both subnets of the same class B network (172.16.0.0/16). They are separated by different classful subnets (192.168.1.0/30 and 192.168.2.0/30) of the same class C networks (192.168.1.0/24 and 192.168.2.0/24).

When R1 forwards an update to R2, RIPv1 does not include the subnet mask information with the update; it only forwards the class B network address 172.16.0.0.

R2 receives and processes the update. It then creates and adds an entry for the class B 172.16.0.0/16 network in the routing table.

R3 forwards an update to R2, it also does not include the subnet mask information and therefore only forwards the classful network address 172.16.0.0.

R2 receives and processes the update and adds another entry for the classful network address 172.16.0.0/16 to its routing table. When there are two entries with identical metrics in the routing table, the router shares the load of the traffic equally among the two links. This is known as load balancing.

This has a negative effect on connectivity to a discontiguous network. Notice the erratic behavior of the **ping** and **traceroute** commands.

3.1.3.6 Classless Routing Protocols

Modern networks no longer use classful IP addressing, and therefore, the subnet mask cannot be determined by the value of the first octet. The classless IPv4 routing protocols (RIPv2, EIGRP, OSPF, and IS-IS) all include the subnet mask information with the network address in routing updates. Classless routing protocols support VLSM and CIDR.

IPv6 routing protocols are classless. The distinction of being classful or classless only applies to IPv4 routing protocols. All IPv6 routing protocols are considered classless because they include the prefix-length with the IPv6 address.

3.1.3.7 Routing Protocol Characteristics

Routing protocols can be compared based on the following characteristics:

- **Speed of Convergence** - Speed of convergence defines how quickly the routers in the network topology share routing information and reach a state of consistent knowledge. The faster the convergence, the more preferable the protocol. Routing loops can occur when inconsistent routing tables are not updated due to slow convergence in a changing network.
- **Scalability** - Scalability defines how large a network can become, based on the routing protocol that is deployed. The larger the network is, the more scalable the routing protocol needs to be.
- **Classful or Classless (Use of VLSM)** - Classful routing protocols do not include the subnet mask and cannot support VLSM. Classless routing protocols include the subnet mask in the updates. Classless routing protocols support VLSM and better route summarization.
- **Resource Usage** - Resource usage includes the requirements of a routing protocol such as memory space (RAM), CPU utilization, and link bandwidth utilization. Higher resource requirements necessitate more powerful hardware to support the routing protocol operation, in addition to the packet forwarding processes.
- **Implementation and Maintenance** - Implementation and maintenance describes the level of knowledge that is required for a network administrator to implement and maintain the network based on the routing protocol deployed.

3.1.3.8 Routing Protocol Metrics

There are cases when a routing protocol learns of more than one route to the same destination. To select the best path, the routing protocol must be able to evaluate and decide between the available paths. This is accomplished through the use of routing metrics.

A metric is a measurable value that is assigned by the routing protocol to different routes based on the usefulness of that route. In situations where there are multiple paths to the same remote network, the routing metrics are used to determine the overall "cost" of a path from source to destination. Routing protocols determine the best path based on the route with the lowest cost.

Different routing protocols use different metrics. The metric used by one routing protocol is not comparable to the metric used by another. As a result, two different routing protocols might choose different paths to the same destination.

The following lists some dynamic protocols and the metrics they use:

- Routing Information Protocol (RIP) - Hop count
- Open Shortest Path First (OSPF) - Cisco's cost based on cumulative bandwidth from source to destination
- Enhanced Interior Gateway Routing Protocol (EIGRP) – Minimum bandwidth, delay, load, reliability, and maximum transmission unit (MTU).

3.1.4 Distance Vector Dynamic Routing

3.1.4.1 Distance Vector Fundamentals

3.1.4.1.1 Dynamic Routing Protocol Operation

All routing protocols are designed to learn about remote networks and to quickly adapt whenever there is a change in the topology. The method that a routing protocol uses to accomplish this depends upon the algorithm it uses and the operational characteristics of that protocol.

In general, the operations of a dynamic routing protocol can be described as follows:

1. The router sends and receives routing messages on its interfaces.

2. The router shares routing messages and routing information with other routers that are using the same routing protocol.

3. Routers exchange routing information to learn about remote networks.

4. When a router detects a topology change, the routing protocol can advertise this change to other routers.

3.1.4.1.2 Cold Start

All routing protocols follow the same patterns of operation. To help illustrate this, consider the following scenario in which all three routers are running RIPv2.

When a router powers up, it knows nothing about the network topology. It does not even know that there are devices on the other end of its links. The only information that a router has is from its own saved configuration file stored in NVRAM. After a router boots successfully, it applies the saved configuration. If the IP addressing is configured correctly, then the router initially discovers its own directly connected networks.

3.1.4.1.3 Network Discovery

After initial boot up and discovery, the routing table is updated with all directly connected networks and the interfaces those networks reside on.

If a routing protocol is configured, the next step is for the router to begin exchanging routing updates to learn about any remote routes.

The router sends an update packet out all interfaces that are enabled on the router. The update contains the information in the routing table, which currently is all directly connected networks.

At the same time, the router also receives and processes similar updates from other connected routers. Upon receiving an update, the router checks it for new network information. Any networks that are not currently listed in the routing table are added.

Refer to the figure above for a topology setup between three routers, R1, R2, and R3 with RIPv2 enabled. Based on this topology, below is a listing of the different updates that R1, R2, and R3 send and receive during initial convergence.

R1:

- Sends an update about network 10.1.0.0 out the Serial0/0/0 interface
- Sends an update about network 10.2.0.0 out the FastEthernet0/0 interface
- Receives an update from R2 about network 10.3.0.0 and increments the hop count by 1
- Stores network 10.3.0.0 in the routing table with a metric of 1

R2:

- Sends an update about network 10.3.0.0 out the Serial 0/0/0 interface

- Sends an update about network 10.2.0.0 out the Serial 0/0/1 interface
- Receives an update from R1 about network 10.1.0.0 and increments the hop count by 1
- Stores network 10.1.0.0 in the routing table with a metric of 1
- Receives an update from R3 about network 10.4.0.0 and increments the hop count by 1
- Stores network 10.4.0.0 in the routing table with a metric of 1

R3:

- Sends an update about network 10.4.0.0 out the Serial 0/0/1 interface
- Sends an update about network 10.3.0.0 out the FastEthernet0/0
- Receives an update from R2 about network 10.2.0.0 and increments the hop count by 1
- Stores network 10.2.0.0 in the routing table with a metric of 1

3.1.4.1.4 Exchanging the Routing Information

At this point the routers have knowledge about their own directly connected networks and about the connected networks of their immediate neighbors. Continuing the journey toward convergence, the routers exchange the next round of periodic updates. Each router again checks the updates for new information.

Refer to the figure above for a topology setup between three routers, R1, R2, and R3. After initial discovery is complete, each router continues the convergence process by sending and receiving the following updates.

R1:

- Sends an update about network 10.1.0.0 out the Serial 0/0/0 interface
- Sends an update about networks 10.2.0.0 and 10.3.0.0 out the FastEthernet0/0 interface
- Receives an update from R2 about network 10.4.0.0 and increments the hop count by 1
- Stores network 10.4.0.0 in the routing table with a metric of 2
- Same update from R2 contains information about network 10.3.0.0 with a metric of 1. There is no change; therefore, the routing information remains the same

R2:

- Sends an update about networks 10.3.0.0 and 10.4.0.0 out of Serial 0/0/0 interface
- Sends an update about networks 10.1.0.0 and 10.2.0.0 out of Serial 0/0/1 interface
- Receives an update from R1 about network 10.1.0.0. There is no change; therefore, the routing information remains the same
- Receives an update from R3 about network 10.4.0.0. There is no change; therefore, the routing information remains the same

R3:

- Sends an update about network 10.4.0.0 out the Serial 0/0/1 interface
- Sends an update about networks 10.2.0.0 and 10.3.0.0 out the FastEthernet0/0 interface
- Receives an update from R2 about network 10.1.0.0 and increments the hop count by 1
- Stores network 10.1.0.0 in the routing table with a metric of 2
- Same update from R2 contains information about network 10.2.0.0 with a metric of 1. There is no change; therefore, the routing information remains the same

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

After routers within a network have converged, the router can then use the information within the routing table to determine the best path to reach a destination. Different routing protocols have different ways of calculating the best path.

3.1.4.1.5 Achieving Convergence

The network has converged when all routers have complete and accurate information about the entire network, as shown in the figure. Convergence time is the time it takes routers to share information, calculate best paths, and update their routing tables. A network is not completely operable until the network has converged; therefore, most networks require short convergence times.

Convergence is both collaborative and independent. The routers share information with each other, but must independently calculate the impacts of the topology change on their own routes. Because they develop an agreement with the new topology independently, they are said to converge on this consensus.

Convergence properties include the speed of propagation of routing information and the calculation of optimal paths. The speed of propagation refers to the amount of time it takes for routers within the network to forward routing information.

Routing protocols can be rated based on the speed to convergence; the faster the convergence, the better the routing protocol. Generally, older protocols, such as RIP, are slow to converge, whereas modern protocols, such as EIGRP and OSPF, converge more quickly.

3.1.4.2 Distance Vector Routing Protocol Operation

3.1.4.2.1 Distance Vector Technologies

Distance vector routing protocols share updates between neighbors. Neighbors are routers that share a link and are configured to use the same routing protocol. The router is only aware of the network addresses of its own interfaces and the remote network addresses it can reach through its neighbors. Routers using distance vector routing are not aware of the network topology.

Some distance vector routing protocols send periodic updates. For example, RIP sends a periodic update to all of its neighbors every 30 seconds. RIP does this even if the topology has not changed. RIPv1 sends these updates as broadcasts to the all-hosts IPv4 address of 255.255.255.255.

The broadcasting of periodic updates is inefficient because the updates consume bandwidth and network device CPU resources. Every network device has to process a broadcast message. Instead of using broadcasts like RIP, RIPv2 and EIGRP can use multicast addresses to reach only specific neighbor routers. EIGRP can also use a unicast message to reach one specific neighbor router. Additionally, EIGRP only sends updates when needed, instead of periodically.

3.1.4.2.2 Distance Vector Algorithm

At the core of the distance vector protocol is the routing algorithm. The algorithm is used to calculate the best paths and then send that information to the neighbors.

The algorithm used for the routing protocols defines the following processes:

- Mechanism for sending and receiving routing information
- Mechanism for calculating the best paths and installing routes in the routing table
- Mechanism for detecting and reacting to topology changes

In the figure below, R1 and R2 are configured with the RIP routing protocol. The algorithm sends and receives updates. Both R1 and R2 then glean new information from the update. In this case, each router learns about a new network. The algorithm on each router makes its calculations independently and updates the routing table with the new information. When the LAN on R2 goes down, the algorithm constructs a triggered update and sends it to R1. R1 then removes the network from the routing table.

Different routing protocols use different algorithms to install routes in the routing table, send updates to neighbors, and make path determination decisions. For example:

- RIP uses the Bellman-Ford algorithm as its routing algorithm. It is based on two algorithms developed in 1958 and 1956 by Richard Bellman and Lester Ford, Jr.
- IGRP and EIGRP use the Diffusing Update Algorithm (DUAL) routing algorithm developed by Dr. J.J. Garcia-Luna-Aceves at SRI International.

3.1.4.3 Types of Distance Vector Routing Protocols

3.1.4.3.1 Routing Information Protocol

The Routing Information Protocol (RIP) was a first generation routing protocol for IPv4 originally specified in RFC 1058. It is easy to configure, making it a good choice for small networks.

RIPv1 has the following key characteristics:

- Routing updates are broadcasted (255.255.255.255) every 30 seconds.
- The hop count is used as the metric for path selection.
- A hop count greater than 15 hops is deemed infinite (too far). That 15th hop router would not propagate the routing update to the next router.

In 1993, RIPv1 was updated to a classless routing protocol known as RIP version 2 (RIPv2). RIPv2 included the following improvements:

- **Classless routing protocol** - It supports VLSM and CIDR, because it includes the subnet mask in the routing updates.
- **Increased efficiency** - It forwards updates to multicast address 224.0.0.9, instead of the broadcast address 255.255.255.255.
- **Reduced routing entries** - It supports manual route summarization on any interface.
- **Secure** - It supports an authentication mechanism to secure routing table updates between neighbors.

RIP updates are encapsulated into a UDP segment, with both source and destination port numbers set to UDP port 520.

In 1997, the IPv6 enabled version of RIP was released. RIPng is based on RIPv2. It still has a 15 hop limitation and the administrative distance is 120.

3.1.4.3.2 Enhanced Interior-Gateway Routing Protocol

The Interior Gateway Routing Protocol (IGRP) was the first proprietary IPv4 routing protocol developed by Cisco in 1984. It used the following design characteristics:

- Bandwidth, delay, load, and reliability are used to create a composite metric.
- Routing updates are broadcast every 90 seconds, by default.
- Maximum limit of 255 hops

In 1992, IGRP was replaced by Enhanced IGRP (EIGRP). Like RIPv2, EIGRP also introduced support for VLSM and CIDR. EIGRP increases efficiency, reduces routing updates, and supports secure message exchange.

EIGRP also introduced:

- **Bounded triggered updates** - It does not send periodic updates. Only routing table changes are propagated, whenever a change occurs. This reduces the amount of load the routing protocol

places on the network. Bounded triggered updates means that EIGRP only sends to the neighbors that need it. It uses less bandwidth, especially in large networks with many routes.

- **Hello keepalive mechanism** - A small Hello message is periodically exchanged to maintain adjacencies with neighboring routers. This requires a very low usage of network resources during normal operation, as compared to periodic updates.
- **Maintains a topology table** - Maintains all the routes received from neighbors (not only the best paths) in a topology table. DUAL can insert backup routes into the EIGRP topology table.
- **Rapid convergence** - In most cases, it is the fastest IGP to converge because it maintains alternate routes, enabling almost instantaneous convergence. If a primary route fails, the router can use the already identified alternate route. The switchover to the alternate route is immediate and does not involve interaction with other routers.
- **Multiple network layer protocol support** - EIGRP uses Protocol Dependent Modules (PDM), which means that it is the only protocol to include support for protocols other than IPv4 and IPv6, such as legacy IPX and AppleTalk.

3.1.5 Link-State Dynamic Routing

3.1.5.1 Link-State Routing Protocol Operation

3.1.5.1.1 Shortest Path First Protocols

Link-state routing protocols are also known as shortest path first protocols and are built around Edsger Dijkstra's shortest path first (SPF) algorithm. The SPF algorithm is discussed in more detail in a later section.

The IPv4 link-state routing protocols are:

- Open Shortest Path First (OSPF)
- Intermediate System-to-Intermediate System (IS-IS)

Link-state routing protocols have the reputation of being much more complex than their distance vector counterparts. However, the basic functionality and configuration of link-state routing protocols is equally straight-forward.

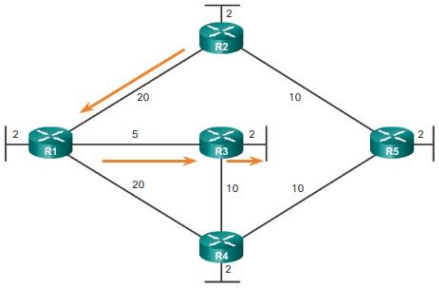
Just like RIP and EIGRP, basic OSPF operations can be configured using the:

- **router ospf** *process-id* global configuration command
- **network** command to advertise networks

3.1.5.1.2 Dijkstra's Algorithm

All link-state routing protocols apply Dijkstra’s algorithm to calculate the best path route. The algorithm is commonly referred to as the shortest path first (SPF) algorithm. This algorithm uses accumulated costs along each path, from source to destination, to determine the total cost of a route.

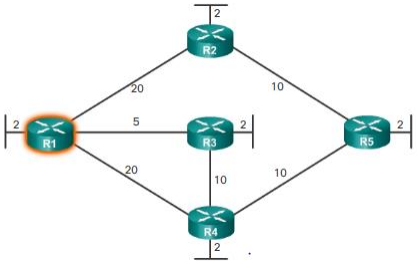
In the figure below, each path is labeled with an arbitrary value for cost. The cost of the shortest path for R2 to send packets to the LAN attached to R3 is 27. Each router determines its own cost to each destination in the topology. In other words, each router calculates the SPF algorithm and determines the cost from its own perspective.



Note: The focus of this section is on cost, which is determined by the SPF tree. For this reason, the graphics throughout this section show the connections of the SPF tree, not the topology. All links are represented with a solid black line.

3.1.5.1.3 SPF Example

The table in Figure below displays the shortest path and the accumulated cost to reach the identified destination networks from the perspective of R1.



The shortest path is not necessarily the path with the least number of hops. For example, look at the path to the R5 LAN. It might be assumed that R1 would send directly to R4 instead of to R3. However, the cost to reach R4 directly (22) is higher than the cost to reach R4 through R3 (17).

3.1.5.2 Link-State Updates

3.1.5.2.1 Link-State Routing Process

So exactly how does a link-state routing protocol work? With link-state routing protocols, a link is an interface on a router. Information about the state of those links is known as link-states.

All routers in an OSPF area will complete the following generic link-state routing process to reach a state of convergence:

1. Each router learns about its own links and its own directly connected networks. This is done by detecting that an interface is in the up state.

2. Each router is responsible for meeting its neighbors on directly connected networks. Link state routers do this by exchanging Hello packets with other link-state routers on directly connected networks.

3. Each router builds a Link-State Packet (LSP) containing the state of each directly connected link. This is done by recording all the pertinent information about each neighbor, including neighbor ID, link type, and bandwidth.

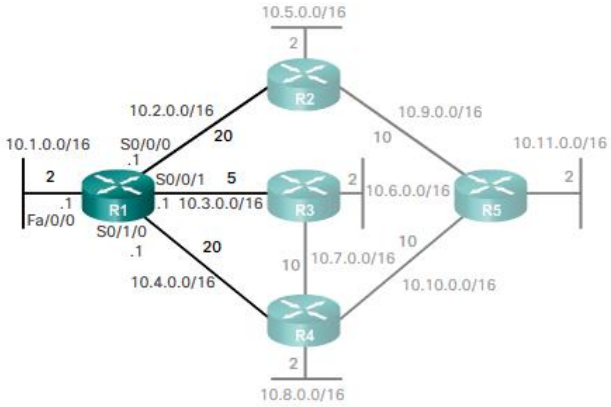
4. Each router floods the LSP to all neighbors. Those neighbors store all LSPs received in a database. They then flood the LSPs to their neighbors until all routers in the area have received the LSPs. Each router stores a copy of each LSP received from its neighbors in a local database.

5. Each router uses the database to construct a complete map of the topology and computes the best path to each destination network. Like having a road map, the router now has a complete map of all destinations in the topology and the routes to reach them. The SPF algorithm is used to construct the map of the topology and to determine the best path to each network.

Note: This process is the same for both OSPF for IPv4 and OSPF for IPv6. The examples in this section will refer to OSPF for IPv4.

3.1.5.2.2 Link and Link-State

The first step in the link-state routing process is that each router learns about its own links, its own directly connected networks. When a router interface is configured with an IP address and subnet mask, the interface becomes part of that network.



Refer to the topology in Figure above. For purposes of this discussion, assume that R1 was previously configured and had full connectivity to all neighbors. However, R1 lost power briefly and had to restart.

During boot up R1 loads the saved startup configuration file. As the previously configured interfaces become active, R1 learns about its own directly connected networks. Regardless of the routing protocols used, these directly connected networks are now entries in the routing table.

As with distance vector protocols and static routes, the interface must be properly configured with an IPv4 address and subnet mask, and the link must be in the up state before the link-state routing protocol can learn about a link. Also, like distance vector protocols, the interface must be included in one of the **network** router configuration statements before it can participate in the link-state routing process.

Figure above shows R1 linked to four directly connected networks:

- FastEthernet 0/0 - 10.1.0.0/16
- Serial 0/0/0 - 10.2.0.0/16
- Serial 0/0/1 - 10.3.0.0/16
- Serial 0/1/0 - 10.4.0.0/16

Note: Cisco's implementation of OSPF specifies the OSPF routing metric as the cost of the link based on the bandwidth of the outgoing interface. For the purposes of this chapter, we are using arbitrary cost values to simplify the demonstration.

3.1.5.2.3 Say Hello

The second step in the link-state routing process is that each router is responsible for meeting its neighbors on directly connected networks.

Routers with link-state routing protocols use a Hello protocol to discover any neighbors on its links. A neighbor is any other router that is enabled with the same link-state routing protocol.

R1 sends Hello packets out of its links (interfaces) to discover any neighbors. R2, R3, and R4 reply to the Hello packet with their own Hello packets because these routers are configured with the same link-state routing protocol. There are no neighbors out the FastEthernet 0/0 interface. Because R1 does not receive a Hello on this interface, it does not continue with the link-state routing process steps for the FastEthernet 0/0 link.

When two link-state routers learn that they are neighbors, they form an adjacency. These small Hello packets continue to be exchanged between two adjacent neighbors and serve as a keepalive function to monitor the state of the neighbor. If a router stops receiving Hello packets from a neighbor, that neighbor is considered unreachable and the adjacency is broken.

3.1.5.2.4 Building the Link-State Packet

The third step in the link-state routing process is that each router builds a link-state packet (LSP) containing the state of each directly connected link.

After a router has established its adjacencies, it can build its LSP that contains the link-state information about its links. A simplified version of the LSP from R1 displayed in the figure above would contain the following:
1. R1; Ethernet network 10.1.0.0/16; Cost 2
2. R1 -> R2; Serial point-to-point network; 10.2.0.0/16; Cost 20
3. R1 -> R3; Serial point-to-point network; 10.3.0.0/16; Cost 5
4. R1 -> R4; Serial point-to-point network; 10.4.0.0/16; Cost 20

3.1.5.2.5 Flooding the LSP

The fourth step in the link-state routing process is that each router floods the LSP to all neighbors, who then store all LSPs received in a database.

Each router floods its link-state information to all other link-state routers in the routing area. Whenever a router receives an LSP from a neighboring router, it immediately sends that LSP out all other interfaces except the interface that received the LSP. This process creates a flooding effect of LSPs from all routers throughout the routing area.

Remember that LSPs do not need to be sent periodically. An LSP only needs to be sent:

- During initial startup of the routing protocol process on that router (e.g., router restart)
- Whenever there is a change in the topology (e.g., a link going down or coming up, a neighbor adjacency being established or broken)

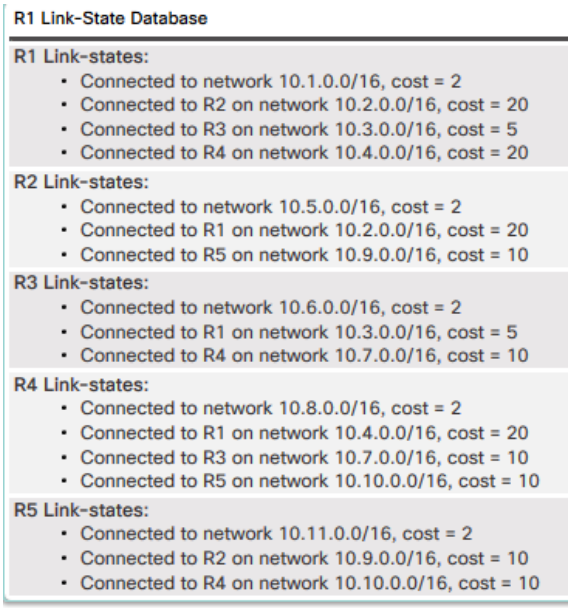
In addition to the link-state information, other information is included in the LSP, such as sequence numbers and aging information, to help manage the flooding process. This information is used by each router to determine if it has already received the LSP from another router or if the LSP has newer information than what is already contained in the link-state database. This process allows a router to keep only the most current information in its link-state database.

3.1.5.2.6 Building the Link-State Database

The final step in the link-state routing process is that each router uses the database to construct a complete map of the topology and computes the best path to each destination network.

Eventually, all routers receive an LSP from every other link-state router in the routing area. These LSPs are stored in the link-state database.

The example in the figure displays the link-state database content of R1.



As a result of the flooding process, R1 has learned the link-state information for each router in its routing area. Notice that R1 also includes its own link-state information in the link-state database.

With a complete link-state database, R1 can now use the database and the shortest path first (SPF) algorithm to calculate the preferred path or shortest path to each network resulting in the SPF tree.

3.1.5.2.7 Building the SPF Tree

Each router in the routing area uses the link-state database and SPF algorithm to construct the SPF tree.

For example, using the link-state information from all other routers, R1 can now begin to construct an SPF tree of the network. To begin, the SPF algorithm interprets each router's LSP to identify networks and associated costs.

The SPF algorithm then calculates the shortest paths to reach each individual network resulting in the SPF tree. R1 now has a complete topology view of the link-state area.

Each router constructs its own SPF tree independently from all other routers. To ensure proper routing, the link-state databases used to construct those trees must be identical on all routers.

3.1.5.2.8 Adding OSPF Routes to the Routing Table

Using the shortest path information determined by the SPF algorithm, these paths can now be added to the routing table. The figure shows the routes that have now been added to R1's IPv4 routing table.

The routing table also includes all directly connected networks and routes from any other sources, such as static routes. Packets are now forwarded according to these entries in the routing table.

3.1.5.3 Link-State Routing Protocol benefits

3.1.5.3.1 Why Use Link-State Protocols?

There are several advantages of link-state routing protocols compared to distance vector routing protocols.

- **Builds a Topological Map** - Link-state routing protocols create a topological map, or SPF tree, of the network topology. Because link-state routing protocols exchange link-states, the SPF algorithm can build an SPF tree of the network. Using the SPF tree, each router can independently determine the shortest path to every network.
- **Fast Convergence** - When receiving an LSP, link-state routing protocols immediately flood the LSP out all interfaces except for the interface from which the LSP was received. In contrast, RIP needs to process each routing update and update its routing table before flooding them out other interfaces.
- **Event-driven Updates** - After the initial flooding of LSPs, link-state routing protocols only send out an LSP when there is a change in the topology. The LSP contains only the information regarding the affected link. Unlike some distance vector routing protocols, link-state routing protocols do not send periodic updates.
- **Hierarchical Design** - Link-state routing protocols use the concept of areas. Multiple areas create a hierarchical design to networks, allowing for better route aggregation (summarization) and the isolation of routing issues within an area.

3.1.5.3.2 Disadvantages of Link-State Protocols

Link-state protocols also have a few disadvantages compared to distance vector routing protocols:

- **Memory Requirements** - Link-state protocols require additional memory to create and maintain the link-state database and SPF tree.
- **Processing Requirements** - Link-state protocols can also require more CPU processing than distance vector routing protocols. The SPF algorithm requires more CPU time than distance vector

algorithms such as Bellman-Ford, because link-state protocols build a complete map of the topology.

- **Bandwidth Requirements** - The flooding of link-state packets can adversely affect the available bandwidth on a network. This should only occur during initial startup of routers, but can also be an issue on unstable networks.

However, modern link-state routing protocols are designed to minimize the effects on memory, CPU, and bandwidth. The use and configuration of multiple areas can reduce the size of the link-state databases. Multiple areas can limit the amount of link-state information flooding in a routing domain and send LSPs only to those routers that need them. When there is a change in the topology, only those routers in the affected area receive the LSP and run the SPF algorithm. This can help isolate an unstable link to a specific area in the routing domain.

3.1.5.3.3 Protocols that Use Link-State

There are only two link-state routing protocols, OSPF and IS-IS.

Open Shortest Path First (OSPF) is the most popular implementation. It was designed by the Internet Engineering Task Force (IETF) OSPF Working Group. The development of OSPF began in 1987 and there are two current versions in use:

- OSPFv2- OSPF for IPv4 networks (RFC 1247 and RFC 2328)
- OSPFv3- OSPF for IPv6 networks (RFC 2740)

Note: With the OSPFv3 Address Families feature, OSPFv3 includes support for both IPv4 and IPv6.

Intermediate System to Intermediate System (IS-IS) was designed by the International Organization for Standardization (ISO) and is described in ISO 10589. The first incarnation of this routing protocol was developed at Digital Equipment Corporation (DEC) and is known as DECnet Phase V. Radia Perlman was the chief designer of the IS-IS routing protocol.

IS-IS was originally designed for the OSI protocol suite and not the TCP/IP protocol suite. Later, Integrated IS-IS, or Dual IS-IS, included support for IP networks. Although IS-IS has been known as the routing protocol used mainly by ISPs and carriers, more enterprise networks are beginning to use IS-IS.

OSPF and IS-IS share many similarities, but also have several differences. There are pro-OSPF and pro-IS-IS factions who discuss and debate the advantages of one routing protocol over the other. However, both routing protocols provide the necessary routing functionality for a large enterprise or ISP.

Note: Further study of IS-IS is beyond the scope of this course.

3.2 RIPv2

3.2.1 Configuring the RIP Protocol

3.2.1.1 Router RIP Configuration Mode

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

In this scenario, all routers have been configured with basic management features and all interfaces identified in the reference topology are configured and enabled. There are no static routes configured and no routing protocols enabled; therefore, remote network access is currently impossible. RIPv1 is used as the dynamic routing protocol. To enable RIP, use the **router rip** command. This command does not directly start the RIP process. Instead, it provides access to the router configuration mode where the RIP routing settings are configured. When enabling RIP, the default version is RIPv1.

To disable and eliminate RIP, use the **no router rip** global configuration command. This command stops the RIP process and erases all existing RIP configurations.

3.2.1.2 Advertise Networks

By entering the RIP router configuration mode, the router is instructed to run RIPv1. But the router still needs to know which local interfaces it should use for communication with other routers, as well as which locally connected networks it should advertise to those routers.

To enable RIP routing for a network, use the **network** *network-address* router configuration mode command. Enter the classful network address for each directly connected network. This command:

- Enables RIP on all interfaces that belong to a specific network. Associated interfaces now both send and receive RIP updates.
- Advertises the specified network in RIP routing updates sent to other routers every 30 seconds.

Note: RIPv1 is a classful routing protocol for IPv4. Therefore, if a subnet address is entered, the IOS automatically converts it to the classful network address. For example, entering the **network 192.168.1.32** command would automatically be converted to **network 192.168.1.0** in the running configuration file. The IOS does not give an error message, but instead corrects the input and enters the classful network address.

3.2.1.3 Verify RIP Routing

The **show ip protocols** command displays the IPv4 routing protocol settings currently configured on the router.

Note: This command is also very useful when verifying the operations of other routing protocols (i.e., EIGRP and OSPF). The **show ip route** command displays the RIP routes installed in the routing table.

3.2.1.4 Enable and Verify RIPv2

By default, when a RIP process is configured on a Cisco router, it is running RIPv1. However, even though the router only sends RIPv1 messages, it can interpret both RIPv1 and RIPv2 messages. A RIPv1 router ignores the RIPv2 fields in the route entry.

Use the **version 2** router configuration mode command to enable RIPv2. Notice how the **show ip protocols** command verifies that R2 is now configured to send and receive version 2 messages only. The RIP process now includes the subnet mask in all updates, making RIPv2 a classless routing protocol.

Note: Configuring **version 1** enables RIPv1 only, while configuring **no version** returns the router to the default setting of sending version 1 updates but listening for version 1 and version 2 updates.

Therefore, the **version 2** command must be configured on all routers in the routing domain.

3.2.1.5 Disable Auto Summarization

RIPv2 automatically summarizes networks at major network boundaries by default, just like RIPv1.

To modify the default RIPv2 behavior of automatic summarization, use the **no auto-summary** router configuration mode command. This command has no effect when using RIPv1. When automatic summarization has been disabled, RIPv2 no longer summarizes networks to their classful address at boundary routers. RIPv2 now includes all subnets and their appropriate masks in its routing updates. The **show ip protocols** now states that "automatic network summarization is not in effect".

Note: RIPv2 must be enabled before automatic summarization is disabled.

3.2.1.6 Configure Passive Interfaces

By default, RIP updates are forwarded out all RIP-enabled interfaces. However, RIP updates really only need to be sent out interfaces that are connected to other RIPenabled routers.

For instance, RIP sends updates out of its G0/0 interface even though no RIP device exists on that LAN. R1 has no way of knowing this and, as a result, sends an update every 30 seconds. Sending out unneeded updates on a LAN impacts the network in three ways:

- **Wasted Bandwidth** - Bandwidth is used to transport unnecessary updates. Because RIP updates are either broadcasted or multicasted, switches also forward the updates out all ports.
- **Wasted Resources** - All devices on the LAN must process the update up to the transport layers, at which point the devices will discard the update.
- **Security Risk** - Advertising updates on a broadcast network is a security risk. RIP updates can be intercepted with packet sniffing software. Routing updates can be modified and sent back to the router, corrupting the routing table with false metrics that misdirect traffic.

Use the **passive-interface** router configuration command to prevent the transmission of routing updates through a router interface, but still allow that network to be advertised to other routers. The command stops routing updates out the specified interface. However, the network that the specified interface belongs to is still advertised in routing updates that are sent out other interfaces.

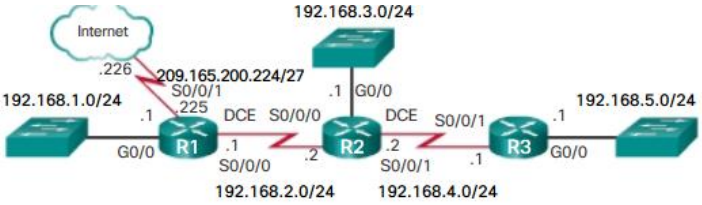
The **show ip protocols** command is then used to verify that the Gigabit Ethernet interface was passive. Notice that the G0/0 interface is no longer listed as sending or receiving version 2 updates, but instead is now listed under the Passive Interface(s) section. Also notice that the network 192.168.1.0 is still listed under Routing for Networks, which means that this network is still included as a route entry in RIP updates that are sent to R2.

Note: All routing protocols support the **passive-interface** command.

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

3.2.1.7 Propagate a Default Route

Refer to Figure. In this scenario, R1 is the edge router, single-homed to a service provider. Therefore, all that is required for R1 to reach the Internet is a default static route going out of the Serial 0/0/1 interface.



Similar default static routes could be configured on R2 and R3, but it is much more scalable to enter it one time on the edge router R1 and then have R1 propagate it to all other routers using RIP. To provide Internet connectivity to all other networks in the RIP routing domain, the default static route needs to be advertised to all other routers that use the dynamic routing protocol.

To propagate a default route in RIP, the edge router must be configured with:

- A default static route using the **ip route 0.0.0.0 0.0.0.0** command.
- The **default-information originate** router configuration command. This instructs R1 to originate default information, by propagating the static default route in RIP updates.

3.3 The Routing Table

3.3.1 Parts of an IPv4 Route Entry

3.3.1.1 Routing Table Entries

Notice that in the topology:

- R1 is the edge router that connects to the Internet; therefore, it is propagating a default static route to R2 and R3.
- R1, R2, and R3 contain discontiguous networks separated by another classful network.

- R3 is also introducing a 192.168.0.0/16 supernet route.

Note: The routing table hierarchy in Cisco IOS was originally implemented with the classful routing scheme. Although the routing table incorporates both classful and classless addressing, the overall structure is still built around this classful scheme.

3.3.1.2 Directly Connected Entries

The routing table of R1 contains three directly connected networks. Notice that two routing table entries are automatically created when an active router interface is configured with an IP address and subnet mask.

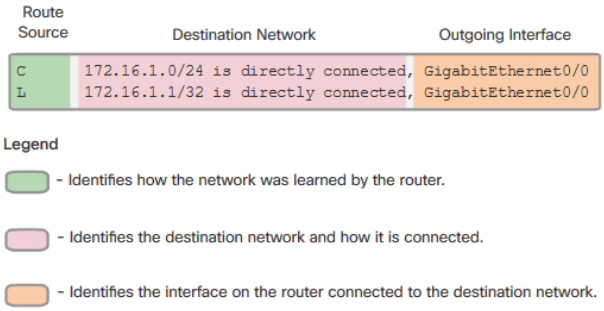


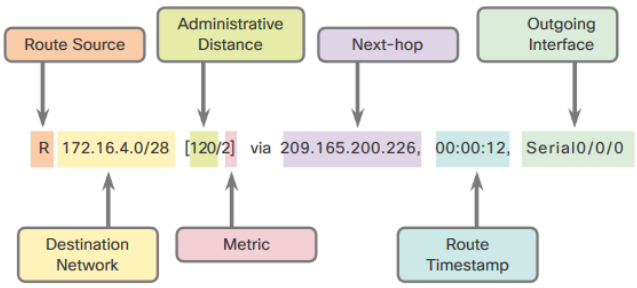
Figure above displays one of the routing table entries on R1 for the directly connected network 172.16.1.0. These entries were automatically added to the routing table when the GigabitEthernet 0/0 interface was configured and activated. The entries contain the following information:

- **Route source** - Identifies how the route was learned. Directly connected interfaces have two route source codes. **C** identifies a directly connected network. Directly connected networks are automatically created whenever an interface is configured with an IP address and activated. **L** identifies that this is a local route. Local routes are automatically created whenever an interface is configured with an IP address and activated.
- **Destination network** - The address of the remote network and how that network is connected.
- **Outgoing interface** - Identifies the exit interface to use when forwarding packets to the destination network.

A router typically has multiple interfaces configured. The routing table stores information about both directly connected and remote routes. As with directly connected networks, the route source identifies how the route was learned. For instance, common codes for remote networks include:

- **S** - Identifies that the route was manually created by an administrator to reach a specific network. This is known as a static route.
- **D** - Identifies that the route was learned dynamically from another router using the EIGRP routing protocol.
- - Identifies that the route was learned dynamically from another router using the OSPF routing protocol.
- **R** - Identifies that the route was learned dynamically from another router using the RIP routing protocol.

3.3.1.3 Remote Network Entries



The figure displays an IPv4 routing table entry on R1 for the route to remote network 172.16.4.0 on R3. The entry identifies the following information:

- **Route source** - Identifies how the route was learned.
- **Destination network** - Identifies the address of the remote network.
- **Administrative distance (AD)** - Identifies the trustworthiness of the route source. The AD for static routes is 1 and the AD for connected routes is 0. Dynamic routing protocols have an AD higher than 1 depending upon the protocol.
- **Metric** - Identifies the value assigned to reach the remote network. Lower values indicate preferred routes. The metric for static and connected routes is 0.
- **Next hop** - Identifies the IPv4 address of the next router to forward the packet to.
- **Route timestamp** - Identifies from when the route was last heard.
- **Outgoing interface** - Identifies the exit interface to use to forward a packet toward the final destination.

3.3.2 Dynamically Learned IPv4 Routes

3.3.2.1 Routing Table Terms

A dynamically built routing table provides a great deal of information. Therefore, it is crucial to understand the output generated by the routing table. Special terms are applied when discussing the contents of a routing table.

The Cisco IP routing table is not a flat database. The routing table is actually a hierarchical structure that is used to speed up the lookup process when locating routes and forwarding packets. Within this structure, the hierarchy includes several levels.

Routes are discussed in terms of:

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

3.3.2.2 Ultimate Route

An ultimate route is a routing table entry that contains either a next-hop IPv4 address or an exit interface. Directly connected, dynamically learned, and local routes are ultimate routes.

3.3.2.3 Level 1 Route

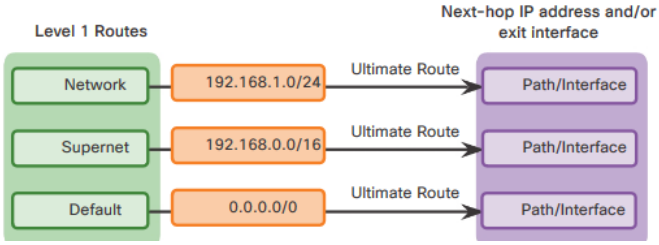
A level 1 route is a route with a subnet mask equal to or less than the classful mask of the network address. Therefore, a level 1 route can be a:

- **Network route** - A network route that has a subnet mask equal to that of the classful mask.
- **Supernet route** - A supernet route is a network address with a mask less than the classful mask, for example, a summary address.
- **Default route** - A default route is a static route with the address 0.0.0.0/0.

The source of the level 1 route can be a directly connected network, static route, or a dynamic routing protocol.

3.3.2.4 Level 1 Parent Route

As illustrated in Figure below, the 172.16.0.0 and 209.165.200.0 routes are level 1 parent routes. A parent route is a level 1 network route that is subnetted. A parent route can never be an ultimate route.



In the routing table, it basically provides a heading for the specific subnets it contains. Each entry displays the classful network address, the number of subnets and the number of different subnet masks into which the classful address has been subdivided.

3.3.2.5 Level 2 Child Route

A level 2 child route is a route that is a subnet of a classful network address. As illustrated in Figure 1, a level 1 parent route is a level 1 network route that is subnetted. Level 1 parent routes contain level 2 child routes.

Like a level 1 route, the source of a level 2 route can be a directly connected network, a static route, or a dynamically learned route. Level 2 child routes are also ultimate routes.

Note: The routing table hierarchy in Cisco IOS has a classful routing scheme. A level 1 parent route is the classful network address of the subnet route. This is the case even if a classless routing protocol is the source of the subnet route.

3.3.3 The IPv4 Route Lookup Process

3.3.3.1 Route Lookup Process

When a packet arrives on a router interface, the router examines the IPv4 header, identifies the destination IPv4 address, and proceeds through the router lookup process.

The router examines level 1 network routes for the best match with the destination address of the IPv4 packet:

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

2. If the best match is a level 1 parent route, proceed to the next step.

The router examines child routes (the subnet routes) of the parent route for a best match:

3. If there is a match with a level 2 child route, that subnet is used to forward the packet.

4. If there is not a match with any of the level 2 child routes, proceed to the next step.

The router continues searching level 1 supernet routes in the routing table for a match, including the default route, if there is one:

5. If there is now a lesser match with a level 1 supernet or default routes, the router uses that route to forward the packet.

6. If there is not a match with any route in the routing table, the router drops the packet.

Note: A route referencing only a next-hop IP address and not an exit interface, must be resolved to a route with an exit interface, if Cisco Express Forwarding (CEF) is not being used. Without CEF, a recursive lookup is performed on the next-hop IP address until the route is resolved to an exit interface. CEF is enabled by default.

3.3.3.2 Best Route = Longest Match

What is meant by the router must find the best match in the routing table? Best match is equal to the longest match.

For there to be a match between the destination IPv4 address of a packet and a route in the routing table, a minimum number of far left bits must match between the IPv4 address of the packet and the route in the routing table. The subnet mask of the route in the routing table is used to determine the minimum number of far left bits that must match. Remember that an IPv4 packet only contains the IPv4 address and not the subnet mask.

The best match is the route in the routing table that has the most number of far left matching bits with the destination IPv4 address of the packet. The route with the greatest number of equivalent far left bits, or the longest match, is always the preferred route.

3.3.4 Analyze an IPv6 Routing Table

3.3.4.1 IPv6 Routing Table Entries

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.

Because IPv6 is classless by design, all routes are effectively level 1 ultimate routes. There is no level 1 parent of level 2 child routes.

Note: Although EIGRP for IPv6 is used to populate the routing tables, the operation and configuration of EIGRP is beyond the scope of this course.

3.3.4.2 Directly Connected Entries

The routing table is displayed using the **show ipv6 route** command. Although, the command output is displayed slightly differently than in the IPv4 version, it still contains the relevant route information.

Directly connected route entries display the following information:

- **Route source** - Identifies how the route was learned. Directly connected interfaces have two route source codes (C identifies a directly connected network while L identifies that this is a local route.)
- **Directly connected network** - The IPv6 address of the directly connected network.
- **Administrative distance** - Identifies the trustworthiness of the route source. IPv6 uses the same distances as IPv4. A value of 0 indicates the best, most trustworthy source.
- **Metric** - Identifies the value assigned to reach the remote network. Lower values indicate preferred routes.
- **Outgoing interface** - Identifies the exit interface to use when forwarding packets to the destination network.

Note: The serial links have reference bandwidths configured to observe how EIGRP metrics select the best route. The reference bandwidth is not a realistic representation of modern networks. It is used only to provide a visual depiction of link speed.

3.4 EIGRP Characteristics

3.4.1 EIGRP Basic Features

3.4.1.1 Features of EIGRP

EIGRP was initially released in 1992 as a proprietary protocol available only on Cisco devices. However, in 2013, Cisco released a basic functionality of EIGRP as an open standard to the IETF, as an informational RFC. This means that other networking vendors can now implement EIGRP on their equipment to interoperate with both Cisco and non-Cisco routers running EIGRP. However, advanced features of EIGRP, such as EIGRP stub, needed for the Dynamic Multipoint Virtual Private Network (DMVPN) deployment, will not be released to the IETF. As an informational RFC, Cisco will continue to maintain control of EIGRP.

EIGRP includes features of both link-state and distance vector routing protocols. However, EIGRP is still based on the key distance vector routing protocol principle, in which information about the rest of the network is learned from directly connected neighbors.

EIGRP is an advanced distance vector routing protocol that includes features not found in other distance vector routing protocols like RIP and IGRP.

In Cisco IOS Release 15.0(1)M, Cisco introduced a new EIGRP configuration option called **named EIGRP**. Named EIGRP enables the configuration of EIGRP for both IPv4 and IPv6 under a single configuration mode. This helps eliminate configuration complexity that occurs when configuring EIGRP for both IPv4 and IPv6. Named EIGRP is beyond the scope of this course.

Features of EIGRP include:

- **Diffusing Update Algorithm -** As the computational engine that drives EIGRP, the Diffusing Update Algorithm (DUAL) resides at the center of the routing protocol. DUAL guarantees loop-free and backup paths throughout the routing domain. Using DUAL, EIGRP stores all available backup routes for destinations so that it can quickly adapt to alternate routes when necessary.
- **Establishing Neighbor Adjacencies -** EIGRP establishes relationships with directly connected routers that are also enabled for EIGRP. Neighbor adjacencies are used to track the status of these neighbors.
- **Reliable Transport Protocol -** The Reliable Transport Protocol (RTP) is unique to EIGRP and provides delivery of EIGRP packets to neighbors. RTP and the tracking of neighbor adjacencies set the stage for DUAL.
- **Partial and Bounded Updates -** EIGRP uses the terms partial and bounded when referring to its updates. Unlike RIP, EIGRP does not send periodic updates and route entries do not age out. The term partial means that the update only includes information about the route changes, such as a new link or a link becoming unavailable. The term bounded refers to the propagation of partial updates that are sent only to those routers that the changes affect. This minimizes the bandwidth that is required to send EIGRP updates.
- **Equal and Unequal Cost Load Balancing -** EIGRP supports equal cost load balancing and unequal cost load balancing, which allows administrators to better distribute traffic flow in their networks.

Note: The term "hybrid routing" protocol may be used in some older documentation to define EIGRP. However, this term is misleading because EIGRP is not a hybrid between distance vector and link-state routing protocols. EIGRP is solely a distance vector routing protocol; therefore, Cisco no longer uses this term to refer to it.

3.3.1.2 Protocol Dependent Modules

EIGRP has the capability for routing different protocols, including IPv4 and IPv6. EIGRP does so by using protocol-dependent modules (PDMs). PDMs were also used to support the now obsolete Novell IPX and Apple Computer's AppleTalk network layer protocols.

PDMs are responsible for network layer protocol-specific tasks. An example is the EIGRP module that is responsible for sending and receiving EIGRP packets that are encapsulated in IPv4. This module is also responsible for parsing EIGRP packets and informing DUAL of the new information that is received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IPv4 routing table.

PDMs are responsible for the specific routing tasks for each network layer protocol, including:

- Maintaining the neighbor and topology tables of EIGRP routers that belong to that protocol suite
- Building and translating protocol-specific packets for DUAL

- Interfacing DUAL to the protocol-specific routing table
- Computing the metric and passing this information to DUAL
- Implementing filtering and access lists
- Performing redistribution functions to and from other routing protocols
- Redistributing routes that are learned by other routing protocols

When a router discovers a new neighbor, it records the neighbor's address and interface as an entry in the neighbor table. One neighbor table exists for each protocol-dependent module, such as IPv4. EIGRP also maintains a topology table. The topology table contains all destinations that are advertised by neighboring routers. There is also a separate topology table for each PDM.

3.4.1.3 Reliable Transport Protocol

EIGRP was designed as a network layer independent routing protocol. Because of this design, EIGRP cannot use the services of UDP or TCP. Instead, EIGRP uses the Reliable Transport Protocol (RTP) for the delivery and reception of EIGRP packets. This allows EIGRP to be flexible and can be used for protocols other than those from the TCP/IP protocol suite, such as the now obsolete IPX and AppleTalk protocols.

Although "reliable" is part of its name, RTP includes both reliable delivery and unreliable delivery of EIGRP packets, similar to TCP and UDP, respectively. Reliable RTP requires an acknowledgment to be returned by the receiver to the sender. An unreliable RTP packet does not require an acknowledgment. For example, an EIGRP update packet is sent reliably over RTP and requires an acknowledgment. An EIGRP Hello packet is also sent over RTP, but unreliably. This means that EIGRP Hello packets do not require an acknowledgment.

RTP can send EIGRP packets as unicast or multicast.

- Multicast EIGRP packets for IPv4 use the reserved IPv4 multicast address 224.0.0.10.
- Multicast EIGRP packets for IPv6 are sent to the reserved IPv6 multicast address FF02::A.

3.4.1.4 Authentication

Like other routing protocols, EIGRP can be configured for authentication. RIPv2, EIGRP, OSPF, IS-IS, and BGP can each be configured to authenticate their routing information.

It is a good practice to authenticate transmitted routing information. Doing so ensures that routers only accept routing information from other routers that have been configured with the same password or authentication information.

Note: Authentication does not encrypt the EIGRP routing updates.

3.4.2 EIGRP Packet Types

3.4.2.1 EIGRP Packet Types

EIGRP uses five different packet types, some in pairs. EIGRP packets are sent using either RTP reliable or unreliable delivery and can be sent as a unicast, multicast, or sometimes both. EIGRP packet types are also called EIGRP packet formats or EIGRP messages.

The five EIGRP packet types include:

Hello packets - Used for neighbor discovery and to maintain neighbor adjacencies.

- Sent with unreliable delivery
- Multicast (on most network types)

Update packets - Propagates routing information to EIGRP neighbors.

- Sent with reliable delivery
- Unicast or multicast

Acknowledgment packets - Used to acknowledge the receipt of an EIGRP message that was sent using reliable delivery.

- Sent with unreliable delivery
- Unicast

Query packets - Used to query routes from neighbors.

- Sent with reliable delivery
- Unicast or multicast

Reply packets - Sent in response to an EIGRP query.

- Sent with reliable delivery
- Unicast

3.4.2.2 EIGRP Hello Packets

EIGRP uses small Hello packets to discover other EIGRP-enabled routers on directly connected links. Hello packets are used by routers to form EIGRP neighbor adjacencies, also known as neighbor relationships.

EIGRP Hello packets are sent as IPv4 or IPv6 multicasts, and use RTP unreliable delivery. This means that the receiver does not reply with an acknowledgment packet.

- The reserved EIGRP multicast address for IPv4 is 224.0.0.10.
- The reserved EIGRP multicast address for IPv6 is FF02::A.

EIGRP routers discover neighbors and establish adjacencies with neighbor routers using the Hello packet. On most modern networks, EIGRP Hello packets are sent as multicast packets every five seconds. However, on multipoint, non-broadcast multiple access (NBMA) networks with access links of T1 (1.544 Mb/s) or slower, Hello packets are sent as unicast packets every 60 seconds.

Note: NBMA networks using slower interfaces include legacy X.25, Frame Relay, and Asynchronous Transfer Mode (ATM).

EIGRP also uses Hello packets to maintain established adjacencies. An EIGRP router assumes that as long as it receives Hello packets from a neighbor, the neighbor and its routes remain viable.

EIGRP uses a Hold timer to determine the maximum time the router should wait to receive the next Hello before declaring that neighbor as unreachable. By default, the hold time is three times the Hello interval, or 15 seconds on most networks and 180 seconds on low-speed NBMA networks. If the hold time expires, EIGRP declares the route as down and DUAL searches for a new path by sending out queries.

3.4.2.3 EIGRP Update and Acknowledgment Packets

EIGRP Update Packets

EIGRP sends Update packets to propagate routing information. Update packets are sent only when necessary. EIGRP updates contain only the routing information needed and are sent only to those routers that require it.

Unlike the older distance vector routing protocol RIP, EIGRP does not send periodic updates and route entries do not age out. Instead, EIGRP sends incremental updates only when the state of a destination changes. This may include when a new network becomes available, an existing network becomes unavailable, or a change occurs in the routing metric for an existing network.

EIGRP uses the terms *partial update* and *bounded update* when referring to its updates. A partial update means that the update only includes information about route changes. A bounded update refers to the sending of partial updates only to the routers that are affected by the changes. Bounded updates help EIGRP minimize the bandwidth that is required to send EIGRP updates.

EIGRP Update packets use reliable delivery, which means the sending router requires an acknowledgment. Update packets are sent as a multicast when required by multiple routers, or as a unicast when required by only a single router.

EIGRP Acknowledgment Packets

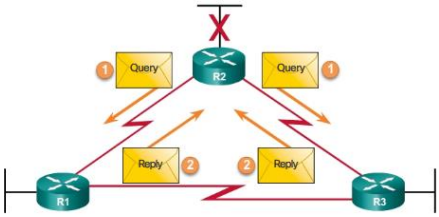
EIGRP sends Acknowledgment (ACK) packets when reliable delivery is used. An EIGRP acknowledgment is an EIGRP Hello packet without any data. RTP uses reliable delivery for Update, Query, and Reply packets. EIGRP Acknowledgment packets are always sent as an unreliable unicast. Unreliable delivery makes sense; otherwise, there would be an endless loop of acknowledgments.

Note: Some documentation refers to the Hello and acknowledgment as a single type of EIGRP packet.

3.4.2.4 EIGRP Query and Reply Packets

EIGRP Query Packets

DUAL uses Query and Reply packets when searching for networks and other tasks. Queries and replies use reliable delivery. Queries can use multicast or unicast, whereas replies are always sent as unicast.



In the figure, R2 has lost connectivity to the LAN and it sends out queries to all EIGRP neighbors searching for any possible routes to the LAN. Because queries use reliable delivery, the receiving router must return an EIGRP acknowledgment. The acknowledgment informs the sender of the query that it has received the query message. To keep this example simple, acknowledgments were omitted in the graphic.

EIGRP Reply Packets

All neighbors must send a reply, regardless of whether or not they have a route to the downed network. Because replies also use reliable delivery, routers, such as R2, must send an acknowledgment.

It may not be obvious why R2 would send out a query for a network it knows is down. Actually, only R2's interface that is attached to the network is down. Another router could be attached to the same LAN and have an alternate path to this same network. Therefore, R2 queries for such a router before completely removing the network from its topology table.

3.4.2 EIGRP Messages

3.4.3.1 Encapsulating EIGRP Messages

The data portion of an EIGRP message is encapsulated in a packet. This data field is called type, length, value (TLV). The types of TLVs relevant to this course are EIGRP parameters, IP internal routes, and IP external routes.

The EIGRP packet header is included with every EIGRP packet, regardless of its type. The EIGRP packet header and TLV are then encapsulated in an IPv4 packet. In the IPv4 packet header, the protocol field is set to 88 to indicate EIGRP, and the IPv4 destination address is set to the multicast 224.0.0.10. If the EIGRP packet is encapsulated in an Ethernet frame, the destination MAC address is also a multicast address, 01-00-5E-00-00-0A.

3.4.3.2 EIGRP Packet Header and TLV

Every EIGRP message includes the header. Important fields include the Opcode field and the Autonomous System Number field. Opcode specifies the EIGRP packet type as follows:

- Update

- Query
- Reply
- Hello

The autonomous system number specifies the EIGRP routing process. Unlike RIP, multiple instances of EIGRP can run on a network. The autonomous system number is used to track each running EIGRP process.

The EIGRP parameters message includes the weights that EIGRP uses for its composite metric. By default, only bandwidth and delay are weighted. Both are weighted equally; therefore, the K1 field for bandwidth and the K3 field for delay are both set to one (1). The other K values are set to zero (0).

The Hold Time is the amount of time the EIGRP neighbor receiving this message should wait before considering the advertising router to be down.

The IP internal message is used to advertise EIGRP routes within an autonomous system. Important fields include the metric fields (delay and bandwidth), the subnet mask field (prefix length), and the destination field.

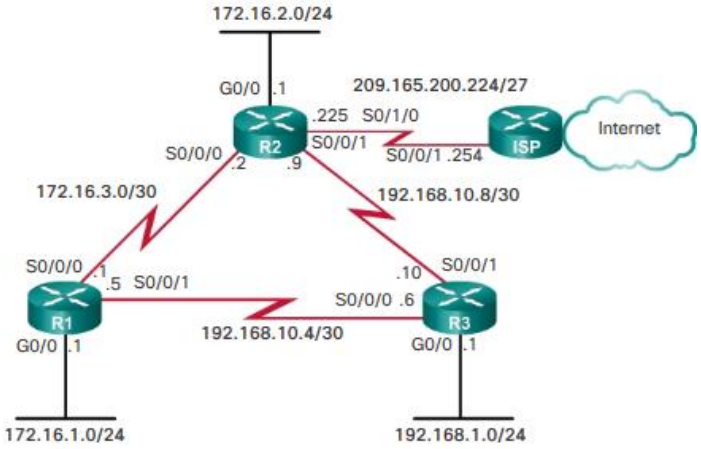
Delay is calculated as the sum of delays from source to destination in units of 10 microseconds. Bandwidth is the lowest configured bandwidth of any interface along the route.

The subnet mask is specified as the prefix length or the number of network bits in the subnet mask. For example, the prefix length for the subnet mask 255.255.255.0 is 24, because 24 is the number of network bits.

Note: The maximum transmission unit (MTU) is not a metric used by EIGRP. The MTU is included in the routing updates, but it is not used to determine the routing metric.

3.5 Implement EIGRP for IPv4
3.5.1 Configure EIGRP with IPv4

3.5.1.1 EIGRP Network Topology
Figure below displays the topology that is used in this chapter to configure EIGRP for IPv4.



The routers in the topology have a starting configuration that includes addresses on the interfaces. There is currently no static routing or dynamic routing configured on any of the routers.

3.5.1.2 Autonomous System Numbers
EIGRP uses the **router eigrp** *autonomous-system* command to enable the EIGRP process. The autonomous system number referred to in the EIGRP configuration is not associated with the Internet Assigned Numbers Authority (IANA) globally assigned autonomous system numbers used by external routing protocols.

So what is the difference between the IANA globally assigned autonomous system number and the EIGRP autonomous system number?

An IANA globally assigned autonomous system is a collection of networks under the administrative control of a single entity that presents a common routing policy to the Internet.

The guidelines for the creation, selection, and registration of an autonomous system are described in RFC 1930. Global autonomous system numbers are assigned by IANA, the same authority that assigns IP address space. The local regional Internet registry (RIR) is responsible for assigning an autonomous system number to an entity from its block of assigned autonomous system numbers. Prior to 2007, assigned autonomous system numbers were 16-bit numbers ranging from 0 to 65,535. Today, 32-bit autonomous system numbers are assigned thereby increasing the number of available autonomous system numbers to over 4 billion.

Usually, only Internet Service Providers (ISPs), Internet backbone providers, and large institutions connecting to other entities require an autonomous system number. These ISPs and large institutions use the exterior gateway routing protocol, Border Gateway Protocol (BGP), to propagate routing information. BGP is the only routing protocol that uses an actual autonomous system number in its configuration.

The vast majority of companies and institutions with IP networks do not need an autonomous system number, because they are controlled by a larger entity, such as an ISP. These companies use interior gateway protocols, such as RIP, EIGRP, OSPF, and IS-IS to route packets within their own networks. They are one of many independent and separate networks within the autonomous system of the ISP. The ISP is responsible for the routing of packets within its autonomous system and between other autonomous systems.

The autonomous system number used for EIGRP configuration is only significant to the EIGRP routing domain. It functions as a process ID to help routers keep track of multiple running instances of EIGRP. This is required because it is possible to have more than one instance of EIGRP running on a network. Each instance of EIGRP can be configured to support and exchange routing updates for different networks.

3.5.1.3 The router eigrp Command

The Cisco IOS includes the processes to enable and configure several different types of dynamic routing protocols. The **router** global configuration mode command is used to begin the configuration of any dynamic routing protocol.

When followed by a question mark (**?**), the **router** global configuration mode command lists of all the available routing protocols supported by this specific IOS release running on the router.

The following global configuration mode command is used to enter the router configuration mode for EIGRP and begin the configuration of the EIGRP process:

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

The *autonomous-system* argument can be assigned to any 16-bit value between the number 1 and 65,535. All routers within the EIGRP routing domain must use the same autonomous system number.

Note: EIGRP and OSPF can support multiple instances of the routing protocol. However, this multiple routing protocol implementation is not usually needed or recommended.

The **router eigrp** *autonomous-system* command does not start the EIGRP process itself. The router does not start sending updates. Rather, this command only provides access to configure the EIGRP settings.

To completely remove the EIGRP routing process from a device, use the **no router eigrp** *autonomous-system* global configuration mode command, which stops the EIGRP process and removes all existing EIGRP router configurations.

3.5.1.4 EIGRP Router ID

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

The router ID is used in both EIGRP and OSPF routing protocols. However, the role of the router ID is more significant in OSPF. In EIGRP IPv4 implementations, the use of the router ID is not that apparent. EIGRP for IPv4 uses the 32-bit router ID to identify the originating router for redistribution of external routes. The need for a router ID becomes more evident in the discussion of EIGRP for IPv6. While the router ID is necessary for redistribution, the details of EIGRP redistribution are beyond the scope of this curriculum. For purposes of this curriculum, it is only necessary to understand what the router ID is and how it is determined.

To determine its router ID, a Cisco IOS router will use the following three criteria in order:
1. Use the address configured with the **eigrp router-id** *ipv4-address* router configuration mode command.

2. If the router ID is not configured, choose the highest IPv4 address of any of its loopback interfaces.

3. If no loopback interfaces are configured, choose the highest active IPv4 address of any of its physical interfaces.

If the network administrator does not explicitly configure a router ID using the **eigrp router-id** command, EIGRP generates its own router ID using either a loopback or physical IPv4 address. A loopback address is a virtual interface and is automatically in the up state when configured. The interface does not need to be enabled for EIGRP, meaning that it does not need to be included in one of the EIGRP **network** commands. However, the interface must be in the up/up state.

Note: The **eigrp router-id** command is used to configure the router ID for EIGRP. Some versions of IOS will accept the command **router-id**, without first specifying **eigrp**. The running-config, however, will display **eigrp router-id** regardless of which command is used.

3.5.1.5 Configuring the EIGRP Router ID

The **eigrp router-id** *ipv4-address* router configuration command is the preferred method used to configure the EIGRP router ID. This method takes precedence over any configured loopback or physical interface IPv4 addresses. The command syntax is:

Note: The IPv4 address used to indicate the router ID is actually any 32-bit number displayed in dotted-decimal notation.

The *ipv4-address* router ID can be configured with any IPv4 address except 0.0.0.0 and 255.255.255.255. The router ID should be a unique 32-bit number in the EIGRP routing domain; otherwise, routing inconsistencies can occur.

If a router ID is not explicitly configured, then the router would use its highest IPv4 address configured on a loopback interface. The advantage of using a loopback interface is that unlike physical interfaces, loopbacks cannot fail. There are no actual cables or adjacent devices on which the loopback interface depends for being in the up state. Therefore, using a loopback address for the router ID can provide a more consistent router ID than using an interface address.

If the **eigrp router-id** command is not used and loopback interfaces are configured, EIGRP chooses the highest IPv4 address of any of its loopback interfaces. The following commands are used to enable and configure a loopback interface:

```
Router(config)# interface loopback number
```

```
Router(config-if)# ip address ipv4-address subnet-mask
```

Verifying the EIGRP Process

The **show ip protocols** command displays the parameters and current state of any active routing protocol processes, including both EIGRP and OSPF. The **show ip protocols** command displays different types of output specific to each routing protocol.

3.5.1.6 The network Command

EIGRP router configuration mode allows for the configuration of the EIGRP routing protocol. To enable EIGRP routing on an interface, use the **network** *ipv4-network-address* router configuration mode command. The *ipv4-netword-address* is the classful network address for each directly connected network.

The **network** command has the same function as in all IGP routing protocols. The **network** command in EIGRP:

- Enables any interface on this router that matches the network address in the **network** router configuration mode command to send and receive EIGRP updates.
- The network of the interfaces is included in EIGRP routing updates.

DUAL automatically generates the notification message because the **eigrp log-neighbor-changes** router configuration mode command is enabled by default. Specifically, the command helps verify neighbor adjacencies during configuration of EIGRP and displays any changes in EIGRP neighbor adjacencies, such as when an EIGRP adjacency has been added or removed.

3.5.1.7 The network Command and Wildcard Mask

By default, when using the **network** command and an IPv4 network address, such as 172.16.0.0, all interfaces on the router that belong to that classful network address are enabled for EIGRP. However, there may be times when the network administrator does not want to include all interfaces within a network when enabling EIGRP. For example, assume that an administrator wants to enable EIGRP on R2, but only for the subnet 192.168.10.8 255.255.255.252, on the S0/0/1 interface.

To configure EIGRP to advertise specific subnets only, use the *wildcard-mask* option with the network command:

```
Router(config-router)# network network-address [wildcard-mask]
```

A wildcard mask is similar to the inverse of a subnet mask. In a subnet mask, binary 1s are significant while binary 0s are not. In a wildcard mask, binary 0s are significant, while binary 1s are not. For example, the inverse of subnet mask 255.255.255.252 is 0.0.0.3.

Calculating a wildcard mask may seem daunting at first but it's actually pretty easy to do. To calculate the inverse of the subnet mask, subtract the subnet mask from 255.255.255.255 as follows:

```
  255.255.255.255
- 255.255.255.252
  ---------------
   0.  0.  0.  3   Wildcard mask
```

Configuring a wildcard mask is the official command syntax of the EIGRP **network** command. However, the Cisco IOS versions also accepts a subnet mask to be used instead. Notice in the output of the **show running-config** command, the IOS converted the subnet mask command to its wildcard mask.

3.5.1.8 Passive Interface

As soon as a new interface is enabled within the EIGRP network, EIGRP attempts to form a neighbor adjacency with any neighboring routers to send and receive EIGRP updates.

At times it may be necessary, or advantageous, to include a directly connected network in the EIGRP routing update, but not allow any neighbor adjacencies off of that interface to form. The **passive-interface** command can be used to prevent the neighbor adjacencies. There are two primary reasons for enabling the **passive-interface** command:

- To suppress unnecessary update traffic, such as when an interface is a LAN interface, with no other routers connected
- To increase security controls, such as preventing unknown rogue routing devices from receiving EIGRP updates

The **passive-interface** router configuration mode command disables the transmission and receipt of EIGRP Hello packets on these interfaces.

```
Router(config)# router eigrp as-number
Router(config-router)# passive-interface interface-type interface-number
```

Without a neighbor adjacency, EIGRP cannot exchange routes with a neighbor. Therefore, the **passive-interface** command prevents the exchange of routes on the interface. Although EIGRP does not send or receive routing updates on an interface configured with the **passive-interface** command, it still includes the address of the interface in routing updates sent out of other non-passive interfaces.

Note: To configure all interfaces as passive, use the **passive-interface default** command. To disable an interface as passive, use the **no passive-interface** *interface-type interface-number* command.

An example of using the passive interface to increase security controls is when a network must connect to a third-party organization, for which the local administrator has no control, such as when connecting to an ISP network. In this case, the local network administrator would need to advertise the interface link through their own network, but would not want the third-party organization to receive or send routing updates to the local routing device, as this is a security risk.

Verifying the Passive Interface

To verify whether any interface on a router is configured as passive, use the **show ip protocols** privileged EXEC mode command.

3.5.2 Verify EIGRP with IPv4

3.5.2.1 Verifying EIGRP: Examining Neighbors

Before EIGRP can send or receive any updates, routers must establish adjacencies with their neighbors. EIGRP routers establish adjacencies with neighbor routers by exchanging EIGRP Hello packets.

Use the **show ip eigrp neighbors** command to view the neighbor table and verify that EIGRP has established an adjacency with its neighbors. For each router, you should be able to see the IPv4 address of the adjacent router and the interface that this router uses to reach that EIGRP neighbor. Using this topology, each router has two neighbors listed in the neighbor table.

The column headers in the **show ip eigrp neighbors** command output identify the following:

- **H** - Lists the neighbors in the order that they were learned.
- **Address** - IPv4 address of the neighbor.
- **Interface** - Local interface on which this Hello packet was received.
- **Hold** - Current hold time. When a Hello packet is received, this value is reset to the maximum hold time for that interface, and then counts down to zero. If zero is reached, the neighbor is considered down.
- **Uptime** - Amount of time since this neighbor was added to the neighbor table.

- **Smooth Round Trip Timer (SRTT)** and **Retransmission Timeout** (**RTO**) - Used by RTP to manage reliable EIGRP packets.
- **Queue Count** - Should always be zero. If more than zero, then EIGRP packets wait to be sent.
- **Sequence Number** - Used to track updates, queries, and reply packets.

The **show ip eigrp neighbors** command is very useful for verifying and troubleshooting EIGRP.

If a neighbor is not listed after adjacencies have been established with a router's neighbors, check the local interface to ensure it is activated with the **show ip interface brief** command. If the interface is active, try to **ping** the IPv4 address of the neighbor. If the **ping** fails, it means that the neighbor interface is down and must be activated. If the **ping** is successful and EIGRP still does not see the router as a neighbor, examine the following configurations:

- Are both routers configured with the same EIGRP autonomous system number?
- Is the directly connected network included in the EIGRP **network** statements?

3.5.2.2 Verifying EIGRP: show ip protocols Command

The **show ip protocols** command is useful to identify the parameters and other information about the current state of any active IPv4 routing protocol processes configured on the router. The **show ip protocols** command displays different types of output specific to each routing protocol.

Note: Prior to IOS 15, EIGRP automatic summarization was enabled by default.

The output from the **show ip protocols** command is useful in debugging routing operations. Information in the Routing Information Sources field can help identify a router suspected of delivering bad routing information. The field lists all the EIGRP routing sources the Cisco IOS software uses to build its IPv4 routing table. For each source, note the following:

- IPv4 address
- Administrative distance
- Time the last update was received from this source

EIGRP has a default AD of 90 for internal routes and 170 for routes imported from an external source, such as default routes. When compared to other IGPs, EIGRP is the most preferred by the Cisco IOS, because it has the lowest administrative distance. EIGRP has a third AD value of 5, for summary routes.

3.5.2.3 Verifying EIGRP: Examine the IPv4 routing table

Another way to verify that EIGRP and other functions of the router are configured properly is to examine the IPv4 routing tables with the **show ip route** command. As with any dynamic routing protocol, the network administrator must verify the information in the routing table to ensure that it is populated as expected, based on configurations entered. For this reason, it is important to have a good understanding of the routing protocol configuration commands, as well as the routing protocol operations and the processes used by the routing protocol to build the IP routing table.

Notice that the outputs used throughout this course are from Cisco IOS 15. Prior to IOS 15, EIGRP automatic summarization was enabled by default. The state of automatic summarization can make a difference in the information displayed in the IPv4 routing table. If a previous version of the IOS is used, automatic summarization can be disabled using the **no auto-summary** router configuration mode command:

```
Router(config-router)# no auto-summary
```

The **show ip route** command verifies that routes received by EIGRP neighbors are installed in the IPv4 routing table. The **show ip route** command displays the entire routing table, including remote networks learned dynamically, directly connected and static routes. For this reason, it is normally the first command

used to check for convergence. After routing is correctly configured on all routers, the **show ip route** command reflects that each router has a full routing table, with a route to each network in the topology.

3.6 EIGRP Operation

3.6.1 EIGRP Initial Route Discovery

3.6.1.1 EIGRP Neighbor Adjacency

The goal of any dynamic routing protocol is to learn about remote networks from other routers and to reach convergence in the routing domain. Before any EIGRP update packets can be exchanged between routers, EIGRP must first discover its neighbors. EIGRP neighbors are other routers running EIGRP on directly connected networks.

EIGRP uses Hello packets to establish and maintain neighbor adjacencies. For two EIGRP routers to become neighbors, several parameters between the two routers must match. For example, two EIGRP routers must use the same EIGRP metric parameters and both must be configured using the same autonomous system number.

Each EIGRP router maintains a neighbor table, which contains a list of routers on shared links that have an EIGRP adjacency with this router. The neighbor table is used to track the status of these EIGRP neighbors.

3.6.1.2 EIGRP Topology Table

EIGRP updates contain networks that are reachable from the router sending the update. As EIGRP updates are exchanged between neighbors, the receiving router adds these entries to its EIGRP topology table.

Each EIGRP router maintains a topology table for each routed protocol configured, such as IPv4 and IPv6. The topology table includes route entries for every destination that the router learns from its directly connected EIGRP neighbors.

3.6.2 EIGRP Metrics

3.6.2.1 EIGRP Composite Metric

By default, EIGRP uses the following values in its composite metric to calculate the preferred path to a network:

- **Bandwidth** - The slowest bandwidth among all of the outgoing interfaces, along the path from source to destination.
- **Delay** - The cumulative (sum) of all interface delay along the path (in tens of microseconds).

The following values can be used, but are not recommended, because they typically result in frequent recalculation of the topology table:

- **Reliability** - Represents the worst reliability between the source and destination, which is based on keepalives.
- **Load** - Represents the worst load on a link between the source and destination, which is computed based on the packet rate and the configured bandwidth of the interface.

Note: Although the MTU is included in the routing table updates, it is not a routing metric used by EIGRP.

The Composite Metric

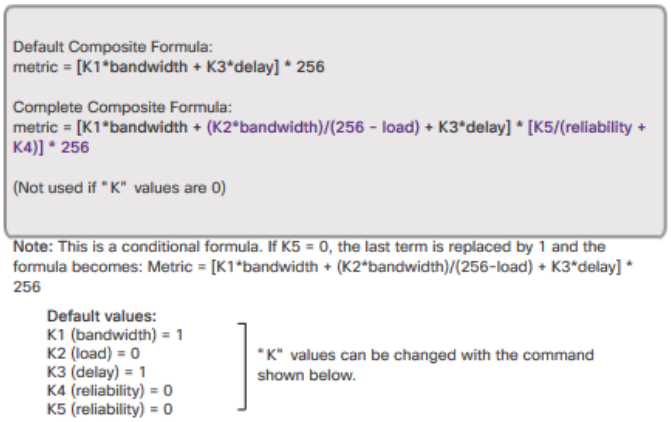


Figure above shows the composite metric formula used by EIGRP. The formula consists of values K1 to K5, known as EIGRP metric weights. K1 and K3 represent bandwidth and delay, respectively. K2 represents load, and K4 and K5 represent reliability. By default, K1 and K3 are set to 1, and K2, K4, and K5 are set to 0. The result is that only the bandwidth and delay values are used in the computation of the default composite metric. EIGRP for IPv4 and EIGRP for IPv6 use the same formula for the composite metric.

The metric calculation method (*k* values) and the EIGRP autonomous system number must match between EIGRP neighbors. If they do not match, the routers do not form an adjacency.

The default *k* values can be changed with the **metric weights** router configuration mode command:

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

Note: Modifying the **metric weights** value is generally not recommended and beyond the scope of this course. However, its relevance is important in establishing neighbor adjacencies. If one router has modified the metric weights and another router has not, an adjacency does not form.

Verifying the *k* Values

The **show ip protocols** command is used to verify the *k* values.

3.6.2.2 Examining Interface Metric Values

The **show interfaces** command displays interface information, including the parameters used to compute the EIGRP metric. The **show interfaces** command for the Serial 0/0/0 interface on Router.

- **BW** - Bandwidth of the interface (in kilobits per second).
- **DLY** - Delay of the interface (in microseconds).
- **Reliability** - Reliability of the interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes. By default, EIGRP does not include its value in computing its metric.
- **Txload, Rxload** - Transmit and receive load on the interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes. By default, EIGRP does not include its value in computing its metric.

Note: Throughout this course, bandwidth is referenced as kb/s. However, router output displays bandwidth using the Kbit/sec abbreviation. Router output also displays delay as usec. In this course, delay is referenced as microseconds.

3.6.2.3 Bandwidth Metric

The bandwidth metric is a static value used by some routing protocols, such as EIGRP and OSPF, to calculate their routing metric. The bandwidth is displayed in kilobits per second (kb/s).

On older routers, the serial link bandwidth metric defaults to 1544 kb/s. This is the bandwidth of a T1 connection. On newer routers, such as the Cisco 4321, serial link bandwidth defaults to the clock rate used on the link.

Note: These bandwidth values do not reflect the more common types of connections found in today's networks.

Always verify bandwidth with the **show interfaces** command. The default value of the bandwidth may or may not reflect the actual physical bandwidth of the interface. If actual bandwidth of the link differs from the default bandwidth value, the bandwidth value should be modified.

Configuring the Bandwidth Parameter

Because both EIGRP and OSPF use bandwidth in default metric calculations, a correct value for bandwidth is very important to the accuracy of routing information.

Use the following interface configuration mode command to modify the bandwidth metric:

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

Use the **no bandwidth** command to restore the default value.

Modify the bandwidth metric on both sides of the link to ensure proper routing in both directions.

Verifying the Bandwidth Parameter

Use the **show interfaces** command to verify the new bandwidth parameters.

Modifying the bandwidth value does not change the actual bandwidth of the link. The **bandwidth** command only modifies the bandwidth metric used by routing protocols, such as EIGRP and OSPF.

3.6.2.4 Delay Metric

Delay is the measure of the time it takes for a packet to traverse a route. The delay (DLY) metric is a static value based on the type of link to which the interface is connected and is expressed in microseconds. Delay is not measured dynamically. In other words, the router does not actually track how long packets take to reach the destination. The delay value, much like the bandwidth value, is a default value that can be changed by the network administrator.

When used to determine the EIGRP metric, delay is the cumulative (sum) of all interface delays along the path (measured in tens of microseconds).

The table in Figure below shows the default delay values for various interfaces. Notice that the default value is 20,000 microseconds for serial interfaces and 10 microseconds for GigabitEthernet interfaces.

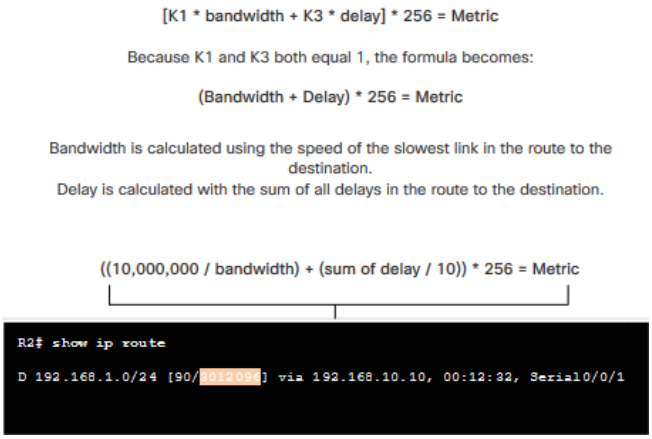
Media	Delay
Ethernet	1,000
Fast Ethernet	100
Gigabit Ethernet	10
16M Token Ring	630
FDDI	100
T1 (Serial Default)	20,000
DS0 (64 kb/s)	20,000
1024 kb/s	20,000
56 kb/s	20,000

Use the **show interfaces** command to verify the delay value on an interfac. Although an interface with various bandwidths can have the same delay value, by default, Cisco recommends not modifying the delay parameter, unless the network administrator has a specific reason to do so.

3.6.2.5 How to Calculate the EIGRP Metric

Although EIGRP automatically calculates the routing table metric used to choose the best path, it is important that the network administrator understands how these metrics were determined.

The figure shows the composite metric used by EIGRP. Using the default values for K1 and K3, the calculation can be simplified to the slowest bandwidth (or minimum bandwidth), plus the sum of all of the delays.



In other words, by examining the bandwidth and delay values for all of the outgoing interfaces of the route, we can determine the EIGRP metric as follows:

Step 1. Determine the link with the slowest bandwidth. Use that value to calculate bandwidth (10,000,000/bandwidth).

Step 2. Determine the delay value for each outgoing interface on the way to the destination. Add the delay values and divide by 10 (sum of delay/10).

Step 3. This composite metric produces a 24-bit value; however, EIGRP uses a 32-bit value. Multiplying the 24-bit value with 256 extends the composite metric into 32 bits. Therefore, add the computed values for bandwidth and delay, and multiply the sum by 256 to obtain the EIGRP metric.

The routing table output for R2 shows that the route to 192.168.1.0/24 has an EIGRP metric of 3,012,096.

3.6.2.6 Calculating the EIGRP Metric

Bandwidth

EIGRP uses the slowest bandwidth in its metric calculation. The slowest bandwidth can be determined by examining each interface between R2 and the destination network 192.168.1.0. The Serial 0/0/1 interface on R2 has a bandwidth of 1,024 kb/s. The GigabitEthernet 0/0 interface on R3 has a bandwidth of 1,000,000 kb/s. Therefore, the slowest bandwidth is 1,024 kb/s and is used in the calculation of the metric.

EIGRP divides a reference bandwidth value of 10,000,000 by the interface bandwidth value in kb/s. This results in higher bandwidth values receiving a lower metric and lower bandwidth values receiving a higher metric. 10,000,000 is divided by 1,024. If the result is not a whole number, then the value is rounded down. In this case, 10,000,000 divided by 1,024 equals 9,765.625. The .625 is dropped to yield 9,765 for the bandwidth portion of the composite metric.

Delay

EIGRP uses the sum of all delays to the destination. The Serial 0/0/1 interface on R2 has a delay of 20,000 microseconds. The Gigabit 0/0 interface on R3 has a delay of 10 microseconds. The sum of these delays is divided by 10. In the example, (20,000+10)/10 results in a value of 2,001 for the delay portion of the composite metric.

Calculate Metric

Use the calculated values for bandwidth and delay in the metric formula. This results in a metric of 3,012,096.

3.6.3 DUAL and the Topology Table

3.6.3.1 DUAL Concepts

EIGRP uses the Diffusing Update Algorithm (DUAL) to provide the best loop-free path and loop-free backup paths.

DUAL uses several terms, which are discussed in more detail throughout this section:

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

These terms and concepts are at the center of the loop avoidance mechanism of DUAL.

3.6.3.2 Introduction to DUAL

EIGRP uses the DUAL convergence algorithm. Convergence is critical to a network to avoid routing loops.

Routing loops, even temporary ones, can be detrimental to network performance. Distance vector routing protocols, such as RIP, prevent routing loops with hold-down timers and split horizon. Although EIGRP uses both of these techniques, it uses them somewhat differently; the primary way that EIGRP prevents routing loops is with the DUAL algorithm.

The DUAL algorithm is used to obtain loop-freedom at every instance throughout a route computation. This allows all routers involved in a topology change to synchronize at the same time. Routers that are not affected by the topology changes are not involved in the recomputation. This method provides EIGRP with faster convergence times than other distance vector routing protocols.

The decision process for all route computations is done by the DUAL Finite State Machine (FSM). An FSM is a workflow model, similar to a flow chart, which is composed of the following:

- A finite number of stages (states)
- Transitions between those stages
- Operations

The DUAL FSM tracks all routes and uses EIGRP metrics to select efficient, loop-free paths, and to identify the routes with the least-cost path to be inserted into the routing table.

Recomputation of the DUAL algorithm can be processor-intensive. EIGRP avoids recomputation whenever possible by maintaining a list of backup routes that DUAL has already determined to be loop-free. If the primary route in the routing table fails, the best backup route is immediately added to the routing table.

3.6.3.3 Successor and Feasible Distance

A successor is a neighboring router that is used for packet forwarding and is the least-cost route to the destination network. The IP address of a successor is shown in a routing table entry right after the word via.

FD is the lowest calculated metric to reach the destination network. FD is the metric listed in the routing table entry as the second number inside the brackets. As with other routing protocols, this is also known as the metric for the route.

3.6.3.4 Feasible Successors, Feasibility Condition, and Reported Distance

DUAL can converge quickly after a change in the topology because it can use backup paths to other networks without recomputing DUAL. These backup paths are known as Feasible Successors (FSs).

An FS is a neighbor that has a loop-free backup path to the same network as the successor, and it satisfies the Feasibility Condition (FC).

The FC is met when a neighbor's Reported Distance (RD) to a network is less than the local router's feasible distance to the same destination network. If the reported distance is less, it represents a loop-free path. The reported distance is simply an EIGRP neighbor's feasible distance to the same destination network. The reported distance is the metric that a router reports to a neighbor about its own cost to that network.

3.6.3.5 Topology Table: show ip eigrp topology Command

The EIGRP topology table contains all of the routes that are known to each EIGRP neighbor. As an EIGRP router learns routes from its neighbors, those routes are installed in its EIGRP topology table.

Use the **show ip eigrp topology** command to view the topology table. The topology table lists all successors and FSs that DUAL has calculated to destination networks. Only the successor is installed into the IP routing table.

3.6.4 DUAL and Convergence

3.6.4.1 DUAL Finite State Machine (FSM)

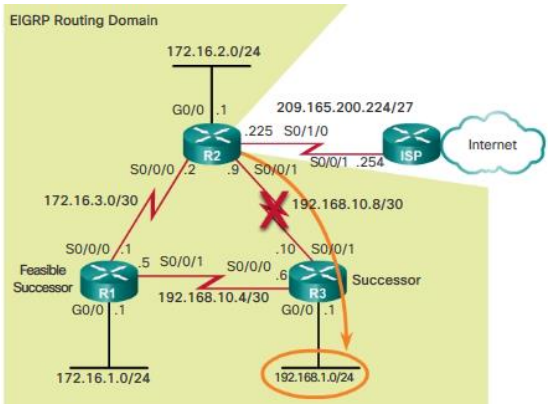
The centerpiece of EIGRP is DUAL and its EIGRP route-calculation engine. The actual name of this technology is DUAL Finite State Machine (FSM). This FSM contains all of the logic used to calculate and compare routes in an EIGRP network.

An FSM is an abstract machine, not a mechanical device with moving parts. FSMs define a set of possible states that something can go through, what events cause those states, and what events result from those states. Designers use FSMs to describe how a device, computer program, or routing algorithm reacts to a set of input events.

FSMs are beyond the scope of this course. However, the concept is used to examine some of the output from EIGRP's FSM using the **debug eigrp fsm** command. Use this command to examine what DUAL does when a route is removed from the routing table.

3.6.4.2 DUAL: Feasible Successor

R2 is currently using R3 as the successor to 192.168.1.0/24. In addition, R2 currently lists R1 as an FS, as shown in Figure below.



The **show ip eigrp topology** output for R2 verifies that R3 is the successor and R1 is the FS for the 192.168.1.0/24 network. To understand how DUAL can use a FS when the path using the successor is no longer available, a link failure is simulated between R2 and R3.

Before simulating the failure, DUAL debugging must be enabled using the **debug eigrp fsm** command on R2. A link failure is simulated using the **shutdown** command on the Serial 0/0/1 interface on R2.

The **debug** output displays the activity generated by DUAL when a link goes down. R2 must inform all EIGRP neighbors of the lost link, as well as update its own routing and topology tables. This example only shows selected **debug** output. In particular, notice that the DUAL FSM searches for and finds an FS for the route in the EIGRP topology table.

The FS R1 now becomes the successor and is installed in the routing table as the new best path to 192.168.1.0/24. With an FS, this change in the routing table happens almost immediately.

If the link between R2 and R3 is made active again, then R3 returns as the successor and R1 once again becomes the FS.

3.6.4.3 DUAL: No Feasible Successor

Occasionally, the path to the successor fails and there are no FSs. In this instance, DUAL does not have a guaranteed loop-free backup path to the network, so the path is not in the topology table as an FS. If there are no FSs in the topology table, DUAL puts the network into the active state. DUAL actively queries its neighbors for a new successor.

Before the link failure is simulated, DUAL debugging is enabled with the **debug eigrp fsm** command on R1. A link failure is simulated using the **shutdown** command on the Serial 0/0/1 interface on R1.

When the successor is no longer available and there is no feasible successor, DUAL puts the route into an active state. DUAL sends EIGRP queries asking other routers for a path to the network. Other routers return EIGRP replies, letting the sender of the EIGRP query know whether or not they have a path to the requested network. If none of the EIGRP replies have a path to this network, the sender of the query does not have a route to this network.

If the sender of the EIGRP queries receives EIGRP replies that include a path to the requested network, the preferred path is added as the new successor and added to the routing table. This process takes longer than if DUAL had an FS in its topology table and was able to quickly add the new route to the routing table.

3.7 Implement EIGRP for IPv6

3.7.1 EIGRP for IPv6

3.7.1.1 EIGRP for IPv6

Similar to its IPv4 counterpart, EIGRP for IPv6 exchanges routing information to populate the IPv6 routing table with remote prefixes. EIGRP for IPv6 was made available in Cisco IOS, Release 12.4(6)T.

Note: In IPv6, the network address is referred to as the prefix and the subnet mask is called the prefix length.

EIGRP for IPv4 runs over the IPv4 network layer, communicating with other EIGRP IPv4 peers, and advertising only IPv4 routes. EIGRP for IPv6 has the same functionality as EIGRP for IPv4, but uses IPv6 as the network layer transport, communicating with EIGRP for IPv6 peers and advertising IPv6 routes.

EIGRP for IPv6 also uses DUAL as the computation engine to guarantee loop-free paths and backup paths throughout the routing domain.

As with all IPv6 routing protocols, EIGRP for IPv6 has separate processes from its IPv4 counterpart. The processes and operations are basically the same as in the IPv4 routing protocol; however, they run independently. EIGRP for IPv4 and EIGRP for IPv6 each have separate EIGRP neighbor tables, EIGRP topology tables, and IP routing tables. EIGRP for IPv6 is a separate protocol-dependent module (PDM).

The EIGRP for IPv6 configuration and verification commands are very similar to those used in EIGRP for IPv4. These commands are described later in this section.

3.7.1.2 Compare EIGRP for IPv4 and IPv6

The following is a comparison of the main features of EIGRP for IPv4 and EIGRP for IPv6:

- **Advertised routes** - EIGRP for IPv4 advertises IPv4 networks; whereas, EIGRP for IPv6 advertises IPv6 prefixes.
- **Distance vector** - Both EIGRP for IPv4 and IPv6 are advanced distance vector routing protocols. Both protocols use the same administrative distances.
- **Convergence technology** - EIGRP for IPv4 and IPv6 both use the DUAL algorithm. Both protocols use the same DUAL techniques and processes, including successor, FS, FD, and RD.
- **Metric** - Both EIGRP for IPv4 and IPv6 use bandwidth, delay, reliability, and load for their composite metric. Both routing protocols use the same composite metric and use only bandwidth and delay, by default.
- **Transport protocol** - The Reliable Transport Protocol (RTP) is responsible for guaranteed delivery of EIGRP packets to all neighbors for both protocols, EIGRP for IPv4 and IPv6.
- **Update messages** - Both EIGRP for IPv4 and IPv6 send incremental updates when the state of a destination changes. The terms, partial and bounded, are used when referring to updates for both protocols.
- **Neighbor discovery mechanism** - EIGRP for IPv4 and EIGRP for IPv6 use a simple Hello mechanism to learn about neighboring routers and form adjacencies.
- **Source and destination addresses** - EIGRP for IPv4 sends messages to the multicast address 224.0.0.10. These messages use the source IPv4 address of the outbound interface. EIGRP for IPv6 sends its messages to the multicast address FF02::A. EIGRP for IPv6 messages are sourced using the IPv6 link-local address of the exit interface.
- **Authentication** - EIGRP for IPv4 and EIGRP for IPv6 use Message Digest 5 (MD5) authentication. Named EIGRP also supports the stronger SHA256 algorithm.
- **Router ID** - Both EIGRP for IPv4 and EIGRP for IPv6 use a 32-bit number for the EIGRP router ID. The 32-bit router ID is represented in dotted-decimal notation and is commonly referred to as an IPv4 address. If the EIGRP for IPv6 router has not been configured with an IPv4 address, the **eigrp router-id** command must be used to configure a 32-bit router ID. The process for determining the router ID is the same for both EIGRP for IPv4 and IPv6.

3.7.1.3 IPv6 Link-local Addresses

Routers running a dynamic routing protocol, such as EIGRP exchange messages between neighbors on the same subnet or link. Routers only need to send and receive routing protocol messages with their directly connected neighbors. These messages are always sent from the source IP address of the router that is doing the forwarding.

IPv6 link-local addresses are ideal for this purpose. An IPv6 link-local address enables a device to communicate with other IPv6-enabled devices on the same link and only on that link (subnet). Packets with a source or destination link-local address cannot be routed beyond the link from where the packet originated.

EIGRP for IPv6 messages are sent using:

- **Source IPv6 address** - This is the IPv6 link-local address of the exit interface.
- **Destination IPv6 address** - When the packet needs to be sent to a multicast address, it is sent to the IPv6 multicast address FF02::A, the all-EIGRP-routers with link-local scope. If the packet can be sent as a unicast address, it is sent to the link-local address of the neighboring router.

Note: IPv6 link-local addresses are in the FE80::/10 range. The /10 indicates that the first 10 bits are 1111 1110 10xx xxxx, which results in the first hextet having a range of 1111 1110 10**00 0000** (FE80) to 1111 1110 10**11 1111** (FEBF).

3.7.2 Configure EIGRP for IPv6

3.7.2.1 EIGRP for IPv6 Network Topology

If the network is running dual-stack, using both IPv4 and IPv6 on all devices, EIGRP for both IPv4 and IPv6 can be configured on all the routers. However, in this section, the focus is solely on EIGRP for IPv6.

Only the IPv6 global unicast addresses have been configured on each router.

Because EIGRP for IPv4 and IPv6 use the same metrics, modifying the bandwidth parameters influences both routing protocols.

3.7.2.2 Configuring IPv6 Link-local Addresses

Link-local addresses are automatically created when an IPv6 global unicast address is assigned to the interface. Global unicast addresses are not required on an interface.

Unless configured manually, Cisco routers create the link-local address using FE80::/10 prefix and the EUI-64 process. EUI-64 involves using the 48-bit Ethernet MAC address, inserting FFFE in the middle and flipping the seventh bit. For serial interfaces, Cisco uses the MAC address of an Ethernet interface. A router with several serial interfaces can assign the same link-local address to each IPv6 interface, because link-local addresses only need to be local on the link.

Link-local addresses created using the EUI-64 format, or in some cases random interface IDs, make it difficult to recognize and remember those addresses. Because IPv6 routing protocols use IPv6 link-local addresses for unicast addressing and next hop address information in the routing table, it is common practice to make it an easily recognizable address. Configuring the link-local address manually provides the ability to create an address that is recognizable and easier to remember.

Link-local addresses can be configured manually using the same interface configuration mode command used to create IPv6 global unicast addresses, but with different parameters:

Router(config-if)# **ipv6 address** *link-local-address* **link-local**

A link-local address has a prefix within the range FE80 to FEBF. When an address begins with this hextet (16-bit segment), the **link-local** keyword must follow the address.

3.7.2.3 Configuring the EIGRP for IPv6 Routing Process

The **ipv6 unicast-routing** global configuration mode command enables IPv6 routing on the router. This command is required before any IPv6 routing protocol can be configured. This command is not required to configure IPv6 addresses on the interfaces, but is necessary for the router to be enabled as an IPv6 router.

EIGRP for IPv6

The following global configuration mode command is used to enter router configuration mode for EIGRP for IPv6:

```
Router(config)# ipv6 router eigrp autonomous-system
```

Similar to EIGRP for IPv4, the *autonomous-system* value must be the same on all routers in the routing domain. The EIGRP for IPv6 routing process could not be configured until IPv6 routing was enabled with the **ipv6 unicast-routing** global configuration mode command.

Configuring the Router ID

The **eigrp router-id** command is used to configure the router ID. EIGRP for IPv6 uses a 32 bit value for the router ID. To obtain that value, EIGRP for IPv6 uses the same process as EIGRP for IPv4. The **eigrp router-id** command takes precedence over any loopback or physical interface IPv4 addresses. If an EIGRP for IPv6 router does not have any active interfaces with an IPv4 address, then the **eigrp router-id** command is required.

The router ID should be a unique 32-bit number in the EIGRP for IP routing domain; otherwise, routing inconsistencies can occur.

Note: The **eigrp router-id** command is used to configure the router ID for EIGRP. Some versions of IOS will accept the command **router-id**, without first specifying **eigrp**. The running-config, however, will display **eigrp router-id** regardless of which command is used.

By default, the EIGRP for IPv6 process is in a shutdown state. The **no shutdown** command is required to activate the EIGRP for IPv6 process. This command is not required for EIGRP for IPv4. Although, EIGRP for IPv6 is enabled, neighbor adjacencies and routing updates cannot be sent and received until EIGRP is activated on the appropriate interfaces.

Both the **no shutdown** command and a router ID are required for the router to form neighbor adjacencies.

3.7.2.4 The ipv6 eigrp Interface Command

EIGRP for IPv6 uses a different method to enable an interface for EIGRP. Instead of using the **network** router configuration mode command to specify matching interface addresses, EIGRP for IPv6 is configured directly on the interface.

Use the following interface configuration mode command to enable EIGRP for IPv6 on an interface:

```
Router(config-if)# ipv6 eigrp autonomous-system
```

The *autonomous-system* value must be the same as the autonomous system number used to enable the EIGRP routing process. Similar to the **network** command used in EIGRP for IPv4, the **ipv6 eigrp** interface command:

- Enables the interface to form adjacencies and send or receive EIGRP for IPv6 updates.
- Includes the prefix (network) of this interface in EIGRP for IPv6 routing updates.

Passive Interface with EIGRP for IPv6

The same **passive-interface** command used for IPv4 is used to configure an interface as passive with EIGRP for IPv6. The **show ipv6 protocols** command is used to verify the configuration.

3.7.3 Verifying EIGRP for IPv6

3.7.3.1 IPv6 Neighbor Table

Similar to EIGRP for IPv4, before any EIGRP for IPv6 updates can be sent or received, routers must establish adjacencies with their neighbors.

Use the **show ipv6 eigrp neighbors** command to view the neighbor table and verify that EIGRP for IPv6 has established an adjacency with its neighbors.

The column headers in the **show ipv6 eigrp neighbors** command output identify the following:

- **H** - Lists the neighbors in the order they were learned.
- **Address** - IPv6 link-local address of the neighbor.
- **Interface** - Local interface on which this Hello packet was received.
- **Hold** - Current hold time. When a Hello packet is received, this value is reset to the maximum hold time for that interface and then counts down to zero. If zero is reached, the neighbor is considered down.

- **Uptime** - Amount of time since this neighbor was added to the neighbor table.
- **SRTT** and **RTO** - Used by RTP to manage reliable EIGRP packets.
- **Queue Count** - Should always be zero. If it is more than zero, then EIGRP packets are waiting to be sent.
- **Sequence Number** - Used to track updates, queries, and reply packets.

The **show ipv6 eigrp neighbors** command is useful for verifying and troubleshooting EIGRP for IPv6. If an expected neighbor is not listed, ensure that both ends of the link are up/up using the **show ipv6 interface brief** command. The same requirements exist for establishing neighbor adjacencies with EIGRP for IPv6 as it does for IPv4. If both sides of the link have active interfaces, check to see:

- Are both routers configured with the same EIGRP autonomous system number?
- Is the interface enabled for EIGRP for IPv6 with the correct autonomous system number?

3.7.3.2 The show ip protocols Command

The **show ipv6 protocols** command displays the parameters and other information about the state of any active IPv6 routing protocol processes currently configured on the router. The **show ipv6 protocols** command displays different types of output specific to each IPv6 routing protocol.

The output from the **show ipv6 protocols** command is useful in debugging routing operations. The Interfaces section shows which interfaces EIGRP for IPv6 have been enabled. This is useful in verifying that EIGRP is enabled on all of the appropriate interfaces with the correct autonomous system number.

3.7.3.3 The EIGRP for IPv6 Routing Table

As with any routing protocol, the goal is to populate the IP routing table with routes to remote networks and the best paths to reaching those networks. As with IPv4, it is important to examine the IPv6 routing table and determine whether it is populated with the correct routes.

The IPv6 routing table is examined using the **show ipv6 route** command. EIGRP for IPv6 routes are denoted in the routing table with a D, as are EIGRP for IPv4 routes in the IPv4 routing table.

3.8 OSPF Characteristics

3.8.1 Open Shortest Path First

3.8.1.1 Evolution of OSPF

OSPF version 2 (OSPFv2) is available for IPv4 while OSPF version 3 (OSPFv3) is available for IPv6.

The initial development of OSPF began in 1987 by the Internet Engineering Task Force (IETF) OSPF Working Group. At that time, the Internet was largely an academic and research network funded by the U.S. government.

In 1989, the specification for OSPFv1 was published in RFC 1131. Two implementations were written. One implementation was developed to run on routers and the other to run on UNIX workstations. The latter implementation became a widespread UNIX process known as GATED. OSPFv1 was an experimental routing protocol and was never deployed.

In 1991, OSPFv2 was introduced in RFC 1247 by John Moy. OSPFv2 offered significant technical improvements over OSPFv1. It is classless by design; therefore, it supports VLSM and CIDR.

At the same time the OSPF was introduced, ISO was working on a link-state routing protocol of their own, Intermediate System-to-Intermediate System (IS-IS). IETF chose OSPF as their recommended Interior Gateway Protocol (IGP).

In 1998, the OSPFv2 specification was updated in RFC 2328, which remains the current RFC for OSPF.

In 1999, OSPFv3 for IPv6 was published in RFC 2740. OSPF for IPv6, created by John Moy, Rob Coltun, and Dennis Ferguson, is not only a new protocol implementation for IPv6, but also a major rewrite of the operation of the protocol.

In 2008, OSPFv3 was updated in RFC 5340 as OSPF for IPv6.

In 2010, the support of the Address Families (AF) feature in OSPFv3 was introduced with RFC 5838. The use of address families allows a routing protocol to support both IPv4 and IPv6 within a single unified configuration process. OSPFv3 with address families is beyond the scope of this curriculum.

Note: In this chapter, unless explicitly identified as OSPFv2 or OSPFv3, the term OSPF is used to indicate concepts that are shared by both.

3.8.1.2 Features of OSPF

OSPF features include:

- **Classless** - OSPFv2 is classless by design; therefore, it supports IPv4 VLSM and CIDR.
- **Efficient** - Routing changes trigger routing updates (no periodic updates). It uses the SPF algorithm to choose the best path.
- **Fast convergence** - It quickly propagates network changes.
- **Scalable** - It works well in small and large network sizes. Routers can be grouped into areas to support a hierarchical system.
- **Secure** - OSPFv2 supports Message Digest 5 (MD5) and Secure Hash Algorithm (SHA) authentication. OSPFv3 uses Internet Protocol Security (IPsec) to add authentication for OSPFv3 packets. When authentication is enabled, OSPF routers only accept encrypted routing updates from peers with the same pre-shared password.

Administrative distance (AD) is the trustworthiness (or preference) of the route source. OSPF has a default administrative distance of 110. OSPF has a lower number (making it a preferred routing protocol over IS-IS and RIP) on Cisco devices.

3.8.1.3 Components of OSPF

All routing protocols share similar components. They all use routing protocol messages to exchange route information. The messages help build data structures, which are then processed using a routing algorithm.

The three main components of the OSPF routing protocol include:

- **Data Structures**

OSPF creates and maintains three databases:

- **Adjacency database** - Creates the neighbor table.
- **Link-state database (LSDB)** - Creates the topology table.
- **Forwarding database** - Creates the routing table.

These tables contain a list of neighboring routers to exchange routing information with and are kept and maintained in RAM.

- **Routing Protocol Messages**

Layer 3 devices (such as routers) running OSPF exchange messages to convey routing information using five types of packets. These packets are:

- Hello packet
- Database description packet
- Link-state request packet
- Link-state update packet

- Link-state acknowledgment packet

These packets are used to discover neighboring routers and also to exchange routing information to maintain accurate information about the network.

- **Algorithm**

The router builds the topology table using results of calculations based on the Dijkstra SPF algorithm. The SPF algorithm is based on the cumulative cost to reach a destination.

The SPF algorithm creates an SPF tree by placing each router at the root of the tree and calculating the shortest path to each node. The SPF tree is then used to calculate the best routes. OSPF places the best routes into the forwarding database, which is used to make the routing table.

3.8.1.4 Link-State Operation

To maintain routing information, OSPF routers complete the following generic link-state routing process to reach a state of convergence:

1. Establish Neighbor Adjacencies - OSPF-enabled routers must recognize each other on the network before they can share information. An OSPF-enabled router sends Hello packets out all OSPF-enabled interfaces to determine if neighbors are present on those links. If a neighbor is present, the OSPF-enabled router attempts to establish a neighbor adjacency with that neighbor.

2. Exchange Link-State Advertisements - After adjacencies are established, routers then exchange link-state advertisements (LSAs). LSAs contain the state and cost of each directly connected link. Routers flood their LSAs to adjacent neighbors. Adjacent neighbors receiving the LSA immediately flood the LSA to other directly connected neighbors, until all routers in the area have all LSAs.

3. Build the Topology Table - After LSAs are received, OSPF-enabled routers build the topology table (LSDB) based on the received LSAs. This database eventually holds all the information about the topology of the network.

4. Execute the SPF Algorithm - Routers then execute the SPF algorithm. The gears in the figure are used to indicate the execution of the SPF algorithm. The SPF algorithm creates the SPF tree.

From the SPF tree, the best paths are offered to the IP routing table. The route will be inserted into the routing table unless there is a route source to the same network with a lower administrative distance, such as a static route. Routing decisions are made based on the entries in the routing table.

3.8.1.5 Single-Area and Multiarea OSPF

To make OSPF more efficient and scalable, OSPF supports hierarchical routing using areas. An OSPF area is a group of routers that share the same link-state information in their LSDBs.

OSPF can be implemented in one of two ways:

- **Single-Area OSPF** - all routers are in one area called the backbone area (area 0).
- **Multiarea OSPF** - OSPF is implemented using multiple areas, in a hierarchal fashion. All areas must connect to the backbone area (area 0). Routers interconnecting the areas are referred to as Area Border Routers (ABRs).

With multiarea OSPF, OSPF can divide one large routing domain into smaller areas, to support hierarchical routing. With hierarchical routing, routing still occurs between the areas (interarea routing), while many of the processor intensive routing operations, such as recalculating the database, are kept within an area.

For instance, any time a router receives new information about a topology change within the area (including the addition, deletion, or modification of a link) the router must rerun the SPF algorithm, create

a new SPF tree, and update the routing table. The SPF algorithm is CPU-intensive and the time it takes for calculation depends on the size of the area.

Note: Routers in other areas receive messages regarding topology changes, but these routers only update the routing table, not rerun the SPF algorithm.

Too many routers in one area would make the LSDBs very large and increase the load on the CPU. Therefore, arranging routers into areas effectively partitions a potentially large database into smaller and more manageable databases.

The hierarchical-topology design options with multiarea OSPF can offer these advantages:

- **Smaller routing tables** - Fewer routing table entries because network addresses can be summarized between areas. Route summarization is not enabled by default.
- **Reduced link-state update overhead** - Designing multiarea OSPF with smaller areas minimizes processing and memory requirements.
- **Reduced frequency of SPF calculations** - Localizes the impact of a topology change within an area. For instance, it minimizes routing update impact because LSA flooding stops at the area boundary.

3.8.2 OSPF Messages

3.8.2.1 Encapsulating OSPF Messages

OSPFv2 messages transmitted over an Ethernet link contain the following information:

- **Data Link Ethernet Frame Header** - Identifies the destination multicast MAC addresses 01-00-5E-00-00-05 or 01-00-5E-00-00-06 when encapsulating an OSPFv2 message.
- **IPv4 Packet Header** -Identifies the IP source address and destination address. The destination address is one of two OSPFv2 multicast addresses, 224.0.0.5 or 224.0.0.6. The header also contains a protocol field which will contain the code of 89 for OSPF.
- **OSPF Packet Header** - Identifies the OSPF packet type, the router ID and the area ID.
- **OSPF Packet Type Specific Data** - Contains the OSPF packet type information. The content differs depending on the packet type.

3.8.2.2 Types of OSPF Packets

OSPF uses link-state packets (LSPs) to establish and maintain neighbor adjacencies and exchange routing updates.

There are the five different types of LSPs used by OSPFv2. OSPFv3 has similar packet types. Each packet serves a specific purpose in the OSPF routing process:

- **Type 1: Hello packet** - Used to establish and maintain adjacency with other OSPF routers.
- **Type 2: Database Description (DBD) packet** - Contains an abbreviated list of the sending router's LSDB and is used by receiving routers to check against the local LSDB. The LSDB must be identical on all link-state routers within an area to construct an accurate SPF tree.
- **Type 3: Link-State Request (LSR) packet** - Receiving routers can then request more information about any entry in the DBD by sending an LSR.
- **Type 4: Link-State Update (LSU) packet** - Used to reply to LSRs and to announce new information. LSUs contain seven different types of LSAs.
- **Type 5: Link-State Acknowledgment (LSAck) packet** - When an LSU is received, the router sends an LSAck to confirm receipt of the LSU. The LSAck data field is empty.

3.8.2.3 Hello Packet

The OSPF Type 1 packet is the Hello packet. Hello packets are used to:

- Discover OSPF neighbors and establish neighbor adjacencies.

- Advertise parameters on which two routers must agree to become neighbors.
- Elect the Designated Router (DR) and Backup Designated Router (BDR) on multiaccess networks like Ethernet and Frame Relay. Point-to-point links do not require DR or BDR.

Important fields contained in the OSPFv2 Type 1 Hello packet include:

- **Type** - Identifies the type of packet. A one (1) indicates a Hello packet. A value 2 identifies a DBD packet, 3 an LSR packet, 4 an LSU packet, and 5 an LSAck packet.
- **Router ID** - A 32-bit value expressed in dotted decimal notation (like an IPv4 address) used to uniquely identify the originating router.
- **Area ID** – Number of the area from which the packet originated.
- **Network Mask** - Subnet mask associated with the sending interface.
- **Hello Interval** - Specifies the frequency, in seconds, at which a router sends Hello packets. The default Hello interval on multiaccess networks is 10 seconds. This timer must be the same on neighboring routers; otherwise, an adjacency is not established.
- **Router Priority** - Used in a DR/BDR election. The default priority for all OSPF routers is 1, but can be manually altered from 0 to 255. The higher the value, the more likely the router becomes the DR on the link.
- **Dead Interval** - Is the time in seconds that a router waits to hear from a neighbor before declaring the neighboring router out of service. By default, the router Dead Interval is four times the Hello interval. This timer must be the same on neighboring routers; otherwise, an adjacency is not established.
- **Designated Router (DR)** - Router ID of the DR.
- **Backup Designated Router (BDR)** - Router ID of the BDR.
- **List of Neighbors** - List that identifies the router IDs of all adjacent routers.

3.8.2.4 Hello Packet Intervals

OSPF Hello packets are transmitted to multicast address 224.0.0.5 in IPv4 and FF02::5 in IPv6 (all OSPF routers) every:

- 10 seconds (default on multiaccess and point-to-point networks)
- 30 seconds (default on non-broadcast multiple access [NBMA] networks; for example, Frame Relay)

The Dead interval is the period that the router waits to receive a Hello packet before declaring the neighbor down. If the Dead interval expires before the routers receive a Hello packet, OSPF removes that neighbor from its LSDB. The router floods the LSDB with information about the down neighbor out all OSPF-enabled interfaces.

Cisco uses a default of 4 times the Hello interval:

- 40 seconds (default on multiaccess and point-to-point networks)
- 120 seconds (default on NBMA networks; for example, Frame Relay)

3.8.2.5 Link-State Updates

Routers initially exchange Type 2 DBD packets, which is an abbreviated list of the sending router's LSDB and is used by receiving routers to check against the local LSDB.

A Type 3 LSR packet is used by the receiving routers to request more information about an entry in the DBD.

The Type 4 LSU packet is used to reply to an LSR packet.

A Type 5 packet is used to acknowledge the receipt of a Type 4 LSU.

LSUs are also used to forward OSPF routing updates, such as link changes. Specifically, an LSU packet can contain 11 different types of OSPFv2 LSAs. OSPFv3 renamed several of these LSAs and also contains two additional LSAs.

Note: The difference between the LSU and LSA terms can sometimes be confusing because these terms are often used interchangeably. However, an LSU contains one or more LSAs.

3.8.3 OSPF Operation

3.8.3.1 OSPF Operational States

When an OSPF router is initially connected to a network, it attempts to:

- Create adjacencies with neighbors
- Exchange routing information
- Calculate the best routes
- Reach convergence

OSPF progresses through several states while attempting to reach convergence:

- Down state
- Init state
- Two-Way state
- ExStart state
- Exchange state
- Loading state
- Full state

3.8.3.2 Establish Neighbor Adjacencies

When OSPF is enabled on an interface, the router must determine if there is another OSPF neighbor on the link. To accomplish this, the router forwards a Hello packet that contains its router ID out all OSPF-enabled interfaces. The OSPF router ID is used by the OSPF process to uniquely identify each router in the OSPF area. A router ID is a 32-bit number formatted like an IP address and assigned to uniquely identify a router among OSPF peers.

When a neighboring OSPF-enabled router receives a Hello packet with a router ID that is not within its neighbor list, the receiving router attempts to establish an adjacency with the initiating router.

The action performed in Two-Way state depends on the type of inter-connection between the adjacent routers:

- If the two adjacent neighbors are interconnected over a point-to-point link, then they immediately transition from the Two-Way state to the database synchronization phase.
- If the routers are interconnected over a common Ethernet network, then a designated router DR and a BDR must be elected.

Hello packets are continually exchanged to maintain router information.

3.8.3.3 OSPF DR and BDR

Why is a DR and BDR election necessary?

Multiaccess networks can create two challenges for OSPF regarding the flooding of LSAs:

- **Creation of multiple adjacencies** - Ethernet networks could potentially interconnect many OSPF routers over a common link. Creating adjacencies with every router is unnecessary and undesirable. It would lead to an excessive number of LSAs exchanged between routers on the same network.

- **Extensive flooding of LSAs** - Link-state routers flood their LSAs any time OSPF is initialized, or when there is a change in the topology. This flooding can become excessive.

To understand the problem with multiple adjacencies, we must study a formula:

For any number of routers (designated as *n*) on a multiaccess network, there are $n\ (n-1)\ /\ 2$ adjacencies.

The solution to managing the number of adjacencies and the flooding of LSAs on a multiaccess network is the DR. On multiaccess networks, OSPF elects a DR to be the collection and distribution point for LSAs sent and received. A BDR is also elected in case the DR fails. All other routers become DROTHERs. A DROTHER is a router that is neither the DR nor the BDR.

Note: The DR is only used for the dissemination of LSAs. The router will still use the best next-hop router indicated in the routing table for the forwarding of all other packets.

3.8.3.4 Synchronizing OSPF Databases

After the Two-Way state, routers transition to database synchronization states. While the Hello packet was used to establish neighbor adjacencies, the other four types of OSPF packets are used during the process of exchanging and synchronizing LSDBs.

In the ExStart state, the two routers decide which router will send the DBD packets first. The router with the higher router ID will be the first router to send DBD packets during the Exchange state.

In the Exchange state, the two routers exchange one or more DBD packets. A DBD packet includes information about the LSA entry header that appears in the router's LSDB. The entries can be about a link or about a network. Each LSA entry header includes information about the link-state type, the address of the advertising router, the link's cost, and the sequence number. The router uses the sequence number to determine the newness of the received link-state information.

After all LSRs have been satisfied for a given router, the adjacent routers are considered synchronized and in a full state.

As long as the neighboring routers continue receiving Hello packets, the network in the transmitted LSAs remain in the topology database. After the topological databases are synchronized, updates (LSUs) are sent only to neighbors:

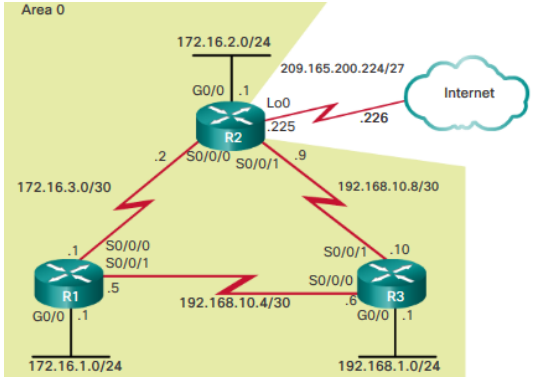
- When a change is perceived (incremental updates)
- Every 30 minutes

3.9 Single-Area OSPFv2

3.9.1 OSPF Router ID

3.9.1.1 OSPF Network Topology

Introduced in 1991, OSPFv2 is a link-state routing protocol for IPv4. OSPF was designed as an alternative to another IPv4 routing protocol, RIP.



The figure shows the topology used for configuring OSPFv2 in this section. The types of serial interfaces and their associated bandwidths may not necessarily reflect the more common types of connections found

in networks today. The bandwidths of the serial links used in this topology were chosen to help explain the calculation of the routing protocol metrics and the process of best path selection.

The routers in the topology have a starting configuration, including interface addresses. There is currently no static routing or dynamic routing configured on any of the routers. All interfaces on routers R1, R2, and R3 (except the loopback on R2) are within the OSPF backbone area. The ISP router is used as the routing domain's gateway to the Internet.

Note: In this topology the loopback interface is used to simulate the WAN link to the Internet.

3.9.1.2 Router OSPF Configuration Mode

Figure above is the reference topology for this topic. OSPFv2 is enabled using the **router ospf** *process-id* global configuration mode command. The *process-id* value represents a number between 1 and 65,535 and is selected by the network administrator. The *process-id* value is locally significant, which means that it does not have to be the same value on the other OSPF routers to establish adjacencies with those neighbors.

3.9.1.3 Router IDs

Every router requires a router ID to participate in an OSPF domain. The router ID can be defined by an administrator or automatically assigned by the router. The router ID is used by the OSPF-enabled router to:

- **Uniquely identify the router** - The router ID is used by other routers to uniquely identify each router within the OSPF domain and all packets that originate from them.
- **Participate in the election of the DR** - In a multiaccess LAN environment, the election of the DR occurs during initial establishment of the OSPF network. When OSPF links become active, the routing device configured with the highest priority is elected the DR. Assuming there is no priority configured, or there is a tie, then the router with the highest router ID is elected the DR. The routing device with the second highest router ID is elected the BDR.

But how does the router determine the router ID? Cisco routers derive the router ID based on one of three criteria, in the following preferential order:

- The router ID is explicitly configured using the OSPF **router-id** *rid* router configuration mode command. The *rid* value is any 32-bit value expressed as an IPv4 address. This is the recommended method to assign a router ID.
- If the router ID is not explicitly configured, the router chooses the highest IPv4 address of any of configured loopback interfaces. This is the next best alternative to assigning a router ID.
- If no loopback interfaces are configured, then the router chooses the highest active IPv4 address of any of its physical interfaces. This is the least recommended method because it makes it more difficult for administrators to distinguish between specific routers.

If the router uses the highest IPv4 address for the router ID, the interface does not need to be OSPF-enabled. This means that the interface address does not need to be included in one of the OSPF *network* commands for the router to use that IPv4 address as the router ID. The only requirement is that the interface is active and in the up state.

Note: The router ID looks like an IPv4 address, but it is not routable and, therefore, is not included in the routing table, unless the OSPF routing process chooses an interface (physical or loopback) that is appropriately defined by a **network** command.

3.9.1.4 Configuring an OSPF Router ID

Use the **router-id** *rid* router configuration mode command to manually assign a 32-bit value expressed as an IPv4 address to a router. An OSPF router identifies itself to other routers using this router ID.

R1 is configured with a router ID of 1.1.1.1, R2 with 2.2.2.2, and R3 with 3.3.3.3.

Use the **show ip protocols** command to verify the router ID.

If the router ID is the same on two neighboring routers, the router displays an error message similar to the one below:

%OSPF-4-DUP_RTRID1: Detected router with duplicate router ID.

To correct this problem, configure all routers so that they have unique OSPF router IDs.

3.9.1.5 Modifying a Router ID

Sometimes a router ID needs to be changed, for example, when a network administrator establishes a new router ID scheme for the network. However, after a router selects a router ID, an active OSPFv2 router does not allow the router ID to be changed until the router is reloaded or the OSPFv2 process cleared.

Note that the current router ID is 192.168.10.5. The router ID should be 1.1.1.1.

The router ID 1.1.1.1 is being assigned to R1. Notice how an informational message appears stating that the OSPFv2 process must be cleared or that the router must be reloaded. The reason is because R1 already has adjacencies with other neighbors using the router ID 192.168.10.5. Those adjacencies must be renegotiated using the new router ID 1.1.1.1.

Clearing the OSPF process is the preferred method to reset the router ID.

The OSPFv2 routing process is cleared using the **clear ip ospf process** privileged EXEC mode command. This forces OSPFv2 on R1 to transition to the Down and Init states. Notice the adjacency change messages from full to down and then from loading to full. The **show ip protocols** command verifies that the router ID has changed.

3.9.1.6 Using a Loopback Interface as the Router ID

A router ID can also be assigned using a loopback interface.

The IPv4 address of the loopback interface should be configured using a 32-bit subnet mask (255.255.255.255). This effectively creates a host route. A 32-bit host route does not get advertised as a route to other OSPF routers.

```
R1(config)# interface loopback 0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# end
R1#
```

The example in the figure displays how to configure a loopback interface with a host route on R1. R1 uses the host route as its router ID, assuming there is no router ID explicitly configured or previously learned.

Note: The **router-id** command is the preferred method. However, some older versions of the IOS do not recognize the **router-id** command; therefore, the best way to set the router ID on those routers is by using a loopback interface.

3.9.2 Configure Single-Area OSPFv2

3.9.2.1 Enabling OSPF on Interfaces

The **network** command determines which interfaces participate in the routing process for an OSPFv2 area. Any interfaces on a router that match the network address in the **network** command are enabled to send and receive OSPF packets. The **network** command also indicates the network (or subnet) address for the interface is included in OSPF routing updates.

The basic command syntax is **network** *network-address wildcard-mask* **area** *area-id*.

The **area** *area-id* syntax refers to the OSPF area. When configuring single-area OSPFv2, the **network** command must be configured with the same *area-id* value on all routers. Although any area ID can be

used, it is good practice to use an area ID of 0 with single-area OSPFv2. This convention makes it easier if the network is later altered to support multiarea OSPFv2.

3.9.2.2 Wildcard Mask

OSPFv2 uses the argument combination of *network-address wildcard-mask* to enable OSPF on interfaces. OSPF is classless by design; therefore, the wildcard mask is always required. When identifying interfaces that are participating in a routing process, the wildcard mask is typically the inverse of the subnet mask configured on that interface.

A wildcard mask is a string of 32 binary digits used by the router to determine which bits of the address to examine for a match. In a subnet mask, binary 1 is equal to a match and binary 0 is not a match. In a wildcard mask, the reverse is true:

- **Wildcard mask bit 0** - Matches the corresponding bit value in the address.
- **Wildcard mask bit 1** - Ignores the corresponding bit value in the address.

The easiest method for calculating a wildcard mask is to subtract the network subnet mask from 255.255.255.255.

The example calculates the wildcard mask from the network address of 192.168.10.0/24. To do so, the subnet mask 255.255.255.0 is subtracted from 255.255.255.255, providing a result of 0.0.0.255. Therefore, 192.168.10.0/24 is 192.168.10.0 with a wildcard mask of 0.0.0.255.

The example calculates the wildcard mask from the network address of 192.168.10.64/26. Again, the subnet mask 255.255.255.192 is subtracted from 255.255.255.255 providing a result of 0.0.0.63. Therefore, 192.168.10.0/26 is 192.168.10.0 with a wildcard mask of 0.0.0.63.

3.9.2.3 The network Command

There are several ways to identify the interfaces that will participate in the OSPFv2 routing process.

```
R1(config)# router ospf 10
R1(config-router)# network 172.16.1.0 0.0.0.255 area 0
R1(config-router)# network 172.16.3.0 0.0.0.3 area 0
R1(config-router)# network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#
```

Figure above displays the required commands to determine which interfaces on R1 participate in the OSPFv2 routing process for an area. Notice the use of wildcard masks to identify the respective interfaces based on their network addresses. Because this is a single-area OSPF network, all area IDs are set to 0.

As an alternative, OSPFv2 can be enabled using the **network** *intf-ip-address* **0.0.0.0** ***area*** *area-id* router configuration mode command.

```
R1(config)# router ospf 10
R1(config-router)# network 172.16.1.1 0.0.0.0 area 0
R1(config-router)# network 172.16.3.1 0.0.0.0 area 0
R1(config-router)# network 192.168.10.5 0.0.0.0 area 0
R1(config-router)#
```

Figure above provides an example of specifying the interface IPv4 address with a quad 0 wildcard mask. Entering **network 172.16.3.1 0.0.0.0 area 0** on R1 tells the router to enable interface Serial0/0/0 for the routing process. As a result, the OSPFv2 process will advertise the network that is on this interface (172.16.3.0/30).

The advantage of specifying the interface is that the wildcard mask calculation is not necessary. OSPFv2 uses the interface address and subnet mask to determine the network to advertise.

Some IOS versions allow the subnet mask to be entered instead of the wildcard mask. The IOS then converts the subnet mask to the wildcard mask format.

Note: While completing the syntax checker, observe the informational messages describing the adjacency between R1 (1.1.1.1) and R2 (2.2.2.2). The IPv4 addressing scheme used for the router ID makes it easy to identify the neighbor.

3.9.2.4 Passive Interface

By default, OSPF messages are forwarded out all OSPF-enabled interfaces. However, these messages really only need to be sent out interfaces connecting to other OSPF-enabled routers.

Refer to the topology in the figure. OSPFv2 messages are forwarded out of all three routers G0/0 interface even though no OSPFv2 neighbor exists on that LAN. Sending out unneeded messages on a LAN affects the network in three ways:

- **Inefficient Use of Bandwidth** - Available bandwidth is consumed transporting unnecessary messages. Messages are multicasted; therefore, switches are also forwarding the messages out all ports.
- **Inefficient Use of Resources** - All devices on the LAN must process the message and eventually discard the message.
- **Increased Security Risk** - Advertising updates on a broadcast network is a security risk. OSPF messages can be intercepted with packet sniffing software. Routing updates can be modified and sent back to the router, corrupting the routing table with false metrics that misdirect traffic.

3.9.2.5 Configuring Passive Interfaces

```
R1(config)# router ospf 10
R1(config-router)# passive-interface GigabitEthernet 0/0
R1(config-router)# end
```

Use the **passive-interface** router configuration mode command to prevent the transmission of routing messages through a router interface, but still allow that network to be advertised to other routers, as shown in Figure above. Specifically, the command stops routing messages from being sent out the specified interface. However, the network that the specified interface belongs to is still advertised in routing messages that are sent out other interfaces.

For instance, there is no need for R1, R2, and R3 to forward OSPF messages out of their LAN interfaces. The configuration identifies the R1 G0/0 interface as passive.

It is important to know that a neighbor adjacency cannot be formed over a passive interface. This is because link-state packets cannot be sent or acknowledged.

The **show ip protocols** command is then used to verify that the Gigabit Ethernet interface was passive. **Note**: OSPFv2 and OSPFv3 both support the **passive-interface** command.

3.9.3 OSPF Cost

3.9.3.1 OSPF Metric = Cost

Recall that a routing protocol uses a metric to determine the best path of a packet across a network. A metric gives indication of the overhead that is required to send packets across a certain interface. OSPF uses cost as a metric. A lower cost indicates a better path than a higher cost.

The cost of an interface is inversely proportional to the bandwidth of the interface. Therefore, a higher bandwidth indicates a lower cost. More overhead and time delays equal a higher cost. Therefore, a 10-Mb/s Ethernet line has a higher cost than a 100-Mb/s Ethernet line.

The formula used to calculate the OSPF cost is:

- **Cost** = *reference bandwidth* / *interface bandwidth*

The default reference bandwidth is 10^8 (100,000,000); therefore, the formula is:

- **Cost** = 100,000,000 bps / *interface bandwidth in bps*

3.9.3.2 Adjusting the Reference Bandwidth

OSPF uses a reference bandwidth of 100 Mb/s for any links that are equal to or faster than a fast Ethernet connection. Therefore, the cost assigned to a fast Ethernet interface with an interface bandwidth of 100 Mb/s would equal 1.

Cost = *100,000,000 bps* / *100,000,000* = *1*

While this calculation works for fast Ethernet interfaces, it is problematic for links that are faster than 100 Mb/s; because the OSPF metric only uses integers as its final cost of a link. If something less than an integer is calculated, OSPF rounds up to the nearest integer. For this reason, from the OSPF perspective, an interface with an interface bandwidth of 100 Mb/s (a cost of 1) has the same cost as an interface with a bandwidth of 100 Gb/s (a cost of 1).

To assist OSPF in making the correct path determination, the reference bandwidth must be changed to a higher value to accommodate networks with links faster than 100 Mb/s.

Adjusting the Reference Bandwidth

Changing the reference bandwidth does not actually affect the bandwidth capacity on the link; rather, it simply affects the calculation used to determine the metric. To adjust the reference bandwidth, use the **auto-cost reference-bandwidth** *Mb/s* router configuration command. This command must be configured on every router in the OSPF domain. Notice that the value is expressed in Mb/s; therefore, to adjust the costs for:

- Gigabit Ethernet -auto-cost reference-bandwidth 1000
- 10 Gigabit Ethernet -auto-cost reference-bandwidth 10000

To return to the default reference bandwidth, use the **auto-cost reference-bandwidth 100** command.

The table in Figure below displays the OSPF cost if the reference bandwidth is set to Gigabit Ethernet. Although the metric values increase, OSPF makes better choices because it can now distinguish between FastEthernet and Gigabit Ethernet links.

Interface Type	Reference Bandwidth in bps		Default Bandwidth in bps	Cost
10 Gigabit Ethernet 10 Gbps	1,000,000,000	÷	10,000,000,000	1
Gigabit Ethernet 1 Gbps	1,000,000,000	÷	1,000,000,000	1
Fast Ethernet 100 Mbps	1,000,000,000	÷	100,000,000	10
Ethernet 10 Mbps	1,000,000,000	÷	10,000,000	100
Serial 1.544 Mbps	1,000,000,000	÷	1,544,000	647
Serial 128 kbps	1,000,000,000	÷	128,000	7812
Serial 64 kbps	1,000,000,000	÷	64,000	15625

Note: The costs represent whole numbers that have been rounded down.

3.9.3.4 Default Interface Bandwidths

All interfaces have default bandwidth values assigned to them. As with reference bandwidth, interface bandwidth values do not actually affect the speed or capacity of the link. Instead, they are used by some routing protocols, like OSPF, to compute the routing metric. Therefore, it is important that the bandwidth value reflect the actual speed of the link so that the routing table has accurate best path information.

Although the bandwidth values of Ethernet interfaces usually match the link speed, some other interfaces may not. For instance, the actual speed of serial interfaces is often different than the default bandwidth. On Cisco routers, the default bandwidth on most serial interfaces is set to 1.544 Mb/s.

Note: Older serial interfaces may default to 128 kb/s.

Use the **show interfaces** command to view the interface bandwidth setting. The bandwidth setting is accurate and therefore the serial interface does not have to be adjusted.

3.9.3.5 Adjusting the Interface Bandwidth
Adjusting the Interface Bandwidth

To adjust the interface bandwidth use the bandwidth *kilobits* interface configuration command. Use the no bandwidth command to restore the default value.

```
R1(config)# int s0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# end
R1#
*Mar 27 10:10:07.735: %SYS-5-CONFIG_I: Configured from console by c
R1#
R1#  show interfaces serial 0/0/1 | include BW
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
R1#
R1#  show ip ospf interface serial 0/0/1 | include Cost:
  Process ID 10, Router ID 1.1.1.1, Network Type
  POINT_TO_POINT, Cost: 15625
```

The example in Figure above adjusts the R1 Serial 0/0/1 interface bandwidth to 64 kb/s. A quick verification confirms that the interface bandwidth setting is now 64 kb/s.

The bandwidth must be adjusted at each end of the serial links, therefore:

- R2 requires its S0/0/1 interface to be adjusted to 1,024 kb/s.
- R3 requires its serial 0/0/0 to be adjusted to 64 kb/s and its serial 0/0/1 to be adjusted to 1,024 kb/s.

Note: A common misconception for students who are new to networking and the Cisco IOS is to assume that the **bandwidth** command changes the physical bandwidth of the link. The command only modifies the bandwidth metric used by EIGRP and OSPF. The command does not modify the actual bandwidth on the link.

3.9.3.6 Manually Setting the OSPF Cost

As an alternative to setting the default interface bandwidth, the cost can be manually configured on an interface using the **ip ospf cost** *value* interface configuration command.

An advantage of configuring a cost over setting the interface bandwidth is that the router does not have to calculate the metric when the cost is manually configured. In contrast, when the interface bandwidth is configured, the router must calculate the OSPF cost based on the bandwidth. The **ip ospf cost** command is useful in multi-vendor environments where non-Cisco routers may use a metric other than bandwidth to calculate the OSPFv2 costs.

Both the **bandwidth** interface command and the **ip ospf cost** interface command achieve the same result, which is to provide an accurate value for use by OSPFv2 in determining the best route.

3.9.4 Verify OSPF

3.9.4.1 Verify OSPF Neighbors

Use the **show ip ospf neighbor** command to verify that the router has formed an adjacency with its neighboring routers. If the router ID of the neighboring router is not displayed, or if it does not show as being in a state of FULL, the two routers have not formed an OSPFv2 adjacency.

If two routers do not establish adjacency, link-state information is not exchanged. Incomplete LSDBs can cause inaccurate SPF trees and routing tables. Routes to destination networks may not exist, or may not be the most optimum path.

For each neighbor, this command displays the following output:

- **Neighbor ID** - The router ID of the neighboring router.
- **Pri** - The OSPFv2 priority of the interface. This value is used in the DR and BDR election.
- **State** - The OSPFv2 state of the interface. FULL state means that the router and its neighbor have identical OSPFv2 LSDBs. On multiaccess networks, such as Ethernet, two routers that are adjacent may have their states displayed as 2WAY. The dash indicates that no DR or BDR is required because of the network type.
- **Dead Time** - The amount of time remaining that the router waits to receive an OSPFv2 Hello packet from the neighbor before declaring the neighbor down. This value is reset when the interface receives a Hello packet.
- **Address** - The IPv4 address of the neighbor's interface to which this router is directly connected.
- **Interface** - The interface on which this router has formed adjacency with the neighbor.

Two routers may not form an OSPFv2 adjacency if:

- The subnet masks do not match, causing the routers to be on separate networks.
- OSPFv2 Hello or Dead Timers do not match.
- OSPFv2 Network Types do not match.
- There is a missing or incorrect OSPFv2 **network** command.

3.9.4.2 Verify OSPF Protocol Settings

The **show ip protocols** command is a quick way to verify vital OSPF configuration information. This includes the OSPFv2 process ID, the router ID, networks the router is advertising, the neighbors the router is receiving updates from, and the default administrative distance, which is 110 for OSPF.

3.9.4.3 Verify OSPF Process Information

The **show ip ospf** command can also be used to examine the OSPFv2 process ID and router ID. This command displays the OSPFv2 area information and the last time the SPF algorithm was calculated.

3.9.4.4 Verify OSPF Interface Settings

The quickest way to verify OSPFv2 interface settings is to use the **show ip ospf interface** command. This command provides a detailed list for every OSPFv2-enabled interface. The command is useful to determine whether the **network** statements were correctly composed.

To get a summary of OSPFv2-enabled interfaces, use the **show ip ospf interface brief** command.

3.10 Single-Area OSPFv3

3.10.1 OSPFv2 vs OSPFv3

3.10.1.1 OSPFv3

OSPFv3 is the OSPFv2 equivalent for exchanging IPv6 prefixes. Recall that in IPv6, the network address is referred to as the prefix and the subnet mask is called the prefix-length.

Similar to its IPv4 counterpart, OSPFv3 exchanges routing information to populate the IPv6 routing table with remote prefixes.

Note: With the OSPFv3 Address Families feature, OSPFv3 includes support for both IPv4 and IPv6. OSPF Address Families is beyond the scope of this curriculum.

OSPFv2 runs over the IPv4 network layer, communicating with other OSPF IPv4 peers, and advertising only IPv4 routes.

OSPFv3 has the same functionality as OSPFv2, but uses IPv6 as the network layer transport, communicating with OSPFv3 peers and advertising IPv6 routes. OSPFv3 also uses the SPF algorithm as the computation engine to determine the best paths throughout the routing domain.

As with all IPv6 routing protocols, OSPFv3 has separate processes from its IPv4 counterpart. The processes and operations are basically the same as in the IPv4 routing protocol, but run independently. OSPFv2 and OSPFv3 each have separate adjacency tables, OSPF topology tables, and IP routing tables.

The OSPFv3 configuration and verification commands are similar those used in OSPFv2.

3.10.1.2 Similarities Between OSPFv2 to OSPFv3

The following are similarities between OSPFv2 and OSPFv3:

- **Link-state** - OSPFv2 and OSPFv3 are both classless link-state routing protocols.
- **Routing algorithm** - OSPFv2 and OSPFv3 use the SPF algorithm to make routing decisions.
- **Metric** - The RFCs for both OSPFv2 and OSPFv3 define the metric as the cost of sending packets out the interface. OSPFv2 and OSPFv3 can be modified using the **auto-cost reference-bandwidth** *ref-bw* router configuration mode command. The command only influences the OSPF metric where it was configured. For example, if this command was entered for OSPFv3, it does not affect the OSPFv2 routing metrics.
- **Areas** - The concept of multiple areas in OSPFv3 is the same as in OSPFv2. Multiareas that minimize link-state flooding and provide better stability with the OSPF domain.
- **OSPF packet types** - OSPFv3 uses the same five basic packet types as OSPFv2 (Hello, DBD, LSR, LSU, and LSAck).
- **Neighbor discovery mechanism** - The neighbor state machine, including the list of OSPF neighbor states and events, remains unchanged. OSPFv2 and OSPFv3 use the Hello mechanism to learn about neighboring routers and form adjacencies. However, in OSPFv3, there is no requirement for matching subnets to form neighbor adjacencies. This is because neighbor adjacencies are formed using IPv6 link-local addresses, not IPv6 global unicast addresses.
- **DR/BDR election process** - The DR/BDR election process remains unchanged in OSPFv3.
- **Router ID** - Both OSPFv2 and OSPFv3 use a 32-bit number for the router ID represented in dotted-decimal notation. Typically this is an IPv4 address. The OSPF **router-id** command must be used to configure the router ID. The process in determining the 32-bit Router ID is the same in both protocols. Use an explicitly-configured router ID; otherwise, the highest loopback or configured active IPv4 address becomes the router ID.

3.10.1.3 Differences Between OSPFv2 and OSPFv3

The differences between OSPFv2 and OSPFv3:

- **Advertises** - OSPFv2 advertises IPv4 routes, whereas OSPFv3 advertises routes for IPv6.
- **Source address** - OSPFv2 messages are sourced from the IPv4 address of the exit interface. In OSPFv3, OSPF messages are sourced using the link-local address of the exit interface.
- **All OSPF router multicast addresses** - OSPFv2 uses 224.0.0.5; whereas, OSPFv3 uses FF02::5.
- **DR/BDR multicast address** - OSPFv2 uses 224.0.0.6; whereas, OSPFv3 uses FF02::6.
- **Advertise networks** - OSPFv2 advertises networks using the **network** router configuration command; whereas, OSPFv3 uses the **ipv6 ospf** *process-id* **area** *area-id* interface configuration command.
- **IP unicast routing** - Enabled, by default, in IPv4; whereas, the **ipv6 unicast-routing** global configuration command must be configured.
- **Authentication** - OSPFv2 uses either plaintext authentication, MD5, or HMAC-SHA authentication. OSPFv3 uses IPsec to add authentication for OSPFv3 packets.

3.10.1.4 Link-Local Addresses

Routers running a dynamic routing protocol, such as OSPF, exchange messages between neighbors on the same subnet or link. Routers only need to send and receive routing protocol messages with their directly connected neighbors. These messages are always sent from the source IP address of the router doing the forwarding.

IPv6 link-local addresses are ideal for this purpose. An IPv6 link-local address enables a device to communicate with other IPv6-enabled devices on the same link and only on that link (subnet). Packets with a source or destination link-local address cannot be routed beyond the link from where the packet originated.

OSPFv3 messages are sent using:

- **Source IPv6 address** - This is the IPv6 link-local address of the exit interface.
- **Destination IPv6 address** - OSPFv3 packets can be sent to a unicast address using the neighbor IPv6 link-local address. They can also be sent using a multicast address. The FF02::5 address is the all OSPF router address, while the FF02::6 is the DR/BDR multicast address.

3.11 Multiarea OSPF Operation

3.11.1 Why Multiarea OSPF?

3.11.1.1 Single-Area OSPF

Single-area OSPF is useful in smaller networks where the web of router links is not complex, and paths to individual destinations are easily deduced.

However, if an area becomes too big, the following issues must be addressed (see the figure for illustration):

- **Large routing table** - OSPF does not perform route summarization by default. If the routes are not summarized, the routing table can become very large, depending on the size of the network.
- **Large link-state database (LSDB)** - In single-area OSPF, the LSDB covers the topology of the entire routing domain. Each router must maintain detailed information about every network in the routing domain.
- **Frequent SPF algorithm calculations** - In a large network, changes are inevitable, so the routers spend many CPU cycles recalculating the SPF algorithm and updating the routing table.

To make OSPF more efficient and scalable, OSPF supports hierarchical routing using areas. An OSPF area is a group of routers that share the same link-state information in their link-state databases.

Note: OSPF route summarization is beyond the scope of this course.

3.11.1.2 Multiarea OSPF

When a large OSPF area is divided into smaller areas, this is called multiarea OSPF. Multiarea OSPF is useful in larger network deployments to reduce processing and memory overhead.

For instance, any time a router receives new information about the topology, as with additions, deletions, or modifications of a link, the router must rerun the SPF algorithm, create a new SPF tree, and update the routing table. The SPF algorithm is CPU-intensive and the time it takes for calculation depends on the size of the area. Too many routers in one area make the LSDB larger and increase the load on the CPU. Therefore, arranging routers into areas effectively partitions one potentially large database into smaller and more manageable databases.

Multiarea OSPF requires a hierarchical network design. The main area is called the backbone area (area 0) and all other areas must connect to the backbone area. With hierarchical routing, routing still occurs between the areas (interarea routing). However, the CPU intensive routing operation of recalculating the

SPF algorithm is done only for routes within an area. A change in one area does not cause an SPF algorithm recalculation in other areas.

The hierarchical-topology possibilities of multiarea OSPF have these advantages:

- **Smaller routing tables** - There are fewer routing table entries as network addresses can be summarized between areas. Also, routers in an area may only receive a default route for destination outside their area. For example, R1 summarizes the routes from area 1 to area 0 and R2 summarizes the routes from area 51 to area 0. R1 and R2 also propagate a default static route to area 1 and area 51.
- **Reduced link-state update overhead** - Minimizes processing and memory requirements, because there are fewer routers exchanging LSAs with detailed topology information.
- **Reduced frequency of SPF calculations** - Localizes impact of a topology change within an area. For instance, it minimizes routing update impact, because LSA flooding stops at the area boundary.

Assume a link fails between two internal routers in area 51. Only the routers in area 51 exchange LSAs that require them to rerun the SPF algorithm for this event. R1 receives a different type of LSA from area 51 and does not recalculate the SPF algorithm. The different types of LSAs are discussed later in this chapter.

3.11.1.3 OSPF Two-Layer Area Hierarchy

Multiarea OSPF is implemented in a two-layer area hierarchy:

- **Backbone (Transit) area** - An OSPF area whose primary function is the fast and efficient movement of IP packets. Backbone areas interconnect with other OSPF area types. Generally, end users are not found within a backbone area. The backbone area is also called OSPF area 0. Hierarchical networking defines area 0 as the core to which all other areas directly connect.
- **Regular (Non-backbone) area** - Connects users and resources. Regular areas are usually set up along functional or geographical groupings. By default, a regular area does not allow traffic from another area to use its links to reach other areas. All traffic from other areas must cross a transit area.

Note: A regular area can have a number of subtypes, including a standard area, stub area, totally stubby area, and not-so-stubby area (NSSA). Stub, totally stubby, and NSSAs are beyond the scope of this chapter.

OSPF enforces this rigid two-layer area hierarchy. The underlying physical connectivity of the network must map to the two-layer area structure, with all non-backbone areas attaching directly to area 0. All traffic moving from one area to another area must traverse the backbone area. This traffic is referred to as interarea traffic.

The optimal number of routers per area varies based on factors such as network stability, but Cisco recommends the following guidelines:

- An area should have no more than 50 routers.
- A router should not be in more than three areas.
- Any single router should not have more than 60 neighbors.

3.11.1.4 Types of OSPF Routers

OSPF routers of different types control the traffic that goes in and out of areas. The OSPF routers are categorized based on the function they perform in the routing domain.

There are four different types of OSPF routers:

- **Internal router** – This is a router that has all of its interfaces in the same area. All internal routers in an area have identical LSDBs.
- **Backbone router** – This is a router in the backbone area. The backbone area is set to area 0.
- **Area Border Router (ABR)** – This is a router that has interfaces attached to multiple areas. It must maintain separate LSDBs for each area it is connected to, and can route between areas. ABRs are exit points for the area, which means that routing information destined for another area can get there only via the ABR of the local area. ABRs can be configured to summarize the routing information from the LSDBs of their attached areas. ABRs distribute the routing information into the backbone. The backbone routers then forward the information to the other ABRs. In a multiarea network, an area can have one or more ABRs.
- **Autonomous System Boundary Router (ASBR)** – This is a router that has at least one interface attached to an external internetwork. An external network is a network that is not part of this OSPF routing domain. For example, a network connection to an ISP. An ASBR can import external network information to the OSPF network, and vice versa, using a process called route redistribution.

Redistribution in multiarea OSPF occurs when an ASBR connects different routing domains (e.g., EIGRP and OSPF) and configures them to exchange and advertise routing information between those routing domains. A static route, including a default route, can also be redistributed as an external route into the OSPF routing domain.

A router can be classified as more than one router type. For example, if a router connects to area 0 and area 1, and in addition maintains routing information for external networks, it falls under three different classifications: a backbone router, an ABR, and an ASBR.

3.11.2 Multiarea OSPF LSA Operation

3.11.2.1 OSPF LSA Types

LSAs are the building blocks of the OSPF LSDB. Individually, they act as database records and provide specific OSPF network details. In combination, they describe the entire topology of an OSPF network or area.

The RFCs for OSPF currently specify up to 11 different LSA types. However, any implementation of multiarea OSPF must support the first five LSAs: LSA 1 to LSA 5. The focus of this topic is on these first five LSAs.

Each router link is defined as an LSA type. The LSA includes a link ID field that identifies, by network number and mask, the object to which the link connects. Depending on the type, the link ID has different meanings. LSAs differ on how they are generated and propagated within the routing domain.

Note: OSPFv3 includes additional LSA types.

3.11.2.2 OSPF LSA Type 1

All routers advertise their directly connected OSPF-enabled links in a type 1 LSA and forward their network information to OSPF neighbors. The LSA contains a list of the directly connected interfaces, link types, neighbors, and link states.

Type 1 LSAs are also referred to as router link entries.

Type 1 LSAs are flooded only within the area in which they originated. ABRs subsequently advertise the networks learned from the type 1 LSAs to other areas as type 3 LSAs.

The type 1 LSA link ID is identified by the router ID of the originating router.

3.11.2.3 OSPF LSA Type 2

A type 2 LSA only exists for multiaccess and non-broadcast multiaccess (NBMA) networks where there is a DR elected and at least two routers on the multiaccess segment. The type 2 LSA contains the router ID and IP address of the DR, along with the router ID of all other routers on the multiaccess segment. A type 2 LSA is created for every multiaccess network in the area.

The purpose of a type 2 LSA is to give other routers information about multiaccess networks within the same area.

The DR floods type 2 LSAs only within the area in which they originated. Type 2 LSAs are not forwarded outside of an area.

Type 2 LSAs are also referred to as network link entries.

ABR1 is the DR for the Ethernet network in area 1. It generates the type 2 LSA and forwards it into area 1. ABR2 is the DR for the multiaccess network in area 0. There are no multiaccess networks in area 2 and therefore, no type 2 LSAs are ever propagated in that area.

The link-state ID for a network LSA is the IP interface address of the DR that advertises it.

3.11.2.4 OSPF LSA Type 3

Type 3 LSAs are used by ABRs to advertise networks from other areas. ABRs collect type 1 LSAs in the LSDB. After an OSPF area has converged, the ABR creates a type 3 LSA for each of its learned OSPF networks. Therefore, an ABR with many OSPF routes must create type 3 LSAs for each network.

ABR1 and ABR2 floods type 3 LSAs from one area to other areas. ABR1 propagates the Area 1 information into Area 0 using Type 3 LSAs. ABR1 also propagates the Area 0 information into Area 1 using Type 3 LSAs. ABR2 does the same thing for Area 2 and Area 0. In a large OSPF deployment with many networks, propagating type 3 LSAs can cause significant flooding problems. For this reason, it is strongly recommended that manual route summarization be configured on the ABR.

The link-state ID is set to the network number and the mask is also advertised.

Receiving a type 3 LSA into an area does not cause a router to run the SPF algorithm. The routes being advertised in the type 3 LSAs are appropriately added to or deleted from the router's routing table, but a full SPF calculation is not necessary.

3.11.2.5 OSPF LSA Type 4

Type 4 and type 5 LSAs are used collectively to identify an ASBR and advertise external networks into an OSPF routing domain.

A type 4 summary LSA is generated by an ABR only when an ASBR exists within an area. A type 4 LSA identifies the ASBR and provides a route to it. All traffic destined to an external network requires routing table knowledge of the ASBR that originated the external routes.

The ASBR sends a type 1 LSA, identifying itself as an ASBR. The LSA includes a special bit known as the external bit (e bit) that is used to identify the router as an ASBR. When ABR1 receives the type 1 LSA, it notices the e bit, it builds a type 4 LSA, and then floods the type 4 LSA to the backbone (area 0). Subsequent ABRs flood the type 4 LSA into other areas.

The link-state ID is set to the ASBR router ID.

3.11.2.6 OSPF LSA Type 5

Type 5 external LSAs describe routes to networks outside the OSPF routing domain. Type 5 LSAs are originated by the ASBR and are flooded to the entire routing domain.

Type 5 LSAs are also referred to as external LSA entries.

The ASBR generates type 5 LSAs for each external route and floods it into the area. Subsequent ABRs also flood the type 5 LSA into other areas. Routers in other areas use the information from the type 5 LSA to reach the external routes.

In a large OSPF deployment with many networks, propagating multiple type 5 LSAs can cause significant flooding problems. For this reason, it is strongly recommended that manual route summarization be configured on the ASBR.

The link-state ID is the external network number.

3.11.3 OSPF Routing Table and Types of Routes

3.11.3.1 OSPF Routing Table Entries

OSPF routes in an IPv4 routing table are identified using the following descriptors:

- **O** - Router (type 1) and network (type 2) LSAs describe the details within an area. The routing table reflects this link-state information with a designation of **O**, meaning that the route is intra-area.
- **O IA** – When an ABR receives a router LSA (type 1) in one area; it sends a summary LSA (type 3) into the adjacent area. Summary LSAs appear in the routing table as IA (interarea routes). Summary LSAs received in one area are also forwarded to other areas.
- **O E1** or **O E2** - External LSAs appear in the routing table marked as external type 1 (E1) or external type 2 (E2) routes. Type 2 (E2) is the default. The difference between type 1 (E1) and type 2 (E2) are beyond the scope of this course.

3.11.3.2 OSPF Route Calculation

Each router uses the SPF algorithm against the LSDB to build the SPF tree. The SPF tree is used to determine the best path(s).

The order in which the best paths are calculated is as follows:

1. All routers calculate the best path(s) to destinations within their area (intra-area) and add these entries to the routing table. These are the type 1 and type 2 LSAs, which are noted in the routing table with a routing designator of O. (1)

2. All routers calculate the best path(s) to the other areas within the internetwork. These best paths are the interarea route entries, or type 3 LSAs, and are noted with a routing designator of O IA. (2)

3. All routers (except those that are in a form of stub area) calculate the best path(s) to the external autonomous system (type 5) destinations. These are noted with either an O E1 or an O E2 route designator, depending on the configuration. (3)

When converged, a router can communicate with any network within or outside the OSPF routing domain.

3.12 Configuring Multiarea OSPF

3.12.1 Configuring Multiarea OSPF

3.12.1.1 Implementing Multiarea OSPF

OSPF can be implemented as single-area or multiarea. The type of OSPF implementation chosen depends on the specific network design requirements and existing topology.

There are 4 steps to implementing multiarea OSPF.

Steps 1 and 2 are part of the planning process.

Step 1. Gather the network requirements and parameters - Gather the network requirements and parameters - This includes determining the number of host and network devices, the IP addressing scheme (if already implemented), the size of the routing domain, the size of the routing tables, the risk of topology changes, whether existing routers can support OSPF, and other network characteristics.

Step 2. Define the OSPF parameters - Based on information gathered during Step 1, the network administrator must determine if single-area or multiarea OSPF is the preferred implementation. If multiarea OSPF is selected, there are several considerations the network administrator must take into account while determining the OSPF parameters, to include:

- **IP addressing plan** - This governs how OSPF can be deployed and how well the OSPF deployment might scale. A detailed IP addressing plan, along with the IP subnetting information, must be created. A good IP addressing plan should enable the usage of OSPF multiarea design and summarization. This plan more easily scales the network, as well as optimizes OSPF behavior and the propagation of LSA.
- **OSPF areas** - Dividing an OSPF network into areas decreases the LSDB size and limits the propagation of link-state updates when the topology changes. The routers that are to be ABRs and ASBRs must be identified, as are those ABRs or ASBRs that are to perform any summarization or redistribution.
- **Network topology** - This consists of links that connect the network equipment and belong to different OSPF areas in a multiarea OSPF design. Network topology is important to determine primary and backup links. Primary and backup links are defined by the changing OSPF cost on interfaces. A detailed network topology plan should also be used to determine the different OSPF areas, ABR, and ASBR as well as summarization and redistribution points, if multiarea OSPF is used.

Step 3. Configure the multiarea OSPF implementation based on the parameters.

Step 4. Verify the multiarea OSPF implementation based on the parameters.

3.12.1.2 Configuring Multiarea OSPFv2

There are no special commands required to implement this multiarea OSPF network. A router simply becomes an ABR when it has two **network** statements in different areas.

```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.1.1.1 0.0.0.0 area 1
R1(config-router)# network 10.1.2.1 0.0.0.0 area 1
R1(config-router)# network 192.168.10.1 0.0.0.0 area 0
R1(config-router)# end
```

As shown in Figure above, R1 is assigned the router ID 1.1.1.1. This example enables OSPF on the two LAN interfaces in area 1. The serial interface is configured as part of OSPF area 0. Because R1 has interfaces connected to two different areas, it is an ABR.

Note: The inverse wildcard masks used to configure R2 and R3 purposely differ to demonstrate the two alternatives to entering **network** statements. The interface method used for R3 is simpler because the wildcard mask is always **0.0.0.0** and does not need to be calculated.

3.12.2 Verifying Multiarea OSPF

3.12.2.1 Verifying Multiarea OSPFv2

The same verification commands used to verify single-area OSPFv2 also can be used to verify the multiarea OSPF topology:

- show ip ospf neighbor
- show ip ospf
- show ip ospf interface

Commands that verify specific multiarea OSPFv2 information include:

- show ip protocols
- show ip ospf interface brief
- show ip route ospf

- show ip ospf database

Note: For the equivalent OSPFv3 command, simply substitute **ip** with **ipv6**.

3.12.2.2 Verify General Multiarea OSPFv2 Settings

Use the **show ip protocols** command to verify the OSPFv2 status. The output of the command reveals which routing protocols are configured on a router. It also includes routing protocol specifics such as the router ID, number of areas in the router, and networks included within the routing protocol configuration.

Use the **show ip ospf interface brief** command to display concise OSPFv2-related information of OSPFv2-enabled interfaces. This command reveals useful information, such as the OSPFv2 process ID that the interface is assigned to, the area that the interfaces are in, and the cost of the interface.

3.12.2.3 Verify the OSPFv2 Routes

The most common command used to verify a multiarea OSPFv2 configuration is the **show ip route** command. Add the **ospf** parameter to display only OSPFv2-related information.

3.12.2.4 Verify the Multiarea OSPFv2 LSDB

Use the **show ip ospf** database command to verify the contents of the OSPFv2 LSDB.

There are many command options available with the **show ip ospf database** command.

CHAPTER 4: SWITCHED NETWORKS

Modern networks continue to evolve to keep pace with the changing way organizations carry out their daily business. Users now expect instant access to company resources from anywhere and at any time. These resources not only include traditional data, but also video and voice. There is also an increasing need for collaboration technologies. These technologies allow real-time sharing of resources between multiple remote individuals, as though they were at the same physical location.

Different devices must seamlessly work together to provide a fast, secure, and reliable connection between hosts. LAN switches provide the connection point for end users into the enterprise network and are also primarily responsible for the control of information within the LAN environment. Routers facilitate the movement of information between LANs, and are generally unaware of individual hosts. All advanced services depend on the availability of a robust routing and switching infrastructure on which they can build. This infrastructure must be carefully designed, deployed, and managed, to provide a stable platform.

4.1 LAN Design

4.1.1 Converged Networks

4.1.1.1 Growing Complexity of Networks

Our digital world is changing. The ability to access the Internet and the corporate network is no longer confined to physical offices, geographical locations, or time zones. In today's globalized workplace, employees can access resources from anywhere in the world and information must be available at any time, and on any device. These requirements drive the need to build next-generation networks that are secure, reliable, and highly available.

These next generation networks must not only support current expectations and equipment, but must also be able to integrate legacy platforms.

4.1.1.2 Elements of a Converged Network

To support collaboration, business networks employ converged solutions using voice systems, IP phones, voice gateways, video support, and video conferencing. Including data services, a converged network with collaboration support may include the following features:

- **Call control** - Telephone call processing, caller ID, call transfer, hold, and conference
- **Voice messaging** - Voicemail
- **Mobility** - Receive important calls wherever you are
- **Automated attendant** - Serve customers faster by routing calls directly to the right department or individual

One of the primary benefits of transitioning to the converged network is that there is just one physical network to install and manage. This results in substantial savings over the installation and management of separate voice, video, and data networks. Such a converged network solution integrates IT management so that any moves, additions, and changes are completed with an intuitive management interface. A converged network solution also provides PC softphone application support, as well as point-to-point video, so that users can enjoy personal communications with the same ease of administration and use as a voice call.

The convergence of services onto the network has resulted in an evolution in networks from a traditional data transport role, to a super-highway for data, voice, and video communication. This one physical network must be properly designed and implemented to allow the reliable handling of the various types of information that it must carry. A structured design is required to allow management of this complex environment.

4.1.1.3 Cisco Borderless Networks

With the increasing demands of the converged network, the network must be developed with an architectural approach that embeds intelligence, simplifies operations, and is scalable to meet future demands. One of the more recent developments in network design is the Cisco Borderless Network.

The Cisco Borderless Network is a network architecture that combines innovation and design. It allows organizations to support a borderless network that can connect anyone, anywhere, anytime, on any device; securely, reliably, and seamlessly. This architecture is designed to address IT and business challenges, such as supporting the converged network and changing work patterns.

The Cisco Borderless Network provides the framework to unify wired and wireless access, including policy, access control, and performance management across many different device types. Using this architecture, the borderless network is built on a hierarchical infrastructure of hardware that is scalable and resilient. By combining this hardware infrastructure with policy-based software solutions, the Cisco Borderless Network provides two primary sets of services: network services, and user and endpoint services that are all managed by an integrated management solution. It enables different network elements to work together, and allows users to access resources from any place, at any time, while providing optimization, scalability, and security.

4.1.1.4 Hierarchy in the Borderless Switched Network

Creating a borderless switched network requires that sound network design principles are used to ensure maximum availability, flexibility, security, and manageability. The borderless switched network must deliver on current requirements and future required services and technologies. Borderless switched network design guidelines are built upon the following principles:

- **Hierarchical** - Facilitates understanding the role of each device at every tier, simplifies deployment, operation, and management, and reduces fault domains at every tier
- **Modularity** - Allows seamless network expansion and integrated service enablement on an on-demand basis
- **Resiliency** - Satisfies user expectations for keeping the network always on
- **Flexibility** - Allows intelligent traffic load sharing by using all network resources

These are not independent principles. Understanding how each principle fits in the context of the others is critical. Designing a borderless switched network in a hierarchical fashion creates a foundation that allows network designers to overlay security, mobility, and unified communication features. Two time-tested and proven hierarchical design frameworks for campus networks are the three-tier layer and the two-tier layer models.

The three critical layers within these tiered designs are the access, distribution, and core layers. Each layer can be seen as a well-defined, structured module with specific roles and functions in the campus network. Introducing modularity into the campus hierarchical design further ensures that the campus network remains resilient and flexible enough to provide critical network services. Modularity also helps to allow for growth and changes that occur over time.

4.1.1.5 Access, Distribution, and Core Layers

Access Layer

The access layer represents the network edge, where traffic enters or exits the campus network. Traditionally, the primary function of an access layer switch is to provide network access to the user. Access layer switches connect to distribution layer switches, which implement network foundation technologies such as routing, quality of service, and security.

To meet network application and end-user demand, the next-generation switching platforms now provide more converged, integrated, and intelligent services to various types of endpoints at the network edge. Building intelligence into access layer switches allows applications to operate on the network more efficiently and securely.

Distribution Layer

The distribution layer interfaces between the access layer and the core layer to provide many important functions, including:

- Aggregating large-scale wiring closet networks
- Aggregating Layer 2 broadcast domains and Layer 3 routing boundaries
- Providing intelligent switching, routing, and network access policy functions to access the rest of the network
- Providing high availability through redundant distribution layer switches to the end-user and equal cost paths to the core
- Providing differentiated services to various classes of service applications at the edge of the network

Core Layer

The core layer is the network backbone. It connects several layers of the campus network. The core layer serves as the aggregator for all of the other campus blocks and ties the campus together with the rest of the network. The primary purpose of the core layer is to provide fault isolation and high-speed backbone connectivity.

To build a simplified, scalable, cost-effective, and efficient physical cable layout design, the recommendation is to build an extended-star physical network topology from a centralized building location to all other buildings on the same campus.

In some cases where extensive physical or network scalability does not exist, maintaining separate distribution and core layers is not required. In smaller campus locations where there are fewer users accessing the network or in campus sites consisting of a single building, separate core and distribution layers may not be needed. In this scenario, the recommendation is the alternate two-tier campus network design, also known as the collapsed core network design.

4.1.2 Switched Networks

4.1.2.1 Role of Switched Networks

The role of switched networks has evolved dramatically in the last two decades. It was not long ago that flat Layer 2 switched networks were the norm. Flat Layer 2 switched networks relied on the Ethernet and the widespread use of hub repeaters to propagate LAN traffic throughout an organization. A switched LAN allows more flexibility, traffic management, and additional features:

- Quality of service
- Additional security
- Support for wireless networking and connectivity
- Support for new technologies, such as IP telephony and mobility services

4.1.2.2 Form Factors

There are various types of switches used in business networks. It is important to deploy the appropriate types of switches based on network requirements.

Common business considerations when selecting Sqitch equipment:

- ✓ **Cost:** The cost of a switch will depend on the number and speed of the interfaces, supported features, and expansion capability.
- ✓ **Port Density:** Network switches must support the appropriate number of devices on the network.
- ✓ **Power:** It is now common to power access points, IP phones, and even compact switches using Power over Ethernet (PoE). In addition to PoE considerations, some chassis.based switches support redundant power supplies.
- ✓ **Reliability:** The switch should provide continuous access to the network.
- ✓ **Port speed:** The speed of the network connection is of primary concern to end users.
- ✓ **Frame buffers:** The ability of the switch to store frames is important in a network where there may be congested ports to servers or other areas of the network.
- ✓ **Scalability:** The number of users on a network typically grows over time; therefore, the switch should provide the opportunity for growth.

When selecting the type of switch, the network designer must choose between a fixed configuration or a modular configuration, and stackable or non-stackable. Another consideration is the thickness of the switch, which is expressed in number of rack units. This is important for switches that are mounted in a rack. These options are sometimes referred to as switch form factors.

Fixed Configuration Switches

Fixed configuration switches do not support features or options beyond those that originally came with the switch. The particular model determines the features and options available. For example, a 24-port gigabit fixed switch cannot support additional ports. There are typically different configuration choices that vary in how many and what types of ports are included with a fixed configuration switch.

Modular Configuration Switches

Modular configuration switches offer more flexibility in their configuration. Modular configuration switches typically come with different sized chassis that allow for the installation of different numbers of modular line cards. The line cards actually contain the ports. The line card fits into the switch chassis the way that expansion cards fit into a PC. The larger the chassis, the more modules it can support. There are many different chassis sizes. A modular switch with a single 24-port line card could have an additional 24-port line card installed to bring the total number of ports up to 48.

Stackable Configuration Switches

Stackable configuration switches can be interconnected using a special cable that provides high-bandwidth throughput between the switches. Cisco StackWise technology allows the interconnection of up to nine switches. Switches can be stacked one on top of the other with cables connecting the switches in a daisy chain fashion. The stacked switches effectively operate as a single larger switch. Stackable switches are desirable where fault tolerance and bandwidth availability are critical and a modular switch is too costly to implement. By cross-connecting these stacked switches,, the network can recover quickly if a single switch fails. Stackable switches use a special port for interconnections. Many Cisco stackable switches also support StackPower technology, which enables power sharing among stack members.

4.2 The Switched Environment

4.2.1 Frame Forwarding

4.2.1.1 Switching as a General Concept in Networking and Telecommunications

The concept of switching and forwarding frames is universal in networking and telecommunications. Various types of switches are used in LANs, WANs, and the public switched telephone network (PSTN). The fundamental concept of switching refers to a device making a decision based on two criteria:

- Ingress port

- Destination address

The decision on how a switch forwards traffic is made in relation to the flow of that traffic. The term ingress is used to describe where a frame enters the device on a port. The term egress is used to describe frames leaving the device from a particular port.

A LAN switch maintains a table that it uses to determine how to forward traffic through the switch.

The only intelligence of the LAN switch is its ability to use its table to forward traffic based on the ingress port and the destination address of a message. With a LAN switch, there is only one master switching table that describes a strict association between addresses and ports; therefore, a message with a given destination address always exits the same egress port, regardless of the ingress port it enters.

Layer 2 Ethernet switches forward Ethernet frames based on the destination MAC address of the frames.

4.2.1.2 Dynamically Populating a Switch MAC Address Table

Switches use MAC addresses to direct network communications through the switch, to the appropriate port, toward the destination. A switch is made up of integrated circuits and the accompanying software that controls the data paths through the switch. For a switch to know which port to use to transmit a frame, it must first learn which devices exist on each port. As the switch learns the relationship of ports to devices, it builds a table called a MAC address, or content addressable memory (CAM) table. CAM is a special type of memory used in high-speed searching applications.

LAN switches determine how to handle incoming data frames by maintaining the MAC address table. A switch builds its MAC address table by recording the MAC address of each device connected to each of its ports. The switch uses the information in the MAC address table to send frames destined for a specific device out the port which has been assigned to that device.

The following two-step process is performed on every Ethernet frame that enters a switch.

Step 1. Learn – Examining the Source MAC Address

Every frame that enters a switch is checked for new information to learn. It does this by examining the frame's source MAC address and port number where the frame entered the switch:

- If the source MAC address does not exist, it is added to the table along with the incoming port number.
- If the source MAC address does exist, the switch updates the refresh timer for that entry. By default, most Ethernet switches keep an entry in the table for five minutes.

Note: If the source MAC address does exist in the table but on a different port, the switch treats this as a new entry. The entry is replaced using the same MAC address, but with the more current port number.

Step 2. Forward – Examining the Destination MAC Address

If the destination MAC address is a unicast address, the switch will look for a match between the destination MAC address of the frame and an entry in its MAC address table:

- If the destination MAC address is in the table, it will forward the frame out the specified port.
- If the destination MAC address is not in the table, the switch will forward the frame out all ports except the incoming port. This is called an unknown unicast.

Note: If the destination MAC address is a broadcast or a multicast, the frame is also flooded out all ports except the incoming port.

4.2.1.3 Switch Forwarding Methods

As networks grew and enterprises began to experience slower network performance, Ethernet bridges (an early version of a switch) were added to networks to limit the size of the collision domains. In the 1990s,

advancements in integrated circuit technologies allowed for Ethernet LAN switches to replace Ethernet bridges. These switches were able to move the Layer 2 forwarding decisions from software to application-specific-integrated circuits (ASICs). ASICs reduce the packet-handling time within the device, and allow the device to handle an increased number of ports without degrading performance. This method of forwarding data frames at Layer 2 was referred to as store-and-forward switching. This term distinguished it from cut-through switching.

The store-and-forward method makes a forwarding decision on a frame after it has received the entire frame and checked the frame for errors using a mathematical error-checking mechanism known as a cyclic redundancy check (CRC).

By contrast, the cut-through method begins the forwarding process after the destination MAC address of an incoming frame and the egress port has been determined.

4.2.1.4 Store-and-Forward Switching

Store-and-forward switching has two primary characteristics that distinguish it from cut-through: error checking and automatic buffering.

Error Checking

A switch using store-and-forward switching performs an error check on an incoming frame. After receiving the entire frame on the ingress port, the switch compares the frame-check-sequence (FCS) value in the last field of the datagram against its own FCS calculations. The FCS is an error checking process that helps to ensure that the frame is free of physical and data-link errors. If the frame is error-free, the switch forwards the frame. Otherwise the frame is dropped.

Automatic Buffering

The ingress port buffering process used by store-and-forward switches provides the flexibility to support any mix of Ethernet speeds. For example, handling an incoming frame traveling into a 100 Mb/s Ethernet port that must be sent out a 1 Gb/s interface would require using the store-and-forward method. With any mismatch in speeds between the ingress and egress ports, the switch stores the entire frame in a buffer, computes the FCS check, forwards it to the egress port buffer and then sends it.

Store-and-forward switching is Cisco's primary LAN switching method.

A store-and-forward switch drops frames that do not pass the FCS check; therefore, it does not forward invalid frames. By contrast, a cut-through switch may forward invalid frames because no FCS check is performed.

4.2.1.5 Cut-Through Switching

An advantage to cut-through switching is the ability of the switch to start forwarding a frame earlier than store-and-forward switching. There are two primary characteristics of cut-through switching: rapid frame forwarding and fragment free.

Rapid Frame Forwarding

Aswitch using the cut-through method can make a forwarding decision as soon as it has looked up the destination MAC address of the frame in its MAC address table. The switch does not have to wait for the rest of the frame to enter the ingress port before making its forwarding decision.

With today's MAC controllers and ASICs, a switch using the cut-through method can quickly decide whether it needs to examine a larger portion of a frame's headers for additional filtering purposes. For example, the switch can analyze past the first 14 bytes (the source MAC address, destination MAC, and the EtherType fields), and examine an additional 40 bytes in order to perform more sophisticated functions relative to IPv4 Layers 3 and 4.

The cut-through switching method does not drop most invalid frames. Frames with errors are forwarded to other segments of the network. If there is a high error rate (invalid frames) in the network, cut-through switching can have a negative impact on bandwidth; thus, clogging up bandwidth with damaged and invalid frames.

Fragment Free

Fragment free switching is a modified form of cut-through switching in which the switch waits for the collision window (64 bytes) to pass before forwarding the frame. This means each frame will be checked into the data field to make sure no fragmentation has occurred. Fragment free switching provides better error checking than cut-through, with practically no increase in latency.

The lower latency speed of cut-through switching makes it more appropriate for extremely demanding, high-performance computing (HPC) applications that require process-to-process latencies of 10 microseconds or less.

4.2.2 Switching Domains

4.2.2.1 Collision Domains

In hub-based Ethernet segments, network devices compete for the medium, because devices must take turns when transmitting. The network segments that share the same bandwidth between devices are known as collision domains. When two or more devices within that the same collision domain try to communicate at the same time, a collision will occur.

If an Ethernet switch port is operating in half duplex, each segment is in its own collision domain. However, Ethernet switch ports operating in full duplex eliminate collisions; therefore, there is no collision domain. By default, Ethernet switch ports will autonegotiate full duplex when the adjacent device can also operate in full duplex. If the switch port is connected to a device operating in half-duplex, such as a legacy hub, then the switch port will operate in half duplex. In the case of half duplex, the switch port will be part of a collision domain.

Full-duplex is chosen if both devices have the capability along with their highest common bandwidth.

4.2.2.2 Broadcast Domains

A collection of interconnected switches forms a single broadcast domain. Only a network layer device, such as a router, can divide a Layer 2 broadcast domain. Routers are used to segment broadcast domains, but will also segment a collision domain.

When a device sends a Layer 2 broadcast, the destination MAC address in the frame is set to all binary ones.

The Layer 2 broadcast domain is referred to as the MAC broadcast domain. The MAC broadcast domain consists of all devices on the LAN that receive broadcast frames from a host.

When a switch receives a broadcast frame, it forwards the frame out each of its ports, except the ingress port where the broadcast frame was received. Each device connected to the switch receives a copy of the broadcast frame and processes it. Broadcasts are sometimes necessary for initially locating other devices and network services, but they also reduce network efficiency. Network bandwidth is used to propagate the broadcast traffic. Too many broadcasts and a heavy traffic load on a network can result in congestion, which slows down network performance.

When two switches are connected together, the broadcast domain is increased, as seen in the second half of the animation. In this case, a broadcast frame is forwarded to all connected ports on switch S1. Switch S1 is connected to switch S2. The frame is then also propagated to all devices connected to switch S2.

4.2.2.3 Alleviating Network Congestion

LAN switches have special characteristics that make them effective at alleviating network congestion. By default, interconnected switch ports attempt to establish a link in full duplex, therefore eliminating collision domains. Each full duplex port of the switch provides the full bandwidth to the device or devices that are connected to that port. A full-duplex connection can carry transmitted and received signals at the same time. Full-duplex connections have dramatically increased LAN network performance, and are required for 1 Gb/s Ethernet speeds and higher.

Switches interconnect LAN segments, use a table of MAC addresses to determine the segment to which the frame is to be sent, and can lessen or eliminate collisions entirely. The following are some important characteristics of switches that contribute to alleviating network congestion:

- **High port density** - Switches have high-port densities: 24- and 48-port switches are often just a single rack unit and operate at speeds of 100 Mb/s, 1 Gb/s, and 10 Gb/s. Large enterprise switches may support many hundreds of ports.
- **Large frame buffers** - The ability to store more received frames before having to start dropping them is useful, particularly when there may be congested ports to servers or other parts of the network.
- **Port speed** - Depending on the cost of a switch, it may be possible to support a mixture of speeds. Ports of 100 Mb/s, and 1 or 10 Gb/s are common (100 Gb/s is also possible).
- **Fast internal switching** - Having fast internal forwarding capabilities allows high performance. The method that is used may be a fast internal bus or shared memory, which affects the overall performance of the switch.
- **Low per-port cost** - Switches provide high-port density at a lower cost.

CHAPTER 5: SWITCH CONFIGURATION

Switches are used to connect multiple devices together on the same network. In a properly designed network, LAN switches are responsible for directing and controlling the data flow at the access layer to networked resources.

Cisco switches are self-configuring and no additional configurations are necessary for them to function out of the box. However, Cisco switches run Cisco IOS, and can be manually configured to better meet the needs of the network. This includes adjusting port speed, bandwidth, and security requirements.

Additionally, Cisco switches can be managed both locally and remotely. To remotely manage a switch, it needs to have an IP address and default gateway configured. These are just two of the configurations discussed in this chapter.

Switches operate at the access layer where client network devices connect directly to the network and IT departments want uncomplicated network access for the users. It is one of the most vulnerable areas of the network because it is so exposed to the user. Switches need to be configured to be resilient to attacks of all types while they are protecting user data and allowing for high speed connections. Port security is one of the security features that Cisco managed switches provide.

5.1 Basic Switch Configuration
5.1.1 Configure a Switch with Initial Settings

5.1.1.1 Switch Boot Sequence
After a Cisco switch is powered on, it goes through the following boot sequence:

1. First, the switch loads a power-on self-test (POST) program stored in ROM. POST checks the CPU subsystem. It tests the CPU, DRAM, and the portion of the flash device that makes up the flash file system.

2. Next, the switch loads the boot loader software. The boot loader is a small program stored in ROM that is run immediately after POST successfully completes.

3. The boot loader performs low-level CPU initialization. It initializes the CPU registers, which control where physical memory is mapped, the quantity of memory, and its speed.

4. The boot loader initializes the flash file system on the system board.

5. Finally, the boot loader locates and loads a default IOS operating system software image into memory and gives control of the switch over to the IOS.

The boot loader finds the Cisco IOS image on the switch as follows: the switch attempts to automatically boot by using information in the BOOT environment variable. If this variable is not set, the switch attempts to load and execute the first executable file it can by performing a recursive, depth-first search throughout the flash file system. In a depth-first search of a directory, each encountered subdirectory is completely searched before continuing the search in the original directory. On Catalyst 2960 Series switches, the image file is normally contained in a directory that has the same name as the image file (excluding the .bin file extension).

The IOS operating system then initializes the interfaces using the Cisco IOS commands found in the startup-config file, which is stored in NVRAM.

5.1.1.2 Recovering From a System Crash
The boot loader provides access into the switch if the operating system cannot be used because of missing or damaged system files. The boot loader has a command-line that provides access to the files stored in flash memory.

The boot loader can be accessed through a console connection following these steps:

Step 1. Connect a PC by console cable to the switch console port. Configure terminal emulation software to connect to the switch.

Step 2. Unplug the switch power cord.

Step 3. Reconnect the power cord to the switch and, within 15 seconds, press and hold down the **Mode** button while the System LED is still flashing green.

Step 4. Continue pressing the **Mode** button until the System LED turns briefly amber and then solid green; then release the **Mode** button.

Step 5. The boot loader **switch:** prompt appears in the terminal emulation software on the PC.

Theboot loader command line supports commands to format the flash file system, reinstall the operating system software, and recover a lost or forgotten password. For example, the **dir** command can be used to view a list of files within a specified directory.

Note: Notice that in this example, the IOS is located in the root of the flash folder.

5.1.1.3 Switch LED Indicators

Cisco Catalyst switches have several status LED indicator lights. You can use the switch LEDs to quickly monitor switch activity and performance. Switches of different models and feature sets will have different LEDs and their placement on the front panel of the switch may also vary.

The following describes the purpose of the LED indicators, and the meaning of their colors:

- **System LED** - Shows whether the system is receiving power and is functioning properly. If the LED is off, it means the system is not powered on. If the LED is green, the system is operating normally. If the LED is amber, the system is receiving power but is not functioning properly.
- **Redundant Power System (RPS) LED** - Shows the RPS status. If the LED is off, the RPS is off, or it is not properly connected. If the LED is green, the RPS is connected and ready to provide backup power. If the LED is blinking green, the RPS is connected but is unavailable because it is providing power to another device. If the LED is amber, the RPS is in standby mode, or in a fault condition. If the LED is blinking amber, the internal power supply in the switch has failed, and the RPS is providing power.
- **Port Status LED** - Indicates that the port status mode is selected when the LED is green. This is the default mode. When selected, the port LEDs will display colors with different meanings. If the LED is off, there is no link, or the port was administratively shut down. If the LED is green, a link is present. If the LED is blinking green, there is activity and the port is sending or receiving data. If the LED is alternating green-amber, there is a link fault. If the LED is amber, the port is blocked to ensure that a loop does not exist in the forwarding domain and is not forwarding data (typically, ports will remain in this state for the first 30 seconds after being activated). If the LED is blinking amber, the port is blocked to prevent a possible loop in the forwarding domain.
- **Port Duplex LED** - Indicates the port duplex mode is selected when the LED is green. When selected, port LEDs that are off are in half-duplex mode. If the port LED is green, the port is in full-duplex mode.
- **Port Speed LED** - Indicates the port speed mode is selected. When selected, the port LEDs will display colors with different meanings. If the LED is off, the port is operating at 10 Mb/s. If the LED is green, the port is operating at 100 Mb/s. If the LED is blinking green, the port is operating at 1000 Mb/s.
- **Power over Ethernet (PoE) Mode LED** - If PoE is supported; a PoE mode LED will be present. If the LED is off, it indicates the PoE mode is not selected and that none of the ports have been denied power or placed in a fault condition. If the LED is blinking amber, the PoE mode is not

selected but at least one of the ports has been denied power, or has a PoE fault. If the LED is green, it indicates the PoE mode is selected and the port LEDs will display colors with different meanings. If the port LED is off, the PoE is off. If the port LED is green, the PoE is on. If the port LED is alternating green-amber, PoE is denied because providing power to the powered device will exceed the switch power capacity. If the LED is blinking amber, PoE is off due to a fault. If the LED is amber, PoE for the port has been disabled.

5.1.1.4 Preparing for Basic Switch Management

To prepare a switch for remote management access, the switch must be configured with an IP address and a subnet mask. Keep in mind, that to manage the switch from a remote network, the switch must be configured with a default gateway. This is very similar to configuring the IP address information on host devices. The switch virtual interface (SVI) on Switch should be assigned an IP address. The SVI is a virtual interface, not a physical port on the switch.

SVI is a concept related to VLANs. VLANs are numbered logical groups to which physical ports can be assigned. Configurations and settings applied to a VLAN are also applied to all the ports assigned to that VLAN.

By default, the switch is configured to have the management of the switch controlled through VLAN 1. All ports are assigned to VLAN 1 by default. For security purposes, it is considered a best practice to use a VLAN other than VLAN 1 for the management VLAN.

Note that these IP settings are only for remote management access to the switch; the IP settings do not allow the switch to route Layer 3 packets.

5.1.1.5 Configuring Basic Switch Management Access with IPv4

Step 1. Configure Management Interface

An IPv4 address and subnet mask is configured on the management SVI of the switch from VLAN interface configuration mode. As shown in Figure below, the interface vlan 99 command is used to enter interface configuration mode. The ip address command is used to configure the IPv4 address. The no shutdown command enables the interface. In this example, VLAN 99 is configured with IPv4 address 172.17.99.11.

Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode for the SVI.	`S1(config)# interface vlan 99`
Configure the management interface IP address.	`S1(config-if)# ip address 172.17.99.11 255.255.255.0`
Enable the management interface.	`S1(config-if)# no shutdown`
Return to the privileged EXEC mode.	`S1(config-if)# end`
Save the running config to the startup config.	`S1# copy running-config startup-config`

The SVI for VLAN 99 will not appear as "up/up" until VLAN 99 is created and there is a device connected to a switch port associated with VLAN 99. To create a VLAN with the vlan_id of 99, and associate it to an interface, use the following commands:

```
S1(config)# vlan vlan_id
S1(config-vlan)# name vlan_name
S1(config-vlan)# exit
S1(config)# interface interface_id
S1(config-if)# switchport access vlan vlan_id
```

Step 2. Configure Default Gateway

The switch should be configured with a default gateway if it will be managed remotely from networks that are not directly connected. The default gateway is the router to which the switch is connected. The switch will forward its IP packets with destination IP addresses outside the local network to the default

gateway. As shown in Figure below, R1 is the default gateway for S1. The interface on R1 connected to the switch has the IPv4 address 172.17.99.1. This address is the default gateway address for S1.

Enter global configuration mode.	`S1# configure terminal`
Configure the default gateway for the switch.	`S1(config)# ip default-gateway 172.17.99.1`
Return to the privileged EXEC mode.	`S1(config)# end`
Save the running config to the startup config.	`S1# copy running-config startup-config`

To configure the default gateway for the switch, use the ip default-gateway command. Enter the IPv4 address of the default gateway. The default gateway is the IPv4 address of the router interface to which the switch is connected. Use the copy running-config startup-config command to back up your configuration.

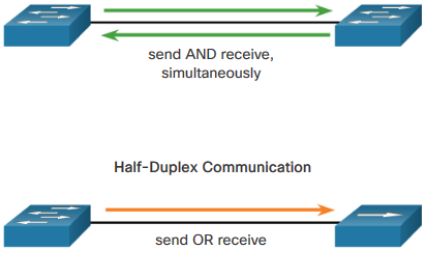
Step 3. Verify Configuration

The show ip interface brief command is useful when determining the status of both physical and virtual interfaces.

5.1.2 Configure Switch Ports

5.1.2.1 Duplex Communication

The figure illustrates full-duplex and half-duplex communication.



Full-duplex communication improves the performance of a switched LAN. Full-duplex communication increases effective bandwidth by allowing both ends of a connection to transmit and receive data simultaneously. This is also known as bidirectional communication. This method of optimizing network performance requires micro-segmentation. A micro-segmented LAN is created when a switch port has only one device connected and is operating in full-duplex mode. When a switch port is operating in full-duplex mode, there is no collision domain associated with the port.

Unlike full-duplex communication, half-duplex communication is unidirectional. Sending and receiving data does not occur at the same time. Half-duplex communication creates performance issues because data can flow in only one direction at a time, often resulting in collisions. Half-duplex connections are typically seen in older hardware, such as hubs. Full-duplex communication has replaced half-duplex in most hardware.

Gigabit Ethernet and 10Gb NICs require full-duplex connections to operate. In full-duplex mode, the collision detection circuit on the NIC is disabled. Frames that are sent by the two connected devices cannot collide because the devices use two separate circuits in the network cable. Full-duplex connections require a switch that supports full-duplex configuration, or a direct connection using an Ethernet cable between two devices.

Standard, shared hub-based Ethernet configuration efficiency is typically rated at 50 to 60 percent of the stated bandwidth. Full-duplex offers 100 percent efficiency in both directions (transmitting and receiving). This results in a 200 percent potential use of the stated bandwidth.

5.1.2.2 Configure Switch Ports at the Physical Layer

Duplex and Speed

Switch ports can be manually configured with specific duplex and speed settings. Use the duplex interface configuration mode command to manually specify the duplex mode for a switch port. Use the speed interface configuration mode command to manually specify the speed for a switch port. In Figure below, port F0/1 on switch S1 and S2 are manually configured with the full keyword for the duplex command, and the 100 keyword for the speed command.

Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface FastEthernet 0/1`
Configure the interface duplex.	`S1(config-if)# duplex full`
Configure the interface speed.	`S1(config-if)# speed 100`
Return to the privileged EXEC mode.	`S1(config-if)# end`
Save the running config to the startup config.	`S1# copy running-config startup-config`

The default setting for both duplex and speed for switch ports on Cisco Catalyst 2960 and 3560 switches is auto. The 10/100/1000 ports operate in either half- or full-duplex mode when they are set to 10 or 100 Mb/s, but when they are set to 1000 Mb/s (1 Gb/s), they operate only in full-duplex mode. Auto-negotiation is useful when the speed and duplex settings of the device connecting to the port are unknown or may change. When connecting to known devices, such as servers, dedicated workstations, or network devices, best practice is to manually set the speed and duplex settings.

When troubleshooting switch port issues, the duplex and speed settings should be checked.

Note: Mismatched settings for the duplex mode and speed of switch ports can cause connectivity issues. Auto-negotiation failure creates mismatched settings.

All fiber optic ports, such as 1000BASE-SX ports, operate only at one preset speed and are always full-duplex.

5.1.2.3 Auto-MDIX

Until recently, certain cable types (straight-through or crossover) were required when connecting devices. Switch-to-switch or switch-to-router connections required using different Ethernet cables. Using the automatic medium-dependent interface crossover (auto-MDIX) feature on an interface eliminates this problem. When auto-MDIX is enabled, the interface automatically detects the required cable connection type (straight-through or crossover) and configures the connection appropriately. When connecting to switches without the auto-MDIX feature, straight-through cables must be used to connect to devices such as servers, workstations, or routers. Crossover cables must be used to connect to other switches or repeaters.

With auto-MDIX enabled, either type of cable can be used to connect to other devices, and the interface automatically adjusts to communicate successfully. On newer Cisco switches, the **mdix auto** interface configuration mode command enables the feature. When using auto-MDIX on an interface, the interface speed and duplex must be set to **auto** so that the feature operates correctly.

The commands to enable auto-MDIX are shown in Figure below.

Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface fastethernet 0/1`
Configure the interface to autonegotiate duplex with the connected device.	`S1(config-if)# duplex auto`
Configure the interface to autonegotiate speed with the connected device.	`S1(config-if)# speed auto`
Enable auto-MDIX on the interface.	`S1(config-if)# mdix auto`
Return to the privileged EXEC mode.	`S1(config-if)# end`
Save the running config to the startup config.	`S1# copy running-config startup-config`

Note: The auto-MDIX feature is enabled by default on Catalyst 2960 and Catalyst 3560 switches, but is not available on the older Catalyst 2950 and Catalyst 3550 switches.

To examine the auto-MDIX setting for a specific interface, use the **show controllers ethernet-controller** command with the **phy** keyword. To limit the output to lines referencing auto-MDIX, use the **include Auto-MDIX** filter.

5.1.2.4 Verifying Switch Port Configuration

Figure below describes some of the options for the **show** command that are helpful in verifying common configurable switch features.

Display interface status and configuration.	`S1# show interfaces [interface-id]`
Display current startup configuration.	`S1# show startup-config`
Display current operating config.	`S1# show running-config`
Display information about flash file system.	`S1# show flash`
Display system hardware and software status.	`S1# show version`
Display history of commands entered.	`S1# show history`
Display IP information about an interface.	`S1# show ip [interface-id]`

Display the MAC address table.	`S1# show mac-address-table`
	OR
	`S1# show mac address-table`

The **show interfaces** command is another commonly used command, which displays status and statistics information on the network interfaces of the switch. The **show interfaces** command is frequently used when configuring and monitoring network devices.

5.1.2.5 Network Access Layer Issues

The output from the **show interfaces** command can be used to detect common media issues. One of the most important parts of this output is the display of the line and data link protocol status.

The first parameter (FastEthernet0/1 is up) refers to the hardware layer and indicates if the interface is receiving a carrier detect signal. The second parameter (line protocol is up) refers to the data link layer and indicates whether the data link layer protocol keepalives are being received.

Based on the output of the **show interfaces** command, possible problems can be fixed as follows:

- If the interface is up and the line protocol is down, a problem exists. There could be an encapsulation type mismatch, the interface on the other end could be error-disabled, or there could be a hardware problem.
- If the line protocol and the interface are both down, a cable is not attached or some other interface problem exists. For example, in a back-to-back connection, the other end of the connection may be administratively down.
- If the interface is administratively down, it has been manually disabled (the **shutdown** command has been issued) in the active configuration.

Some media errors are not severe enough to cause the circuit to fail, but do cause network performance issues.

“Input errors” is the sum of all errors in datagrams that were received on the interface being examined. This includes runts, giants, CRC, no buffer, frame, overrun, and ignored counts. The reported input errors from the **show interfaces** command include the following:

- **Runt Frames** - Ethernet frames that are shorter than the 64-byte minimum allowed length are called runts. Malfunctioning NICs are the usual cause of excessive runt frames, but they can also be caused by collisions.
- **Giants** - Ethernet frames that are larger than the maximum allowed size are called giants.
- **CRC errors** - On Ethernet and serial interfaces, CRC errors usually indicate a media or cable error. Common causes include electrical interference, loose or damaged connections, or incorrect

cabling. If you see many CRC errors, there is too much noise on the link and you should inspect the cable. You should also search for and eliminate noise sources.

"Output errors" is the sum of all errors that prevented the final transmission of datagrams out the interface that is being examined. The reported output errors from the **show interfaces** command include the following:

- **Collisions** - Collisions in half-duplex operations are normal. However, you should never see collisions on an interface configured for full-duplex communication.
- **Late collisions** - A late collision refers to a collision that occurs after 512 bits of the frame have been transmitted. Excessive cable lengths are the most common cause of late collisions. Another common cause is duplex misconfiguration. For example, you could have one end of a connection configured for full-duplex and the other for half-duplex. You would see late collisions on the interface that is configured for half-duplex. In that case, you must configure the same duplex setting on both ends. A properly designed and configured network should never have late collisions.

5.1.2.6 Troubleshooting Network Access Layer Issues

Most issues that affect a switched network are encountered during the original implementation. Theoretically, after it is installed, a network continues to operate without problems. However, cabling gets damaged, configurations change, and new devices are connected to the switch that require switch configuration changes. Ongoing maintenance and troubleshooting of the network infrastructure is required.

To troubleshoot scenarios involving no connection, or a bad connection, between a switch and another device, follow this general process:

Use the **show interfaces** command to check the interface status.

If the interface is down:

- Check to make sure that the proper cables are being used. Additionally, check the cable and connectors for damage. If a bad or incorrect cable is suspected, replace the cable.
- If the interface is still down, the problem may be due to a mismatch in speed setting. The speed of an interface is typically auto-negotiated; therefore, even if it is manually configured on one interface, the connecting interface should auto-negotiate accordingly. If a speed mismatch does occur through misconfiguration, or a hardware or software issue, then that may result in the interface going down. Manually set the same speed on both connection ends if a problem is suspected.

If the interface is up, but issues with connectivity are still present:

- Using the **show interfaces** command, check for indications of excessive noise. Indications may include an increase in the counters for runts, giants, and CRC errors. If there is excessive noise, first find and remove the source of the noise, if possible. Also, verify that the cable does not exceed the maximum cable length and check the type of cable that is used.
- If noise is not an issue, check for excessive collisions. If there are collisions or late collisions, verify the duplex settings on both ends of the connection. Much like the speed setting, the duplex setting is usually auto-negotiated. If there does appear to be a duplex mismatch, manually set the duplex to full on both ends of the connection.

5.2 Switch Security

5.2.1 Secure Remote Access

5.2.1.1 SSH Operation

Secure Shell (SSH) is a protocol that provides a secure (encrypted) management connection to a remote device. SSH should replace Telnet for management connections. Telnet is an older protocol that uses unsecure plaintext transmission of both the login authentication (username and password) and the data transmitted between the communicating devices. SSH provides security for remote connections by providing strong encryption when a device is authenticated (username and password) and also for the transmitted data between the communicating devices. SSH is assigned to TCP port 22. Telnet is assigned to TCP port 23.

To enable SSH on a Catalyst 2960 switch, the switch must be using a version of the IOS software including cryptographic (encrypted) features and capabilities. Use the **show version** command on the switch to see which IOS the switch is currently running.An IOS filename that includes the combination "k9" supports cryptographic (encrypted) features and capabilities.

5.2.1.2 Configuring SSH

Before configuring SSH, the switch must be minimally configured with a unique hostname and the correct network connectivity settings.

Step 1. Verify SSH support.

Use the **show ip ssh** command to verify that the switch supports SSH. If the switch is not running an IOS that supports cryptographic features, this command is unrecognized.

Step 2. Configure the IP domain.

Configure the IP domain name of the network using the **ip domain-name** *domain-name* global configuration mode command.

Step 3. Generate RSA key pairs.

Not all versions of the IOS default to SSH version 2, and SSH version 1 has known security flaws. To configure SSH version 2, issue the **ip ssh version 2** global configuration mode command. Generating an RSA key pair automatically enables SSH. Use the **crypto key generate rsa** global configuration mode command to enable the SSH server on the switch and generate an RSA key pair. When generating RSA keys, the administrator is prompted to enter a modulus length. A longer modulus length is more secure, but it takes longer to generate and to use.

Note: To delete the RSA key pair, use the **crypto key zeroize rsa** global configuration mode command. After the RSA key pair is deleted, the SSH server is automatically disabled.

Step 4. Configure user authentication.

The SSH server can authenticate users locally or using an authentication server. To use the local authentication method, create a username and password pair using the **username** *username* **secret** *password* global configuration mode command.

Step 5. Configure the vty lines.

Enable the SSH protocol on the vty lines using the **transport input ssh** line configuration mode command. The Catalyst 2960 has vty lines ranging from 0 to 15. This configuration prevents non-SSH (such as Telnet) connections and limits the switch to accept only SSH connections. Use the **line vty** global configuration mode command and then the **login local** line configuration mode command to require local authentication for SSH connections from the local username database.

Step 6. Enable SSH version 2.

By default, SSH supports both versions 1 and 2. When supporting both versions, this is shown in the **show ip ssh** output as supporting version 1.99. Version 1 has known vulnerabilities. For this reason, it is recommended to enable only version 2. Enable SSH version using the **ip ssh version 2** global configuration command.

5.2.1.3 Verifying SSH

On a PC, an SSH client such as PuTTY, is used to connect to an SSH server.

To display the version and configuration data for SSH on the device that you configured as an SSH server, use the **show ip ssh** command. In the example, SSH version 2 is enabled. To check the SSH connections to the device, use the **show ssh** command.

5.2.2 Switch Port Security

5.2.2.1 Secure Unused Ports

Disable Unused Ports

A simple method that many administrators use to help secure the network from unauthorized access is to disable all unused ports on a switch. For example, if a Catalyst 2960 switch has 24 ports and there are three Fast Ethernet connections in use, it is good practice to disable the 21 unused ports. Navigate to each unused port and issue the Cisco IOS shutdown command. If, later on, a port must be reactivated, it can be enabled with the no shutdown command. The figure shows partial output for this configuration.

It is simple to make configuration changes to multiple ports on a switch. If a range of ports must be configured, use the interface range command.

```
Switch(config)# interface range type module/first-number – last-number
```

The process of enabling and disabling ports can be time-consuming, but it enhances security on the network and is well worth the effort.

5.2.2.2 Port Security: Operation

Port Security

All switch ports (interfaces) should be secured before the switch is deployed for production use. One way to secure ports is by implementing a feature called port security. Port security limits the number of valid MAC addresses allowed on a port. The MAC addresses of legitimate devices are allowed access, while other MAC addresses are denied.

Port security can be configured to allow one or more MAC addresses. If the number of MAC addresses allowed on the port is limited to one, then only the device with that specific MAC address can successfully connect to the port.

If a port is configured as a secure port and the maximum number of MAC addresses is reached, any additional attempts to connect by unknown MAC addresses will generate a security violation.

Secure MAC Address Types

There are a number of ways to configure port security. The type of secure address is based on the configuration and includes:

- **Static secure MAC addresses** - MAC addresses that are manually configured on a port by using the **switchport port-security mac-address** *mac-address* interface configuration mode command. MAC addresses configured in this way are stored in the address table and are added to the running configuration on the switch.
- **Dynamic secure MAC addresses** - MAC addresses that are dynamically learned and stored only in the address table. MAC addresses configured in this way are removed when the switch restarts.

Sticky secure MAC addresses - MAC addresses that can be dynamically learned or manually configured, then stored in the address table and added to the running configuration.

Sticky Secure MAC addresses

To configure an interface to convert dynamically learned MAC addresses to sticky secure MAC addresses and add them to the running configuration, you must enable sticky learning. Sticky learning is enabled on an interface by using the **switchport port-security mac-address sticky** interface configuration mode command.

When this command is entered, the switch converts all dynamically learned MAC addresses, including those that were dynamically learned before sticky learning was enabled, into sticky secure MAC addresses. All sticky secure MAC addresses are added to the address table and to the running configuration.

Sticky secure MAC addresses can also be manually defined. When sticky secure MAC addresses are configured by using the **switchport port-security mac-address sticky** *mac-address* interface configuration mode command, all specified addresses are added to the address table and the running configuration.

If the sticky secure MAC addresses are saved to the startup configuration file, then when the switch restarts or the interface shuts down, the interface does not need to relearn the addresses. If the sticky secure addresses are not saved, they will be lost.

If sticky learning is disabled by using the **no switchport port-security mac-address sticky** interface configuration mode command, the sticky secure MAC addresses remain part of the address table, but are removed from the running configuration.

The following are the characteristics of sticky secure MAC addresses.

- Learned dynamically, converted to sticky secure MAC addresses stored in the running-config.
- Removed from the running-config if port security is disabled.
- Lost when the switch reboots (power cycled).
- Saving sticky secure MAC addresses in the startup-config makes them permanent and the switch retains them after a reboot.
- Disabling sticky learning converts sticky MAC addresses to dynamic secure addresses and removes them from the running-config.

Note: The port security feature will not work until port security is enabled on the interface using the **switchport port-security** command.

5.2.2.3 Port Security: Violation Modes

An interface can be configured for one of three violation modes, specifying the action to be taken if a violation occurs. The following presents which kinds of data traffic are forwarded when one of the following security violation modes are configured on a port:

- **Protect** - When the number of secure MAC addresses reaches the limit allowed on the port, packets with unknown source addresses are dropped until a sufficient number of secure MAC addresses are removed, or the number of maximum allowable addresses is increased. There is no notification that a security violation has occurred.
- **Restrict** - When the number of secure MAC addresses reaches the limit allowed on the port, packets with unknown source addresses are dropped until a sufficient number of secure MAC addresses are removed, or the number of maximum allowable addresses is increased. In this mode, there is a notification that a security violation has occurred.

- **Shutdown** - In this (default) mode, a port security violation causes the interface to immediately become error-disabled and turns off the port LED. It increments the violation counter. When a secure port is in the error-disabled state, it can be brought out of this state by entering the **shutdown** interface configuration mode command followed by the **no shutdown** command.

To change the violation mode on a switch port, use the **switchport port-security violation** {**protect** | **restrict** | **shutdown**} interface configuration mode command.

5.2.2.4 Port Security: Configuring

Figure below shows the Cisco IOS CLI commands needed to configure port security on the Fast Ethernet F0/18 port on the S1 switch. Notice that the example does not specify a violation mode. In this example, the violation mode is shutdown (the default mode).

Specify the interface to be configured for port security.	`S1(config)# interface fastethernet 0/18`
Set the interface mode to access.	`S1(config-if)# switchport mode access`
Enable port security on the interface.	`S1(config-if)# switchport port-security`

Figure below shows how to enable sticky secure MAC addresses for port security on Fast Ethernet port 0/19 of switch S1. As stated earlier, the maximum number of secure MAC addresses can be manually configured. In this example, the Cisco IOS command syntax is used to set the maximum number of MAC addresses to 10 for port 0/19. The violation mode is set to shutdown, by default.

Specify the interface to be configured for port security.	`S1(config)# interface fastethernet 0/19`
Set the interface mode to access.	`S1(config-if)# switchport mode access`
Enable port security on the interface.	`S1(config-if)# switchport port-security`
Set the maximum number of secure addresses allowed on the port.	`S1(config-if)# switchport port-security maximum 10`
Enable sticky learning.	`S1(config-if)# switchport port-security mac-address sticky`

5.2.2.5 Port Security: Verifying

Verify Port Security

After configuring port security on a switch, check each interface to verify that the port security is set correctly, and check to ensure that the static MAC addresses have been configured correctly.

Verify Port Security Settings

To display port security settings for the switch, or for the specified interface, use the **show port-security interface** [*interface-id*] command.

Note: The MAC address is identified as a sticky MAC.

Sticky MAC addresses are added to the MAC address table and to the running configuration.

Verify Secure MAC Addresses

To display all secure MAC addresses configured on all switch interfaces, or on a specified interface with aging information for each, use the **show port-security address** command.

5.2.2.6 Ports in Error Disabled State

When a port is configured with port security, a violation can cause the port to become error disabled. When a port is error disabled, it is effectively shut down and no traffic is sent or received on that port. A series of port security related messages display on the console.

Note: The port protocol and link status is changed to down.

The port LED will turn off. The **show interfaces** command identifies the port status as **err-disabled**. The output of the **show port-security interface** command now shows the port status as **secure-shutdown**. Because the port security violation mode is set to shutdown, the port with the security violation goes to the error disabled state.

The administrator should determine what caused the security violation before re-enabling the port. If an unauthorized device is connected to a secure port, the port should not be re-enabled until the security threat is eliminated. To re-enable the port, use the **shutdown** interface configuration mode command. Then, use the **no shutdown** interface configuration command to make the port operational.

CHAPTER 6: VLANS

Network performance is an important factor in the productivity of an organization. One of the technologies used to improve network performance is the separation of large broadcast domains into smaller ones. By design, routers will block broadcast traffic at an interface. However, routers normally have a limited number of LAN interfaces. A router's primary role is to move information between networks, not to provide network access to end devices.

The role of providing access into a LAN is normally reserved for an access layer switch. A virtual local area network (VLAN) can be created on a Layer 2 switch to reduce the size of broadcast domains, similar to a Layer 3 device. VLANs are commonly incorporated into network design making it easier for a network to support the goals of an organization. While VLANs are primarily used within switched local area networks, modern implementations of VLANs allow them to span MANs and WANs.

Because VLANs segment the network, a Layer 3 process is required to allow traffic to move from one network segment to another.

This Layer 3 routing process can either be implemented using a router or a Layer 3 switch interface. The use of a Layer 3 device provides a method for controlling the flow of traffic between network segments, including network segments created by VLANs.

6.1 VLAN Segmentation
6.1.1 Overview of VLANs

6.1.1.1 VLAN Definitions

Within a switched internetwork, VLANs provide segmentation and organizational flexibility. VLANs provide a way to group devices within a LAN. A group of devices within a VLAN communicate as if they were attached to the same cable. VLANs are based on logical connections, instead of physical connections.

VLANs allow an administrator to segment networks based on factors such as function, project team, or application, without regard for the physical location of the user or device. Devices within a VLAN act as if they are in their own independent network, even if they share a common infrastructure with other VLANs. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations within the VLAN where the packets are sourced. Each VLAN is considered a separate logical network. Packets destined for stations that do not belong to the VLAN must be forwarded through a device that supports routing.

Multiple IP subnets can exist on a switched network, without the use of multiple VLANs. However, the devices will be in the same Layer 2 broadcast domain. This means that any Layer 2 broadcasts, such as an ARP request, will be received by all devices on the switched network, even by those not intended to receive the broadcast.

A VLAN creates a logical broadcast domain that can span multiple physical LAN segments. VLANs improve network performance by separating large broadcast domains into smaller ones. If a device in one VLAN sends a broadcast Ethernet frame, all devices in the VLAN receive the frame, but devices in other VLANs do not.

VLANs enable the implementation of access and security policies according to specific groupings of users. Each switch port can be assigned to only one VLAN (with the exception of a port connected to an IP phone or to another switch).

6.1.1.2 Benefits of VLANs

User productivity and network adaptability are important for business growth and success. VLANs make it easier to design a network to support the goals of an organization. The primary benefits of using VLANs are as follows:

- **Security** - Groups that have sensitive data are separated from the rest of the network, decreasing the chances of confidential information breaches.
- **Cost reduction** - Cost savings result from reduced need for expensive network upgrades and more efficient use of existing bandwidth and uplinks.
- **Better performance** - Dividing flat Layer 2 networks into multiple logical workgroups (broadcast domains) reduces unnecessary traffic on the network and boosts performance.
- **Reduce the size of broadcast domains** - Dividing a network into VLANs reduces the number of devices in the broadcast domain.
- **Improved IT staff efficiency** - VLANs make it easier to manage the network because users with similar network requirements share the same VLAN. When a new switch is provisioned, all the policies and procedures already configured for the particular VLAN are implemented when the ports are assigned. It is also easy for the IT staff to identify the function of a VLAN by giving it an appropriate name.
- **Simpler project and application management** - VLANs aggregate users and network devices to support business or geographic requirements. Having separate functions makes managing a project or working with a specialized application easier; an example of such an application is an e-learning development platform for faculty.

Each VLAN in a switched network corresponds to an IP network. Therefore, VLAN design must take into consideration the implementation of a hierarchical network-addressing scheme. Hierarchical network addressing means that IP network numbers are applied to network segments or VLANs in an orderly fashion that takes the network as a whole into consideration. Blocks of contiguous network addresses are reserved for and configured on devices in a specific area of the network.

6.1.1.3 Types of VLANs

There are a number of distinct types of VLANs used in modern networks. Some VLAN types are defined by traffic classes. Other types of VLANs are defined by the specific function that they serve.

Data VLAN

A data VLAN is a VLAN that is configured to carry user-generated traffic. A VLAN carrying voice or management traffic would not be a data VLAN. It is common practice to separate voice and management traffic from data traffic. A data VLAN is sometimes referred to as a user VLAN. Data VLANs are used to separate the network into groups of users or devices.

Default VLAN

All switch ports become a part of the default VLAN after the initial boot up of a switch loading the default configuration. Switch ports that participate in the default VLAN are part of the same broadcast domain. This allows any device connected to any switch port to communicate with other devices on other switch ports. The default VLAN for Cisco switches is VLAN 1. Notice that all ports are assigned to VLAN 1 by default.

VLAN 1 has all the features of any VLAN, except it cannot be renamed or deleted. By default, all Layer 2 control traffic is associated with VLAN 1.

Native VLAN

A native VLAN is assigned to an 802.1Q trunk port. Trunk ports are the links between switches that support the transmission of traffic associated with more than one VLAN. An 802.1Q trunk port supports traffic coming from many VLANs (tagged traffic), as well as traffic that does not come from a VLAN (untagged traffic). Tagged traffic refers to traffic that has a 4-byte tag inserted within the original Ethernet frame header, specifying the VLAN to which the frame belongs. The 802.1Q trunk port places untagged traffic on the native VLAN, which by default is VLAN 1.

Native VLANs are defined in the IEEE 802.1Q specification to maintain backward compatibility with untagged traffic common to legacy LAN scenarios. A native VLAN serves as a common identifier on opposite ends of a trunk link.

It is a best practice to configure the native VLAN as an unused VLAN, distinct from VLAN 1 and other VLANs. In fact, it is not unusual to dedicate a fixed VLAN to serve the role of the native VLAN for all trunk ports in the switched domain.

Management VLAN

A management VLAN is any VLAN configured to access the management capabilities of a switch. VLAN 1 is the management VLAN by default. To create the management VLAN, the switch virtual interface (SVI) of that VLAN is assigned an IP address and a subnet mask, allowing the switch to be managed via HTTP, Telnet, SSH, or SNMP. Because the out-of-the-box configuration of a Cisco switch has VLAN 1 as the default VLAN, VLAN 1 would be a bad choice for the management VLAN.

In the past, the management VLAN for a 2960 switch was the only active SVI. On 15.x versions of the Cisco IOS for Catalyst 2960 Series switches, it is possible to have more than one active SVI. Cisco IOS 15.x requires that the particular active SVI assigned for remote management be documented. While theoretically a switch can have more than one management VLAN, having more than one increases exposure to network attacks.

6.1.1.4 Voice VLANs

A separate VLAN is needed to support Voice over IP (VoIP). VoIP traffic requires:

- Assured bandwidth to ensure voice quality
- Transmission priority over other types of network traffic
- Ability to be routed around congested areas on the network
- Delay of less than 150 ms across the network

To meet these requirements, the entire network has to be designed to support VoIP. The details of how to configure a network to support VoIP are beyond the scope of this course, but it is useful to summarize how a voice VLAN works between a switch, a Cisco IP phone, and a computer.

6.1.2 VLANs in a Multi-Switched Environment

6.1.2.1 VLAN Trunks

A trunk is a point-to-point link between two network devices that carries more than one VLAN. A VLAN trunk extends VLANs across an entire network. Cisco supports IEEE 802.1Q for coordinating trunks on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces.

VLANs would not be very useful without VLAN trunks. VLAN trunks allow all VLAN traffic to propagate between switches, so that devices which are in the same VLAN, but connected to different switches, can communicate without the intervention of a router.

A VLAN trunk does not belong to a specific VLAN; rather, it is a conduit for multiple VLANs between switches and routers. A trunk could also be used between a network device and server or other device that

is equipped with an appropriate 802.1Q-capable NIC. By default, on a Cisco Catalyst switch, all VLANs are supported on a trunk port.

6.1.2.2 Controlling Broadcast Domains with VLANs

Network without VLANs

In normal operation, when a switch receives a broadcast frame on one of its ports, it forwards the frame out all other ports except the port where the broadcast was received. In this example, all devices are on the same IPv4 subnet. If there were devices on other IPv4 subnets, they would also receive the same broadcast frame. Broadcasts such as an ARP request, are intended only for devices on the same subnet.

Network with VLANs

When VLANs are implemented on a switch, the transmission of unicast, multicast, and broadcast traffic from a host in a particular VLAN are restricted to the devices that are in that VLAN.

6.1.2.3 Tagging Ethernet Frames for VLAN Identification

Catalyst 2960 Series switches are Layer 2 devices. They use the Ethernet frame header information to forward packets. They do not have routing tables. The standard Ethernet frame header does not contain information about the VLAN to which the frame belongs; thus, when Ethernet frames are placed on a trunk, information about the VLANs to which they belong must be added. This process, called tagging, is accomplished by using the IEEE 802.1Q header, specified in the IEEE 802.1Q standard. The 802.1Q header includes a 4-byte tag inserted within the original Ethernet frame header, specifying the VLAN to which the frame belongs.

When the switch receives a frame on a port configured in access mode and assigned a VLAN, the switch inserts a VLAN tag in the frame header, recalculates the Frame Check Sequence (FCS), and sends the tagged frame out of a trunk port.

VLAN Tag Field Details

The VLAN tag field consists of a Type field, a Priority field, a Canonical Format Identifier field, and VLAN ID field:

- **Type** - A 2-byte value called the tag protocol ID (TPID) value. For Ethernet, it is set to hexadecimal 0x8100.
- **User priority** - A 3-bit value that supports level or service implementation.
- **Canonical Format Identifier (CFI)** - A 1-bit identifier that enables Token Ring frames to be carried across Ethernet links.
- **VLAN ID (VID)** - A 12-bit VLAN identification number that supports up to 4096 VLAN IDs.

After the switch inserts the Type and tag control information fields, it recalculates the FCS values and inserts the new FCS into the frame.

6.1.2.4 Native VLANs and 802.1Q Tagging

Tagged Frames on the Native VLAN

Some devices that support trunking add a VLAN tag to native VLAN traffic. Control traffic sent on the native VLAN should not be tagged. If an 802.1Q trunk port receives a tagged frame with the VLAN ID that is the same as the native VLAN, it drops the frame. Consequently, when configuring a switch port on a Cisco switch, configure devices so that they do not send tagged frames on the native VLAN. Devices from other vendors that support tagged frames on the native VLAN include IP phones, servers, routers, and non-Cisco switches.

Untagged Frames on the Native VLAN

When a Cisco switch trunk port receives untagged frames (which are unusual in a well-designed network), it forwards those frames to the native VLAN. If there are no devices associated with the native VLAN (which is not unusual) and there are no other trunk ports (which is not unusual), then the frame is dropped. The default native VLAN is VLAN 1. When configuring an 802.1Q trunk port, a default Port VLAN ID (PVID) is assigned the value of the native VLAN ID. All untagged traffic coming in or out of the 802.1Q port is forwarded based on the PVID value. For example, if VLAN 99 is configured as the native VLAN, the PVID is 99 and all untagged traffic is forwarded to VLAN 99. If the native VLAN has not been reconfigured, the PVID value is set to VLAN 1.

6.1.2.5 Voice VLAN Tagging

Recall that to support VoIP, a separate voice VLAN is required.

An access port that is used to connect a Cisco IP phone can be configured to use two separate VLANs: one VLAN for voice traffic and another VLAN for data traffic from a device attached to the phone. The link between the switch and the IP phone acts as a trunk to carry both voice VLAN traffic and data VLAN traffic.

The Cisco IP Phone contains an integrated three-port 10/100 switch. The ports provide dedicated connections to these devices:

- Port 1 connects to the switch or other VoIP device.
- Port 2 is an internal 10/100 interface that carries the IP phone traffic.
- Port 3 (access port) connects to a PC or other device.

On the switch, the access is configured to send Cisco Discovery Protocol (CDP) packets that instruct an attached IP phone to send voice traffic to the switch in one of three ways, depending on the type of traffic:

- In a voice VLAN tagged with a Layer 2 class of service (CoS) priority value
- In an access VLAN tagged with a Layer 2 CoS priority value
- In an access VLAN, untagged (no Layer 2 CoS priority value)

6.2 VLAN Implementations

6.2.1 VLAN Assignment

6.2.1.1 VLAN Ranges on Catalyst Switches

Different Cisco Catalyst switches support various numbers of VLANs. The number of supported VLANs is large enough to accommodate the needs of most organizations. For example, the Catalyst 2960 and 3560 Series switches support over 4,000 VLANs. Normal range VLANs on these switches are numbered 1 to 1,005 and extended range VLANs are numbered 1,006 to 4,094.

Normal Range VLANs

- Used in small- and medium-sized business and enterprise networks.
- Identified by a VLAN ID between 1 and 1005.
- IDs 1002 through 1005 are reserved for Token Ring and Fiber Distributed Data Interface (FDDI) VLANs.
- IDs 1 and 1002 to 1005 are automatically created and cannot be removed.
- Configurations are stored within a VLAN database file, called vlan.dat. The vlan.dat file is located in the flash memory of the switch.
- The VLAN Trunking Protocol (VTP), which helps manage VLAN configurations between switches, can only learn and store normal range VLANs.

Extended Range VLANs

- Enable service providers to extend their infrastructure to a greater number of customers. Some global enterprises could be large enough to need extended range VLAN IDs.

- Are identified by a VLAN ID between 1006 and 4094.
- Configurations are not written to the vlan.dat file.
- Support fewer VLAN features than normal range VLANs.
- Saved, by default, in the running configuration file.
- VTP does not learn extended range VLANs.

Note: 4096 is the upper boundary for the number of VLANs available on Catalyst switches, because there are 12 bits in the VLAN ID field of the IEEE 802.1Q header.

6.2.1.2 Creating a VLAN

When configuring normal range VLANs, the configuration details are stored in flash memory on the switch, in a file called vlan.dat. Flash memory is persistent and does not require the **copy running-config startup-config** command. However, because other details are often configured on a Cisco switch at the same time that VLANs are created, it is good practice to save running configuration changes to the startup configuration.

Figure below displays the Cisco IOS command syntax used to add a VLAN to a switch and give it a name. Naming each VLAN is considered a best practice in switch configuration.

Enter global configuration mode.	`S1# configure terminal`
Create a VLAN with a valid id number.	`S1(config)# vlan vlan-id`
Specify a unique name to identify the VLAN.	`S1(config-vlan)# name vlan-name`
Return to the privileged EXEC mode.	`S1(config-vlan)# end`

In addition to entering a single VLAN ID, a series of VLAN IDs can be entered separated by commas, or a range of VLAN IDs separated by hyphens using the **vlan** *vlan-id* command. For example, use the following command to create VLANs 100, 102, 105, 106, and 107:

```
S1(config)# vlan 100,102,105-107
```

6.2.1.3 Assigning Ports to VLANs

After creating a VLAN, the next step is to assign ports to the VLAN. An access port can belong to only one VLAN at a time. One exception to this rule is that of a port connected to an IP phone, in which case, there are two VLANs associated with the port: one for voice and one for data.

Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Set the port to access mode.	`S1(config-if)# switchport mode access`
Assign the port to a VLAN.	`S1(config-if)# switchport access vlan vlan_id`

Figure above displays the syntax for defining a port to be an access port and assigning it to a VLAN. The **switchport mode access** command is optional, but strongly recommended as a security best practice. With this command, the interface changes to permanent access mode.

Note: Use the **interface range** command to simultaneously configure multiple interfaces.

The **switchport access vlan** command forces the creation of a VLAN if it does not already exist on the switch. For example, VLAN 30 is not present in the **show vlan brief** output of the switch. If the **switchport access vlan 30** command is entered on any interface with no previous configuration, then the switch displays the following:

% Access VLAN does not exist. Creating vlan 30

6.2.1.4 Changing VLAN Port Membership

There are a number of ways to change VLAN port membership with the **no switchport access vlan** interface configuration mode command.

Interface F0/18 was previously assigned to VLAN 20. The **no switchport access vlan** command is entered for interface F0/18. The **show vlan brief** command displays the VLAN assignment and membership type for all switch ports. The **show vlan brief** command displays one line for each VLAN. The output for each VLAN includes the VLAN name, status, and switch ports.

A port can easily have its VLAN membership changed. It is not necessary to first remove a port from a VLAN to change its VLAN membership. When an access port has its VLAN membership reassigned to another existing VLAN, the new VLAN membership simply replaces the previous VLAN membership.

6.2.1.5 Deleting VLANs

The **no vlan** *vlan-id* global configuration mode command is used to remove VLAN 20 from the switch. Switch S1 had a minimal configuration with all ports in VLAN 1 and an unused VLAN 20 in the VLAN database. The **show vlan brief** command verifies that VLAN 20 is no longer present in the vlan.dat file after using the **no vlan 20** command.

Caution: Before deleting a VLAN, reassign all member ports to a different VLAN first. Any ports that are not moved to an active VLAN are unable to communicate with other hosts after the VLAN is deleted and until they are assigned to an active VLAN.

Alternatively, the entire vlan.dat file can be deleted using the **delete flash:vlan.dat** privileged EXEC mode command. The abbreviated command version (**delete vlan.dat**) can be used if the vlan.dat file has not been moved from its default location. After issuing this command and reloading the switch, the previously configured VLANs are no longer present. This effectively places the switch into its factory default condition with regard to VLAN configurations.

Note: For a Catalyst switch, the **erase startup-config** command must accompany the **delete vlan.dat** command prior to reload to restore the switch to its factory default condition.

6.2.1.6 Verifying VLAN Information

After a VLAN is configured, VLAN configurations can be validated using Cisco IOS show commands.

You can use the **show vlan** and **show interfaces** command options to verify **vlan** and port assignment.

The **show vlan summary** command displays the count of all configured VLANs.

6.2.2 VLAN Trunks

6.2.2.1 Configuring IEEE 802.1Q Trunk Links

A VLAN trunk is an OSI Layer 2 link between two switches that carries traffic for all VLANs (unless the allowed VLAN list is restricted manually or dynamically). To enable trunk links, configure the ports on either end of the physical link with parallel sets of commands.

To configure a switch port on one end of a trunk link, use the **switchport mode trunk** command. With this command, the interface changes to permanent trunking mode. The port enters into a Dynamic Trunking Protocol (DTP) negotiation to convert the link into a trunk link even if the interface connecting to it does not agree to the change. In this course, the **switchport mode trunk** command is the only method implemented for trunk configuration.

Note: DTP is beyond the scope of this course.

The Cisco IOS command syntax to specify a native VLAN (other than VLAN 1) is shown in Figure below. In the example, VLAN 99 is configured as the native VLAN using the **switchport trunk native vlan 99** command.

Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Force the link to be a trunk link.	`S1(config-if)# switchport mode trunk`
Specify a native VLAN for untagged frames.	`S1(config-if)# switchport trunk native vlan vlan_id`
Specify the list of VLANs to be allowed on the trunk link.	`S1(config-if)# switchport trunk allowed vlan vlan-list`
Return to the privileged EXEC mode.	`S1(config-if)# end`

Use the Cisco IOS **switchport trunk allowed vlan** *vlan-list* command to specify the list of VLANs to be allowed on the trunk link.

Note: This configuration assumes the use of Cisco Catalyst 2960 switches which automatically use 802.1Q encapsulation on trunk links. Other switches may require manual configuration of the encapsulation. Always configure both ends of a trunk link with the same native VLAN. If 802.1Q trunk configuration is not the same on both ends, Cisco IOS Software reports errors.

6.2.2.2 Resetting the Trunk to Default State

Figure below shows the commands to remove the allowed VLANs and reset the native VLAN of the trunk. When reset to the default state, the trunk allows all VLANs and uses VLAN 1 as the native VLAN.

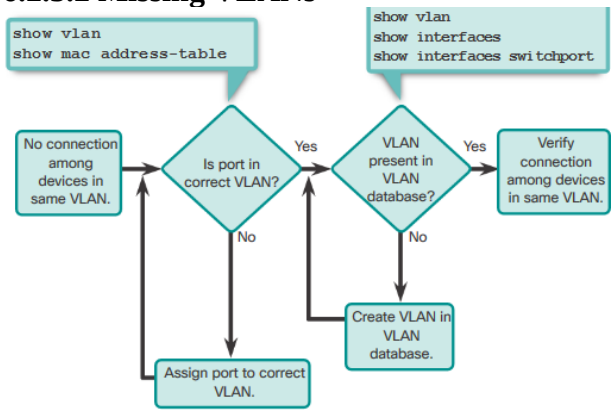
Enter global configuration mode.	`S1# configure terminal`
Enter interface configuration mode.	`S1(config)# interface interface_id`
Set trunk to allow all VLANs.	`S1(config-if)# no switchport trunk allowed vlan`
Reset native VLAN to default.	`S1(config-if)# no switchport trunk native vlan`
Return to the privileged EXEC mode.	`S1(config-if)# end`

6.2.3 Troubleshoot VLANs and Trunks

6.2.3.1 IP Addressing Issues with VLAN

Each VLAN must correspond to a unique IP subnet. If two devices in the same VLAN have different subnet addresses, they cannot communicate. This is a common problem, and it is easy to solve by identifying the incorrect configuration and changing the subnet address to the correct one.

6.2.3.2 Missing VLANs



If there is still no connection between devices in a VLAN, but IP addressing issues have been ruled out, refer to the flowchart in Figure above to troubleshoot:

Step 1. Use the **show vlan** command to check whether the port belongs to the expected VLAN. If the port is assigned to the wrong VLAN, use the **switchport access vlan** command to correct the VLAN membership. Use the **show mac address-table** command to check which addresses were learned on a particular port of the switch, and to which VLAN that port is assigned, as show in Figure 2.

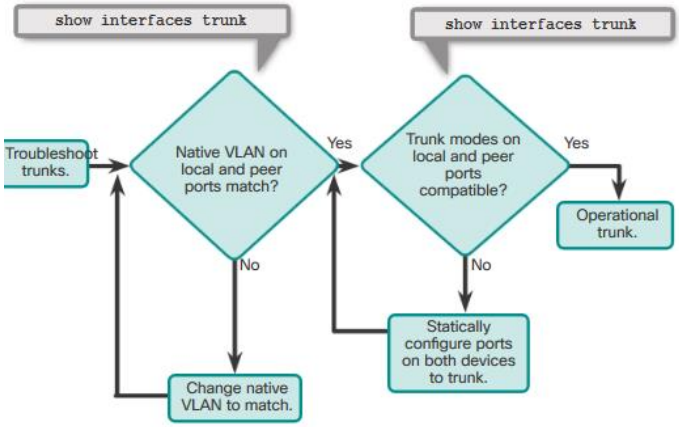
Step 2. If the VLAN to which the port is assigned is deleted, the port becomes inactive. The ports of a deleted VLAN will not be listed in the output of the **show vlan** command. Use the **show interfaces switchport** command to verify the inactive VLAN is assigned to the port, as shown in Figure 2.

Each port in a switch belongs to a VLAN. If the VLAN to which the port belongs is deleted, the port becomes inactive. All ports belonging to the VLAN that was deleted are unable to communicate with the rest of the network. Use the **show interface f0/1 switchport** command to check whether the port is inactive. If the port is inactive, it is not functional until the missing VLAN is created using the **vlan** *vlan-id* global configuration command or the VLAN is removed from the port with the **no switchport access vlan** *vlan-id* command.

6.2.3.3 Introduction to Troubleshooting Trunks

A common task of a network administrator is to troubleshoot trunk formation, or ports incorrectly behaving as trunk ports. Sometimes a switch port may behave like a trunk port even if it is not configured as a trunk port. For example, an access port might accept frames from VLANs different from the VLAN to which it is assigned. This is called VLAN leaking.

Figure below displays a flowchart of general trunk troubleshooting guidelines.



To troubleshoot issues when a trunk is not forming or when VLAN leaking is occurring, proceed as follows:

Step 1. Use the **show interfaces trunk** command to check whether the local and peer native VLANs match. If the native VLAN does not match on both sides, VLAN leaking occurs.

Step 2. Use the **show interfaces trunk** command to check whether a trunk has been established between switches. Statically configure trunk links whenever possible. Cisco Catalyst switch ports use DTP by default and attempt to negotiate a trunk link.

To display the status of the trunk, the native VLAN used on that trunk link, and verify trunk establishment, use the **show interfaces trunk** command.

CDP displays a notification of a native VLAN mismatch on a trunk link with this message:

*Mar 1 06:45:26.232: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/1 (2), with S2 FastEthernet0/1 (99).

Connectivity issues occur in the network if a native VLAN mismatch exists. Data traffic for VLANs, other than the two native VLANs configured, successfully propagates across the trunk link, but data associated with either of the native VLANs does not successfully propagate across the trunk link.

6.2.3.4 Common Problems with Trunks

Trunking issues are usually associated with incorrect configurations. When configuring VLANs and trunks on a switched infrastructure, the following types of configuration errors are the most common:

- **Native VLAN mismatches** - Trunk ports are configured with different native VLANs. This configuration error generates console notifications, and can cause inter-VLAN routing issues, among other problems. This poses a security risk.
- **Trunk mode mismatches** - One trunk port is configured in a mode that is not compatible for trunking on the corresponding peer port. This configuration error causes the trunk link to stop working. Be sure both sides of the trunk are configured with the **switchport mode trunk** command. Other trunk configuration commands are beyond the scope of this course.
- **Allowed VLANs on trunks** - The list of allowed VLANs on a trunk has not been updated with the current VLAN trunking requirements. In this situation, unexpected traffic (or no traffic) is being sent over the trunk.

If an issue with a trunk is discovered and if the cause is unknown, start troubleshooting by examining the trunks for a native VLAN mismatch. If that is not the cause, check for trunk mode mismatches, and finally check for the allowed VLAN list on the trunk. The next two pages examine how to fix the common problems with trunks.

6.2.3.5 Incorrect Port Mode

Trunk links are normally configured statically with the **switchport mode trunk** command. Cisco Catalyst switch trunk ports use DTP to negotiate the state of the link. When a port on a trunk link is configured with a trunk mode that is incompatible with the neighboring trunk port, a trunk link fails to form between the two switches.

6.2.3.6 Incorrect Port Mode

Trunk links are normally configured statically with the **switchport mode trunk** command. Cisco Catalyst switch trunk ports use DTP to negotiate the state of the link. When a port on a trunk link is configured with a trunk mode that is incompatible with the neighboring trunk port, a trunk link fails to form between the two switches.

6.2.3.6 Incorrect VLAN List

For traffic from a VLAN to be transmitted across a trunk, it must be allowed on the trunk. To do so, use the **switchport trunk allowed vlan***vlan-id* command.

6.3 Inter-VLAN Routing Using Routers

6.3.1 Inter-VLAN Routing Operation

6.3.1.1 What is Inter-VLAN Routing?

VLANs are used to segment switched networks. Layer 2 switches, such as the Catalyst 2960 Series, can be configured with over 4,000 VLANs. A VLAN is a broadcast domain, so computers on separate VLANs are unable to communicate without the intervention of a routing device. Layer 2 switches have very limited IPv4 and IPv6 functionality and cannot perform the dynamic routing function of routers. While Layer 2 switches are gaining more IP functionality, such as the ability to perform static routing, this is insufficient to handle these large number of VLANs.

Any device that supports Layer 3 routing, such as a router or a multilayer switch, can be used to perform the necessary routing functionality. Regardless of the device used, the process of forwarding network traffic from one VLAN to another VLAN using routing is known as inter-VLAN routing.

There are three options for inter-VLAN routing :

- Legacy inter-VLAN routing
- Router-on-a-Stick
- Layer 3 switching using SVIs

Note: This chapter focuses on the first two options. Layer 3 switching using SVIs is beyond the scope of this course.

6.3.1.2 Legacy Inter-VLAN Routing

Historically, the first solution for inter-VLAN routing relied on routers with multiple physical interfaces. Each interface had to be connected to a separate network and configured with a distinct subnet.

In this legacy approach, inter-VLAN routing is performed by connecting different physical router interfaces to different physical switch ports. The switch ports connected to the router are placed in access mode and each physical interface is assigned to a different VLAN. Each router interface can then accept traffic from the VLAN associated with the switch interface that it is connected to, and traffic can be routed to the other VLANs connected to the other interfaces.

Note: This method of inter-VLAN routing is not efficient and is generally no longer implemented in switched networks. It is shown in this course for explanation purposes only.

6.3.1.3 Router-on-a-Stick Inter-VLAN Routing

While legacy inter-VLAN routing requires multiple physical interfaces on both the router and the switch, a more common, present-day implementation of inter-VLAN routing does not. Instead, some router software permits configuring a router interface as a trunk link, meaning only one physical interface is required on the router and the switch to route packets between multiple VLANs.

'Router-on-a-stick' is a type of router configuration in which a single physical interface routes traffic between multiple VLANs on a network.

The router interface is configured to operate as a trunk link and is connected to a switch port that is configured in trunk mode. The router performs inter-VLAN routing by accepting VLAN-tagged traffic on the trunk interface coming from the adjacent switch, and then, internally routing between the VLANs using subinterfaces. The router then forwards the routed traffic, VLAN-tagged for the destination VLAN, out the same physical interface as it used to receive the traffic.

Subinterfaces are software-based virtual interfaces, associated with a single physical interface. Subinterfaces are configured in software on a router and each subinterface is independently configured with an IP address and VLAN assignment. Subinterfaces are configured for different subnets corresponding to their VLAN assignment to facilitate logical routing. After a routing decision is made based on the destination VLAN, the data frames are VLAN-tagged and sent back out the physical interface.

Note: The router-on-a-stick method of inter-VLAN routing does not scale beyond 50 VLANs.

6.3.2 Configure Legacy Inter-VLAN Routing

6.3.2.1 Configure Legacy Inter-VLAN Routing: Preparation

Legacy inter-VLAN routing requires routers to have multiple physical interfaces. The router accomplishes the routing by having each of its physical interfaces connected to a unique VLAN. Each interface is also configured with an IPv4 address for the subnet associated with the particular VLAN to which it is

connected. By configuring the IPv4 addresses on the physical interfaces, network devices connected to each of the VLANs can communicate with the router using the physical interface connected to the same VLAN. In this configuration, network devices can use the router as a gateway to access the devices connected to the other VLANs.

The routing process requires the source device to determine if the destination device is local or remote to the local subnet. The source device accomplishes this by comparing the source and destination IPv4 addresses against the subnet mask. When the destination IPv4 address has been determined to be on a remote network, the source device must identify where it needs to forward the packet to reach the destination device. The source device examines the local routing table to determine where it needs to send the data. Devices use their default gateway as the Layer 2 destination for all traffic that must leave the local subnet. The default gateway is the route that the device uses when it has no other explicitly defined route to the destination network. The IPv4 address of the router interface on the local subnet acts as the default gateway for the sending device.

When the source device has determined that the packet must travel through the local router interface on the connected VLAN, the source device sends out an ARP request to determine the MAC address of the local router interface. When the router sends its ARP reply back to the source device, the source device can use the MAC address to finish framing the packet before it sends it out on the network as unicast traffic.

Because the Ethernet frame has the destination MAC address of the router interface, the switch knows exactly which switch port to forward the unicast traffic out of to reach the router interface for that VLAN. When the frame arrives at the router, the router removes the source and destination MAC address information to examine the destination IPv4 address of the packet. The router compares the destination address to entries in its routing table to determine where it needs to forward the data to reach its final destination. If the router determines that the destination network is a locally connected network, as is the case with inter-VLAN routing, the router sends an ARP request out the interface that is physically connected to the destination VLAN. The destination device responds back to the router with its MAC address, which the router then uses to frame the packet. The router then sends the unicast traffic to the switch, which forwards it out the port where the destination device is connected.

Even though there are many steps in the process of inter-VLAN routing, when two devices on different VLANs communicate through a router, the entire process happens in a fraction of a second.

6.3.2.2 Configure Legacy Inter-VLAN Routing: Switch Configuration

To configure legacy inter-VLAN routing, start by configuring the switch.

Use the **vlan** *vlan_id* global configuration mode command to create VLANs. In this example, VLANs 10 and 30 were created on switch S1.

After the VLANs have been created, the switch ports are assigned to the appropriate VLANs. The **switchport access vlan** *vlan_id* command is executed from interface configuration mode on the switch for each interface to which the router connects.

In this example, interfaces F0/4 and F0/11 have been assigned to VLAN 10 using the **switchport access vlan 10** command. The same process is used to assign interface F0/5 and F0/6 on switch S1 to VLAN 30.

Finally, to protect the configuration so that it is not lost after a reload of the switch, the **copy running-config startup-config** command is executed to back up the running configuration to the startup configuration.

6.3.2.3 Configure Legacy Inter-VLAN Routing: Router Interface Configuration

Now the router can be configured to perform inter-VLAN routing.

Router interfaces are configured in a manner similar to configuring VLAN interfaces on switches. To configure a specific interface, change to interface configuration mode from global configuration mode.

Router interfaces are disabled by default and must be enabled using the **no shutdown** command before they are used. After the **no shutdown** interface configuration mode command has been issued, a notification displays, indicating that the interface state has changed to up. This indicates that the interface is now enabled.

The process is repeated for all router interfaces. Each router interface must be assigned to a unique subnet for routing to occur.

After the IPv4 addresses are assigned to the physical interfaces and the interfaces are enabled, the router is capable of performing inter-VLAN routing.

Examine the routing table using the **show ip route** command.

Notice the letter **C** to the left of each of the route entries for the VLANs. This letter indicates that the route is local for a connected interface, which is also identified in the route entry.

6.3.3 Configure Router-on-a-Stick Inter-VLAN Routing

6.3.3.1 Configure Router-on-a-Stick: Preparation

Legacy inter-VLAN routing using physical interfaces has a significant limitation. Routers have a limited number of physical interfaces to connect to different VLANs. As the number of VLANs increases on a network, having one physical router interface per VLAN quickly exhausts the physical interface capacity of a router. An alternative in larger networks is to use VLAN trunking and subinterfaces. VLAN trunking allows a single physical router interface to route traffic for multiple VLANs. This technique is termed router-on-a-stick and uses virtual subinterfaces on the router to overcome the hardware limitations based on physical router interfaces.

Subinterfaces are software-based virtual interfaces that are assigned to physical interfaces. Each subinterface is configured independently with its own IP address and prefix length. This allows a single physical interface to simultaneously be part of multiple logical networks.

Note: The term prefix length can be used to refer to the IPv4 subnet mask when associated with an IPv4 address, and the IPv6 prefix length when associated with an IPv6 address.

When configuring inter-VLAN routing using the router-on-a-stick model, the physical interface of the router must be connected to a trunk link on the adjacent switch. On the router, subinterfaces are created for each unique VLAN on the network. Each subinterface is assigned an IP address specific to its subnet/VLAN and is also configured to tag frames for that VLAN. This way, the router can keep the traffic from each subinterface separate as it traverses the trunk link back to the switch.

Functionally, the router-on-a-stick model is the same as using the legacy inter-VLAN routing model, but instead of using the physical interfaces to perform the routing, subinterfaces of a single physical interface are used.

Using trunk links and subinterfaces decreases the number of router and switch ports used. Not only can this save money, it can also reduce configuration complexity. Consequently, the router subinterface approach can scale to a much larger number of VLANs than a configuration with one physical interface per VLAN design.

6.3.3.2 Configure Router-on-a-Stick: Switch Configuration

To enable inter-VLAN routing using router-on-a stick, start by enabling trunking on the switch port that is connected to the router.

Router R1 is connected to switch S1 on trunk port F0/5. VLANs 10 and 30 are added to switch S1.

Because switch port F0/5 is configured as a trunk port, the port does not need to be assigned to any VLAN. To configure switch port F0/5 as a trunk port, execute the**switchport mode trunk**command in interface configuration mode for port F0/5.

The router can now be configured to perform inter-VLAN routing.

6.3.3.3 Configure Router-on-a-Stick: Router Subinterface Configuration

The configuration of the router is different when a router-on-a-stick configuration is used, compared to legacy inter-VLAN routing.

Each subinterface is created using the**interface** *interface_id subinterface_id* global configuration mode command. The syntax for the subinterface is the physical interface, in this case g0/0, followed by a period and a subinterface number. The subinterface number is typically configured to reflect the VLAN number.

Before assigning an IP address to a subinterface, the subinterface must be configured to operate on a specific VLAN using the **encapsulation dot1q** *vlan_id* command. In this example, subinterface G0/0.10 is assigned to VLAN 10.

Note: There is a**native**keyword option that can be appended to this command to set the IEEE 802.1Q native VLAN. In this example, the**native**keyword option was excluded to leave the native VLAN default as VLAN 1.

Next, assign the IPv4 address for the subinterface using the**ip address** *ip_address subnet_mask* subinterface configuration mode command. In this example, subinterface G0/0.10 is assigned the IPv4 address 172.17.10.1 using the **ip address 172.17.10.1 255.255.255.0** command.

This process is repeated for all router subinterfaces required to route between the VLANs configured on the network. Each router subinterface must be assigned an IP address on a unique subnet for routing to occur. For example, the other router subinterface, G0/0.30, is configured to use IPv4 address 172.17.30.1, which is on a different subnet from subinterface G0/0.10.

After a physical interface is enabled, subinterfaces will automatically be enabled upon configuration. Subinterfaces do not need to be enabled with the **no shutdown** command at the subinterface configuration mode level of the Cisco IOS software.

If the physical interface is disabled, all subinterfaces are disabled. In this example, the command **no shutdown** is entered in interface configuration mode for interface G0/0, which in turn, enables all of the configured subinterfaces.

Individual subinterfaces can be administratively shut down with the **shutdown** command. Also, individual subinterfaces can be enabled independently with the **no shutdown** command in the subinterface configuration mode.

6.3.3.4 Configure Router-on-a-Stick: Verifying Subinterfaces

By default, Cisco routers are configured to route traffic between local subinterfaces. As a result, routing does not specifically need to be enabled.

The **show vlan** command displays information about the Cisco IOS VLAN subinterfaces.

6.3.3.5 Configure Router-on-a-Stick: Verifying Routing

After the router and switch have been configured to perform inter-VLAN routing, the next step is to verify host-to-host connectivity. Access to devices on remote VLANs can be tested using the **ping** command.

Ping Test

The **ping** command sends an ICMP echo request to the destination address. When a host receives an ICMP echo request, it responds with an ICMP echo reply to confirm that it received the ICMP echo request. The **ping** command calculates the elapsed time using the difference between the time the echo request was sent and the time the echo reply was received. This elapsed time is used to determine the latency of the connection. Successfully receiving a reply confirms that there is a path between the sending device and the receiving device.

Tracert Test

Tracert is a useful utility for confirming the routed path taken between two devices. On UNIX systems, the utility is specified by **traceroute.** Tracert also uses ICMP to determine the path taken, but it uses ICMP echo requests with specific time-to-live values defined on the frame.

The time-to-live value determines exactly how many router hops away the ICMP echo is allowed to reach. The first ICMP echo request is sent with a time-to-live value set to expire at the first router on route to the destination device.

When the ICMP echo request times out on the first route, an ICMP message is sent back from the router to the originating device. The device records the response from the router and proceeds to send out another ICMP echo request, but this time with a greater time-to-live value. This allows the ICMP echo request to traverse the first router and reach the second device on route to the final destination. The process repeats recursively until finally the ICMP echo request is sent all the way to the final destination device. After the **tracert** utility finishes running, it displays a list of ingress router interfaces that the ICMP echo request reached on its way to the destination.

CHAPTER 7: ACCESS CONTROL LISTS

One of the most important skills a network administrator needs is mastery of access control lists (ACLs). ACLs provide security for a network.

Network designers use firewalls to protect networks from unauthorized use. Firewalls are hardware or software solutions that enforce network security policies. Consider a lock on a door to a room inside a building. The lock allows only authorized users with a key or access card to pass through the door. Similarly, a firewall filters unauthorized or potentially dangerous packets from entering the network.

On a Cisco router, you can configure a simple firewall that provides basic traffic filtering capabilities using ACLs. Administrators use ACLs to stop traffic or permit only specified traffic on their networks.

7.1 ACL Operation

7.1.1 Purpose of ACLs

7.1.1.1 What is an ACL?

An ACL is a series of IOS commands that control whether a router forwards or drops packets based on information found in the packet header. ACLs are among the most commonly used features of Cisco IOS software.

When configured, ACLs perform the following tasks:

- Limit network traffic to increase network performance. For example, if corporate policy does not allow video traffic on the network, ACLs that block video traffic could be configured and applied. This would greatly reduce the network load and increase network performance.
- Provide traffic flow control. ACLs can restrict the delivery of routing updates to ensure that the updates are from a known source.
- Provide a basic level of security for network access. ACLs can allow one host to access a part of the network and prevent another host from accessing the same area. For example, access to the Human Resources network can be restricted to authorized users.
- Filter traffic based on traffic type. For example, an ACL can permit email traffic, but block all Telnet traffic.
- Screen hosts to permit or deny access to network services. ACLs can permit or deny a user to access file types, such as FTP or HTTP.

By default, a router does not have ACLs configured; therefore, by default a router does not filter traffic. Traffic that enters the router is routed solely based on information within the routing table. However, when an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded.

In addition to either permitting or denying traffic, ACLs can be used for selecting types of traffic to be analyzed, forwarded, or processed in other ways. For example, ACLs can be used to classify traffic to enable priority processing. This capability is similar to having a VIP pass at a concert or sporting event. The VIP pass gives selected guests privileges not offered to general admission ticket holders, such as priority entry or being able to enter a restricted area.

7.1.1.2 Packet Filtering

An ACL is a sequential list of permit or deny statements, known as access control entries (ACEs). ACEs are also commonly called ACL statements. When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering.

Packet filtering controls access to a network by analyzing the incoming and outgoing packets and forwarding them or discarding them based on given criteria. Packet filtering can occur at Layer 3 or Layer 4. Standard ACLs only filter at Layer 3. Extended ACLs filter at Layer 3 and Layer 4.

The source IPv4 address is the filtering criteria set in each ACE of a standard IPv4 ACL. A router configured with a standard IPv4 ACL extracts the source IPv4 address from the packet header. The router starts at the top of the ACL and compares the address to each ACE sequentially. When a match is made, the router carries out the instruction, either permitting or denying the packet. After a match is made, the remaining ACEs in the ACL, if any, are not analyzed. If the source IPv4 address does not match any ACEs in the ACL, the packet is discarded.

The last statement of an ACL is always an implicit deny. This statement is automatically inserted at the end of each ACL even though it is not physically present. The implicit deny blocks all traffic. Because of this implicit deny, an ACL that does not have at least one permit statement will block all traffic.

7.1.1.3 ACL Operation

ACLs define the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router, and packets that exit outbound interfaces of the router. ACLs do not act on packets that originate from the router itself.

ACLs can be configured to apply to inbound traffic and outbound traffic:

- **Inbound ACLs** - Incoming packets are processed before they are routed to the outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet is discarded. If the packet is permitted by the ACL, it is then processed for routing. Inbound ACLs are best used to filter packets when the network attached to an inbound interface is the only source of packets that need to be examined.
- **Outbound ACLs** - Incoming packets are routed to the outbound interface, and then they are processed through the outbound ACL. Outbound ACLs are best used when the same filter will be applied to packets coming from multiple inbound interfaces before exiting the same outbound interface.

7.1.2 Wildcard Masks in ACLs

7.1.2.1 Introducing ACL Wildcard Masking

Wildcard Masking

IPv4 ACEs include the use of wildcard masks. A wildcard mask is a string of 32 binary digits used by the router to determine which bits of the address to examine for a match.

As with subnet masks, the numbers 1 and 0 in the wildcard mask identify how to treat the corresponding IPv4 address bits. However, in a wildcard mask, these bits are used for different purposes and follow different rules.

Subnet masks use binary 1s and 0s to identify the network, subnet, and host portion of an IPv4 address. Wildcard masks use binary 1s and 0s to filter individual IPv4 addresses or groups of IPv4 addresses to permit or deny access to resources.

Wildcard masks and subnet masks differ in the way they match binary 1s and 0s. Wildcard masks use the following rules to match binary 1s and 0s:

- Wildcard mask bit 0 - Match the corresponding bit value in the address.
- Wildcard mask bit 1 - Ignore the corresponding bit value in the address.

Wildcard masks are often referred to as an inverse mask. The reason is that, unlike a subnet mask in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask the reverse is true.

Using a Wildcard Mask

Remember that a binary 0 indicates a value that is matched.

Note: Unlike IPv4 ACLs, IPv6 ACLs do not use wildcard masks. Instead, the prefix-length is used to indicate how much of an IPv6 source or destination address should be matched.

7.1.2.2 Calculating the Wildcard Mask

Calculating wildcard masks can be challenging. One shortcut method is to subtract the subnet mask from 255.255.255.255.

Wildcard Mask Calculation: Example 1

In the first example, assume you wanted to permit access to all users in the 192.168.3.0 network. Because the subnet mask is 255.255.255.0, you could take the 255.255.255.255 and subtract the subnet mask 255.255.255.0. The solution produces the wildcard mask 0.0.0.255.

Wildcard Mask Calculation: Example 2

In the second example, assume you wanted to permit network access for the 14 users in the subnet 192.168.3.32/28. The subnet mask for the IPv4 subnet is 255.255.255.240, therefore take 255.255.255.255 and subtract the subnet mask 255.255.255.240. The solution this time produces the wildcard mask 0.0.0.15.

Wildcard Mask Calculation: Example 3

In the third example, assume you wanted to match only networks 192.168.10.0 and 192.168.11.0. Again, you take the 255.255.255.255 and subtract the regular subnet mask which in this case would be 255.255.254.0. The result is 0.0.1.255.

You could accomplish the same result with statements like the two shown below:

```
R1(config)# access-list 10 permit 192.168.10.0
R1(config)# access-list 10 permit 192.168.11.0
```

It is far more efficient to configure the wildcard mask in the following way:

```
R1(config)# access-list 10 permit 192.168.10.0 0.0.1.255
```

Consider an example in which you need to match networks in the range between 192.168.16.0/24 to 192.168.31.0/24. These networks would summarize to 192.168.16.0/20. In this case, 0.0.15.255 is the correct wildcard mask to configure one efficient ACL statement, as shown below:

```
R1(config)# access-list 10 permit 192.168.16.0 0.0.15.255
```

7.1.2.3 Wildcard Mask Keywords

Working with decimal representations of binary wildcard mask bits can be tedious. To simplify this task, the keywords **host** and **any** help identify the most common uses of wildcard masking. These keywords eliminate entering wildcard masks when identifying a specific host or an entire network. These keywords also make it easier to read an ACL by providing visual clues as to the source or destination of the criteria.

The **host** keyword substitutes for the 0.0.0.0 mask. This mask states that all IPv4 address bits must match to filter just one host address.

The **any** option substitutes for the IPv4 address and 255.255.255.255 mask. This mask says to ignore the entire IPv4 address or to accept any addresses.

7.1.3 Guidelines for ACL Creation

7.1.3.1 General Guidelines for Creating ACLs

Writing ACLs can be a complex task. For every interface there may be multiple policies needed to manage the type of traffic allowed to enter or exit that interface. If the router has two interfaces configured for IPv4 and IPv6. If we needed ACLs for both protocols, on both interfaces and in both directions, this would require eight separate ACLs. Each interface would have four ACLs; two ACLs for IPv4 and two ACLs for IPv6. For each protocol, one ACL is for inbound traffic and one for outbound traffic.

Note: ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the requirements being implemented.

Here are some guidelines for using ACLs:

- Use ACLs in firewall routers positioned between your internal network and an external network such as the Internet.
- Use ACLs on a router positioned between two parts of your network to control traffic entering or exiting a specific part of your internal network.
- Configure ACLs on border routers, that is, routers situated at the edges of your networks. This provides a very basic buffer from the outside network, or between a less controlled area of your own network and a more sensitive area of your network.
- Configure ACLs for each network protocol configured on the border router interfaces.

Rules for Applying ACLs

You can configure one ACL per protocol, per direction, per interface:

- **One ACL per protocol** - To control traffic flow on an interface, an ACL must be defined for each protocol enabled on the interface.
- **One ACL per direction** - ACLs control traffic in one direction at a time on an interface. Two separate ACLs must be created to control inbound and outbound traffic.
- **One ACL per interface** - ACLs control traffic for an interface, for example, GigabitEthernet 0/0.

7.1.3.2 ACL Best Practices

Using ACLs requires attention to detail and great care. Mistakes can be costly in terms of downtime, troubleshooting efforts, and poor network service. Before configuring an ACL, basic planning is required. The figure presents guidelines that form the basis of an ACL best practices list.

Guideline	Benefit
Base your ACLs on the security policy of the organization.	This will ensure you implement organizational security guidelines.
Prepare a description of what you want your ACLs to do.	This will help you avoid inadvertently creating potential access problems.
Use a text editor to create, edit, and save ACLs.	This will help you create a library of reusable ACLs.
Test your ACLs on a development network before implementing them on a production network.	This will help you avoid costly errors.

7.1.4.1 Where to Place ACLs

The proper placement of an ACL can make the network operate more efficiently. An ACL can be placed to reduce unnecessary traffic. For example, traffic that will be denied at a remote destination should not be forwarded using network resources along the route to that destination.

Every ACL should be placed where it has the greatest impact on efficiency. The basic rules are:

- **Extended ACLs** - Locate extended ACLs as close as possible to the source of the traffic to be filtered. This way, undesirable traffic is denied close to the source network without crossing the network infrastructure.

- **Standard ACLs** - Because standard ACLs do not specify destination addresses, place them as close to the destination as possible. Placing a standard ACL at the source of the traffic will effectively prevent that traffic from reaching any other networks through the interface where the ACL is applied.

Placement of the ACL and therefore, the type of ACL used may also depend on:

- **The extent of the network administrator's control** - Placement of the ACL can depend on whether or not the network administrator has control of both the source and destination networks.
- **Bandwidth of the networks involved** - Filtering unwanted traffic at the source prevents transmission of the traffic before it consumes bandwidth on the path to a destination. This is especially important in low bandwidth networks.
- **Ease of configuration** - If a network administrator wants to deny traffic coming from several networks, one option is to use a single standard ACL on the router closest to the destination. The disadvantage is that traffic from these networks will use bandwidth unnecessarily. An extended ACL could be used on each router where the traffic originated. This will save bandwidth by filtering the traffic at the source but requires creating extended ACLs on multiple routers.

7.1.4.2 Standard ACL Placement

The administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.

Following the basic placement guidelines of placing the standard ACL close to the destination.

7.2 Standard IPv4 ACLs

7.2.1 Configure Standard IPv4 ACLs

7.2.1.1 Numbered Standard IPv4 ACL Syntax

To use numbered standard ACLs on a Cisco router, you must first create the standard ACL and then activate the ACL on an interface.

The **access-list** global configuration command defines a standard ACL with a number in the range of 1 through 99. Cisco IOS Software Release 12.0.1 extended these numbers by allowing 1300 to 1999 to be used for standard ACLs. This allows for a maximum of 798 possible standard ACLs. These additional numbers are referred to as expanded IPv4 ACLs.

The full syntax of the standard ACL command is as follows:

```
Router(config)# access-list access-list-number { deny | permit | remark } source [ source-wildcard ][ log ]
```

ACEs can permit or deny an individual host or a range of host addresses. To create a host statement in numbered ACL 10 that permits a specific host with the IPv4 address 192.168.10.10, you would enter:

```
R1(config)# access-list 10 permit host 192.168.10.10
```

To create a statement that will permit a range of IPv4 addresses in a numbered ACL 10 that permits all IPv4 addresses in the network 192.168.10.0/24, you would enter:

```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
```

To remove the ACL, the global configuration **no access-list** command is used. Issuing the **show access-list** command confirms that access list 10 has been removed.

Typically, when an administrator creates an ACL, the purpose of each statement is known and understood. However, to ensure that the administrator and others recall the purpose of a statement, remarks should be included. The **remark** keyword is used for documentation and makes access lists a great deal easier to understand. Each remark is limited to 100 characters.

7.2.1.2 Applying Standard IPv4 ACLs to Interfaces

After a standard IPv4 ACL is configured, it is linked to an interface using the **ip access-group** command in interface configuration mode:

```
Router(config-if)# ip access-group { access-list-number | access-list-name } { in | out }
```

To remove an ACL from an interface, first enter the **no ip access-group** command on the interface, and then enter the global **no access-list** command to remove the entire ACL.

7.2.1.4 Named Standard IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. When you identify your ACL with a name instead of with a number, the configuration mode and command syntax are slightly different.

The steps required to create a standard named ACL.

Step 1. Starting from the global configuration mode, use the **ip access-list** command to create a named ACL. ACL names are alphanumeric, case sensitive, and must be unique. The **ip access-list standard** *name* command is used to create a standard named ACL. After entering the command, the router is in standard (std) named ACL (nacl) configuration mode.

Note: Numbered ACLs use the global configuration command **access-list**, whereas named IPv4 ACLs use the **ip access-list** command.

Step 2. From the named ACL configuration mode, use **permit** or **deny** statements to specify one or more conditions for determining whether a packet is forwarded or dropped. You can use **remark** to add a comment to the ACL.

Step 3. Apply the ACL to an interface using the **ip access-group** *name* command. Specify whether the ACL should be applied to packets as they enter the interface (**in**) or applied to packets as they exit the interface (**out**).

Capitalizing ACL names is not required, but makes them stand out when viewing the running-config output. It also makes it less likely that you will accidentally create two different ACLs with the same name but with different uses of capitalization.

7.2.2 Modify IPv4 ACLs

7.2.2.1 Method 1 - Use a Text Editor

After someone is familiar with creating and editing ACLs, it may be easier to construct the ACL using a text editor such as Microsoft Notepad. This allows you to create or edit the ACL and then paste it into the router interface. For an existing ACL, you can use the **show running-config** command to display the ACL, copy and paste it into the text editor, make the necessary changes, and paste it back in to the router interface.

Configuration: For example, assume that the host IPv4 address was incorrectly entered. Instead of the 192.168.10.99 host, it should have been the 192.168.10.10 host. Here are the steps to edit and correct ACL 1:

Step 1. Display the ACL using the **show running-config** command.

Step 2. Highlight the ACL, copy it, and then paste it into Microsoft Notepad. Edit the list as required. After the ACL is correctly displayed in Microsoft Notepad, highlight it and copy it.

Step 3. In global configuration mode, remove the access list using the **no access-list 1** command. Otherwise, the new statements would be appended to the existing ACL. Then paste the new ACL into the configuration of the router.

Step 4. Using the **show running-config** command, verify the changes

It should be mentioned that when using the **no access-list** command, different IOS software releases act differently. If the ACL that has been deleted is still applied to an interface, some IOS versions act as if no ACL is protecting your network while others deny all traffic. For this reason it is good practice to remove the reference to the access list from the interface before modifying the access list. If there is an error in the new list, disable it and troubleshoot the problem. In that instance, the network has no ACL during the correction process.

7.2.2.2 Method 2 - Use Sequence Numbers

The initial configuration of ACL 1 included a host statement for host 192.168.10.99. This was in error. The host should have been configured as 192.168.10.10. To edit the ACL using sequence numbers follow these steps:

Step 1. Display the current ACL using the **show access-lists 1** command. The output from this command will be discussed in more detail later in this section. The sequence number is displayed at the beginning of each statement. The sequence number was automatically assigned when the access list statement was entered. Notice that the misconfigured statement has the sequence number 10.

Step 2. Enter the **ip access-lists standard** command that is used to configure named ACLs. The ACL number 1, is used as the name. First, the misconfigured statement needs to be deleted using the **no 10** command with 10 referring to the sequence number. Next, a new sequence number 10 statement is added using the command, **10 deny host 192.168.10.10**.

Note: Statements cannot be overwritten using the same sequence number as an existing statement. The current statement must be deleted first, and then the new one can be added.

Step 3. Verify the changes using the **show access-lists** command.

7.2.2.3 Editing Standard Named ACLs

Sequence numbers were used to edit a standard numbered IPv4 ACL. By referring to the statement sequence numbers, individual statements can easily be inserted or deleted. This method can also be used to edit standard named ACLs.

Note: In named access list configuration mode, use the **no** *sequence-number* command to quickly delete individual statements.

7.2.2.4 Verifying ACLs

The **show ip interface** command is used to verify the ACL on the interface. The output from this command includes the number or name of the access list and the direction in which the ACL was applied. To view an individual access list use the **show access-lists** command followed by the access list number or name. The NO_ACCESS statements may look strange.

7.2.2.5 ACL Statistics

After an ACL has been applied to an interface and some testing has occurred, the **show access-lists** command will show statistics for each statement that has been matched. Both permit and deny statements will track statistics for matches; however, recall that every ACL has an implied deny any as the last statement. This statement will not appear in the **show access-lists** command; therefore, statistics for that statement will not appear. To view statistics for the implied deny any statement, the statement can be configured manually and will appear in the output.

During testing of an ACL, the counters can be cleared using the **clear access-list counters** command. This command can be used alone or with the number or name of a specific ACL.

7.2.3 Securing VTY ports with a Standard IPv4 ACL

7.2.3.1 The access-class Command

You can improve the security of administrative lines by restricting VTY access. Restricting VTY access is a technique that allows you to define which IP addresses are allowed remote access to the router EXEC process. You can specify which IP addresses are allowed remote access to your router with an ACL and an **access-class** statement configured on your VTY lines. Use this technique with SSH to further improve administrative access security.

The **access-class** command configured in line configuration mode restricts incoming and outgoing connections between a particular VTY (into a Cisco device) and the addresses in an access list.

The command syntax of the **access-class** command is:

```
Router(config-line)# access-class access-list-number { in [ vrf-also ] | out }
```

The parameter **in** restricts incoming connections between the addresses in the access list and the Cisco device, while the parameter **out** restricts outgoing connections between a particular Cisco device and the addresses in the access list.

The following should be considered when configuring access lists on VTYs:

- Both named and numbered access lists can be applied to VTYs.
- Identical restrictions should be set on all the VTYs, because a user can attempt to connect to any of them.

Note: Access lists apply to packets that travel through a router. They are not designed to block packets that originate within the router. By default, an outbound ACL does not prevent remote access connections initiated from the router.

7.3 Extended IPv4 ACLs

7.3.1 Structure of an Extended IPv4 ACLs

7.3.1.1 Extended ACLs

Testing Packets with Extended ACLs

For more precise traffic-filtering control, extended IPv4 ACLs can be created. Extended ACLs are numbered 100 to 199 and 2000 to 2699, providing a total of 799 possible extended numbered ACLs. Extended ACLs can also be named.

Extended ACLs are used more often than standard ACLs because they provide a greater degree of control. Standard ACLs, extended ACLs have the ability to check source addresses of packets, but they also have the ability to check the destination address, protocols, and port numbers (or services). This provides a greater range of criteria on which to base the ACL. For example, one extended ACL can allow email traffic from a network to a specific destination while denying file transfers and web browsing.

7.3.1.2 Filtering Ports and Services

The ability to filter on protocol and port number allows network administrators to build very specific extended ACLs. An application can be specified by configuring either the port number or the name of a well-known port.

Using Port Numbers

```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 23
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 20
```

Using Keywords

```
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq telnet
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp-data
```

Figures above shows some examples of how an administrator specifies a TCP or UDP port number by placing it at the end of the extended ACL statement. Logical operations can be used, such as equal (eq), not equal (neq), greater than (gt), and less than (lt).

7.3.2 Configure Extended IPv4 ACLs

7.3.2.1 Configuring Extended ACLs

The procedural steps for configuring extended ACLs are the same as for standard ACLs. The extended ACL is first configured, and then it is activated on an interface. However, the command syntax and parameters are more complex to support the additional features provided by extended ACLs.

Note: The internal logic applied to the ordering of standard ACL statements does not apply to extended ACLs. The order in which the statements are entered during configuration is the order they are displayed and processed.

Parameter	Description
access-list-number	Identifies the access list using a number in the range 100 to 199 (for an extended IP ACL) and 2000 to 2699 (expanded IP ACLs).
deny	Denies access if the conditions are matched.
permit	Permits access if the conditions are matched.
remark	Adds a remark about entries in an IP access list to make the list easier to understand and scan.
protocol	Name or number of an Internet protocol. Common keywords include **icmp**, **ip**, **tcp**, or **udp**. To match any Internet protocol (including ICMP, TCP, and UDP) use the **ip** keyword.
source	Number of the network or host from which the packet is being sent.
source-wildcard	Wildcard bits to be applied to source.
destination	Number of the network or host to which the packet is being sent.
destination-wildcard	Wildcard bits to be applied to the destination.
operator	(Optional) Compares source or destination ports. Possible operands include **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range** (inclusive range).
port	(Optional) The decimal number or name of a TCP or UDP port.
established	(Optional) For the TCP protocol only; indicates an established connection.

Figure above shows the common command syntax for extended IPv4 ACLs. Note that there are many keywords and parameters for extended ACLs. It is not necessary to use all of the keywords and parameters when configuring an extended ACL. Recall that the **?** can be used to get help when entering complex commands.

Figure below shows an example of an extended ACL. In this example, the network administrator has configured ACLs to restrict network access to allow website browsing only from the LAN attached to interface G0/0 to any external network. ACL 103 allows traffic coming from any address on the 192.168.10.0 network to go to any destination, subject to the limitation that the traffic is using ports 80 (HTTP) and 443 (HTTPS) only.

```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
```

The nature of HTTP requires that traffic flow back into the network from websites accessed from internal clients. The network administrator wants to restrict that return traffic to HTTP exchanges from requested websites, while denying all other traffic. ACL 104 does that by blocking all incoming traffic, except for previously established connections. The permit statement in ACL 104 allows inbound traffic using the **established** parameter.

The **established** parameter allows only responses to traffic that originates from the 192.168.10.0/24 network to return to that network. A match occurs if the returning TCP segment has the ACK or reset (RST) bits set, which indicates that the packet belongs to an existing connection. Without the **established** parameter in the ACL statement, clients could send traffic to a web server, but not receive traffic returning from the web server.

7.3.2.2 Applying Extended ACLs to Interfaces

In the previous example, the network administrator configured an ACL to allow users from the 192.168.10.0/24 network to browse both insecure and secure websites. Even though it has been configured, the ACL will not filter traffic until it is applied to an interface. To apply an ACL to an

interface, first consider whether the traffic to be filtered is going in or out. When a user on the internal LAN accesses a website on the Internet, traffic is traffic going out to the Internet. When an internal user receives an email from the Internet, traffic is coming into the local router. However, when applying an ACL to an interface, in and out take on different meanings. From an ACL consideration, in and out are in reference to the router interface.

In the topology, R1 has three interfaces. It has a serial interface, S0/0/0, and two Gigabit Ethernet interfaces, G0/0 and G0/1. Recall that an extended ACL should typically be applied close to the source. In this topology the interface closest to the source of the target traffic is the G0/0 interface.

```
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)# access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0
R1(config-if)# ip access-group 103 in
R1(config-if)# ip access-group 104 out
```

Web request traffic from users on the 192.168.10.0/24 LAN is inbound to the G0/0 interface. Return traffic from established connections to users on the LAN is outbound from the G0/0 interface. The example applies the ACL to the G0/0 interface in both directions. The inbound ACL, 103, checks for the type of traffic. The outbound ACL, 104, checks for return traffic from established connections. This will restrict 192.168.10.0 Internet access to allow only website browsing.

Note: The access lists could have been applied to the S0/0/0 interface but in that case, the router's ACL process would have to examine all packets entering the router, not only traffic to and from 192.168.11.0. This would cause unnecessary processing by the router.

7.3.2.3 Filtering Traffic with Extended ACLs

The first example denies FTP traffic from subnet 192.168.11.0 that is going to subnet 192.168.10.0, but permits all other traffic. Remember that FTP uses TCP ports 20 and 21; therefore, the ACL requires both port name keywords **ftp** and **ftp-data** or **eq 20** and **eq 21** to deny FTP.

If using port numbers instead of port names, the commands would be written as:

access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 20

access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 21

To prevent the implied **deny any** statement at the end of the ACL from blocking all traffic, the **permit ip any any** statement is added. Without at least one **permit** statement in an ACL, all traffic on the interface where that ACL was applied would be dropped. The ACL should be applied inbound on the G0/1 interface so that traffic from the 192.168.11.0/24 LAN is filtered as it enters the router interface.

The second example , denies Telnet traffic from any source to the 192.168.11.0/24 LAN, but allows all other IP traffic. Because traffic destined for the 192.168.11.0/24 LAN is outbound on interface G0/1, the ACL would be applied to G0/1 using the **out** keyword. Note the use of the **any** keywords in the permit statement. This permit statement is added to ensure that no other traffic is blocked.

Note: The two examples both use the **permit ip any any** statement at the end of the ACL. For greater security the **permit 192.168.11.0 0.0.0.255 any** command may be used.

7.3.2.4 Creating Named Extended ACLs

Named extended ACLs are created in essentially the same way that named standard ACLs are created. Follow these steps to create an extended ACL, using names:

Step 1. From global configuration mode, use the **ip access-list extended** *name* command to define a name for the extended ACL.

Step 2. In named ACL configuration mode, specify the conditions to **permit** or **deny**.

Step 3. From interface configuration mode, apply the named ACL using the **ip access-group** [**in** | **ou**t] *name* command.

Step 4. Return to privileged EXEC mode and verify the ACL with the **show access-lists** *name* command.

Step 5. Save the entries in the configuration file with the **copy running-config startup-config** command.

To remove a named extended ACL, use the **no ip access-list extended** *name* global configuration command.

7.3.2.5 Verifying Extended ACLs

After an ACL has been configured and applied to an interface, use Cisco IOS **show** commands to verify the configuration.

Unlike standard ACLs, extended ACLs do not implement the same internal logic and hashing function. The output and sequence numbers displayed in the **show access-lists** command output is the order in which the statements were entered. Host entries are not automatically listed prior to range entries.

The **show ip interface** command is used to verify the ACL on the interface and the direction in which it was applied. The output from this command includes the number or name of the access list and the direction in which the ACL was applied. The capitalized ACL names BROWSING and SURFING stand out in the screen output.

After an ACL configuration has been verified, the next step is to confirm that the ACLs work as planned; blocking and permitting traffic as expected.

The guidelines discussed earlier in this section, suggest that ACLs should be configured on a test network and then implemented on the production network.

7.3.2.6 Editing Extended ACLs

An extended ACL can be edited in one of two ways:

- **Method 1 Text editor** - Using this method, the ACL is copied and pasted into the text editor where the changes are made. The current access list is removed using the **no access-list** command. The modified ACL is then pasted back into the configuration.
- **Method 2 Sequence numbers** - Sequence numbers can be used to delete or insert an ACL statement. The **ip access-list extended** *name* command is used to enter named-ACL configuration mode. If the ACL is numbered instead of named, the ACL number is used in the *name* parameter. ACEs can be inserted or removed.

7.4 Troubleshoot ACLs

7.4.1 Processing Packets with ACLs

7.4.1.1 The Implicit Deny Any

A single-entry ACL with only one deny entry has the effect of denying all traffic. At least one permit ACE must be configured in an ACL or all traffic is blocked.

For the network in the figure, applying either ACL 1 or ACL 2 to the S0/0/0 interface of R1 in the outbound direction will have the same effect. Network 192.168.10.0 will be permitted to access the networks reachable through S0/0/0, while 192.168.11.0 will not be allowed to access those networks. In ACL 1, if a packet does not match the permit statement, it is discarded.

7.4.1.2 The Order of ACEs in an ACL

Cisco IOS applies an internal logic when accepting and processing standard ACEs. As discussed previously, ACEs are processed sequentially; therefore, the order in which ACEs are entered is important.

7.4.1.3 Cisco IOS Reorders Standard ACLs

The order in which standard ACEs are entered may not be the order that they are stored, displayed, or processed by the router.

The **show running-config** command is used to verify the ACL configuration. Notice that the statements are listed in a different order than they were entered. We will use the **show access-lists** command to understand the logic behind this.

The order in which the standard ACEs are listed is the sequence used by the IOS to process the list. Notice that the statements are grouped into two sections, host statements followed by range statements. The sequence number indicates the order that the statement was entered, not the order the statement will be processed.

The host statements are listed first but not necessarily in the order that they were entered. The IOS puts host statements in an order using a special hashing function. The resulting order optimizes the search for a host ACL entry. The range statements are displayed after the host statements. These statements are listed in the order in which they were entered.

Note: The hashing function is only applied to host statements in an IPv4 standard access list. The details of the hashing function are beyond the scope of this course.

Recall that standard and numbered ACLs can be edited using sequence numbers. When inserting a new ACL statement, the sequence number will only affect the location of a range statement in the list. Host statements will always be put in order using the hashing function.

7.4.1.4 Routing Processes and ACLs

If the frame address is accepted, the frame information is stripped off and the router checks for an ACL on the inbound interface. If an ACL exists, the packet is tested against the statements in the list.

If the packet matches a statement, the packet is either permitted or denied. If the packet is accepted, it is then checked against routing table entries to determine the destination interface. If a routing table entry exists for the destination, the packet is then switched to the outgoing interface, otherwise the packet is dropped.

Next, the router checks whether the outgoing interface has an ACL. If an ACL exists, the packet is tested against the statements in the list. If the packet matches a statement, it is either permitted or denied.

If there is no ACL or the packet is permitted, the packet is encapsulated in the new Layer 2 protocol and forwarded out the interface to the next device.

CHAPTER 8: DHCP

Every device that connects to a network needs a unique IP address. Network administrators assign static IP addresses to routers, servers, printers, and other network devices whose locations (physical and logical) are not likely to change. These are usually devices that provide services to users and devices on the network; therefore, the addresses assigned to them should remain constant. Additionally, static addresses enable administrators to manage these devices remotely. It is easier for network administrators to access a device when they can easily determine its IP address.

However, computers and users in an organization often change locations, physically and logically. It can be difficult and time consuming for administrators to assign new IP addresses every time an employee moves. Additionally, for mobile employees working from remote locations, manually setting the correct network parameters can be challenging. Even for desktop clients, the manual assignment of IP addresses and other addressing information presents an administrative burden, especially as the network grows.

Introducing a Dynamic Host Configuration Protocol (DHCP) server to the local network simplifies IP address assignment to both desktop and mobile devices. Using a centralized DHCP server enables organizations to administer all dynamic IP address assignments from a single server. This practice makes IP address management more effective and ensures consistency across the organization, including branch offices.

DHCP is available for both IPv4 (DHCPv4) and for IPv6 (DHCPv6).

8.1 DHCPv4
8.1.1 DHCPv4 Operation

8.1.1.1 Introducing DHCPv4
DHCPv4 assigns IPv4 addresses and other network configuration information dynamically. Because desktop clients typically make up the bulk of network nodes, DHCPv4 is an extremely useful and timesaving tool for network administrators.

A dedicated DHCPv4 server is scalable and relatively easy to manage. However, in a small branch or SOHO location, a Cisco router can be configured to provide DHCPv4 services without the need for a dedicated server. Cisco IOS software supports an optional, full-featured DHCPv4 server.

The DHCPv4 server dynamically assigns, or leases, an IPv4 address from a pool of addresses for a limited period of time chosen by the server, or until the client no longer needs the address.

Clients lease the information from the server for an administratively defined period. Administrators configure DHCPv4 servers to set the leases to time out at different intervals. The lease is typically anywhere from 24 hours to a week or more. When the lease expires, the client must ask for another address, although the client is typically reassigned the same address.

8.1.1.2 DHCPv4 Operation
DHCPv4 works in a client/server mode. When a client communicates with a DHCPv4 server, the server assigns or leases an IPv4 address to that client. The client connects to the network with that leased IP address until the lease expires. The client must contact the DHCP server periodically to extend the lease. This lease mechanism ensures that clients that move or power off do not keep addresses that they no longer need. When a lease expires, the DHCP server returns the address to the pool where it can be reallocated as necessary.

Lease Origination

When the client boots (or otherwise wants to join a network), it begins a four step process to obtain a lease. The process with a broadcast DHCPDISCOVER message with its own MAC address to discover available DHCPv4 servers.

DHCP Discover (DHCPDISCOVER)

The DHCPDISCOVER message finds DHCPv4 servers on the network. Because the client has no valid IPv4 information at bootup, it uses Layer 2 and Layer 3 broadcast addresses to communicate with the server.

DHCP Offer (DHCPOFFER)

When the DHCPv4 server receives a DHCPDISCOVER message, it reserves an available IPv4 address to lease to the client. The server also creates an ARP entry consisting of the MAC address of the requesting client and the leased IPv4 address of the client. The DHCPv4 server sends the binding DHCPOFFER message to the requesting client. The DHCPOFFER message is sent as a unicast, using the Layer 2 MAC address of the server as the source address and the Layer 2 MAC address of the client as the destination.

DHCP Request (DHCPREQUEST)

When the client receives the DHCPOFFER from the server, it sends back a DHCPREQUEST message. This message is used for both lease origination and lease renewal. When used for lease origination, the DHCPREQUEST serves as a binding acceptance notice to the selected server for the parameters it has offered and an implicit decline to any other servers that may have provided the client a binding offer.

Many enterprise networks use multiple DHCPv4 servers. The DHCPREQUEST message is sent in the form of a broadcast to inform this DHCPv4 server and any other DHCPv4 servers about the accepted offer.

DHCP Acknowledgment (DHCPACK)

On receiving the DHCPREQUEST message, the server verifies the lease information with an ICMP ping to that address to ensure it is not being used already, creates a new ARP entry for the client lease, and replies with a unicast DHCPACK message. The DHCPACK message is a duplicate of the DHCPOFFER, except for a change in the message type field. When the client receives the DHCPACK message, it logs the configuration information and performs an ARP lookup for the assigned address. If there is no reply to the ARP, the client knows that the IPv4 address is valid and starts using it as its own.

Lease Renewal

DHCP Request (DHCPREQUEST)

Before the lease expires, the client sends a DHCPREQUEST message directly to the DHCPv4 server that originally offered the IPv4 address. If a DHCPACK is not received within a specified amount of time, the client broadcasts another DHCPREQUEST so that one of the other DHCPv4 servers can extend the lease.

DHCP Acknowledgment (DHCPACK)

On receiving the DHCPREQUEST message, the server verifies the lease information by returning a DHCPACK.

7.1.1.3 DHCPv4 Message Format

The DHCPv4 message format is used for all DHCPv4 transactions. DHCPv4 messages are encapsulated within the UDP transport protocol. DHCPv4 messages sent from the client use UDP source port 68 and destination port 67. DHCPv4 messages sent from the server to the client use UDP source port 67 and destination port 68.

The following fields shows the format of a DHCPv4 message:

- **Operation (OP) Code** - Specifies the general type of message. A value of 1 indicates a request message; a value of 2 is a reply message.
- **Hardware Type** - Identifies the type of hardware used in the network. For example, 1 is Ethernet, 15 is Frame Relay, and 20 is a serial line. These are the same codes used in ARP messages.
- **Hardware Address Length** - Specifies the length of the address.
- **Hops** - Controls the forwarding of messages. Set to 0 by a client before transmitting a request.
- **Transaction Identifier** - Used by the client to match the request with replies received from DHCPv4 servers.
- **Seconds** - Identifies the number of seconds elapsed since a client began attempting to acquire or renew a lease. Used by DHCPv4 servers to prioritize replies when multiple client requests are outstanding.
- **Flags** - Used by a client that does not know its IPv4 address when it sends a request. Only one of the 16 bits is used, which is the broadcast flag. A value of 1 in this field tells the DHCPv4 server or relay agent receiving the request that the reply should be sent as a broadcast.
- **Client IP Address** - Used by a client during lease renewal when the address of the client is valid and usable, not during the process of acquiring an address. The client puts its own IPv4 address in this field if and only if it has a valid IPv4 address while in the bound state; otherwise, it sets the field to 0.
- **Your IP Address** - Used by the server to assign an IPv4 address to the client.
- **Server IP Address** - Used by the server to identify the address of the server that the client should use for the next step in the bootstrap process, which may or may not be the server sending this reply. The sending server always includes its own IPv4 address in a special field called the Server Identifier DHCPv4 option.
- **Gateway IP Address** - Routes DHCPv4 messages when DHCPv4 relay agents are involved. The gateway address facilitates communications of DHCPv4 requests and replies between the client and a server that are on different subnets or networks.
- **Client Hardware Address** - Specifies the physical layer of the client.
- **Server Name** - Used by the server sending a DHCPOFFER or DHCPACK message. The server may optionally put its name in this field. This can be a simple text nickname or a DNS domain name, such as dhcpserver.netacad.net.
- **Boot Filename** - Optionally used by a client to request a particular type of boot file in a DHCPDISCOVER message. Used by a server in a DHCPOFFER to fully specify a boot file directory and filename.
- **DHCP Options** - Holds DHCP options, including several parameters required for basic DHCP operation. This field is variable in length. Both client and server may use this field.

7.1.1.4 DHCPv4 Discover and Offer Messages

If a client is configured to receive its IPv4 settings dynamically and wants to join the network, it requests addressing values from the DHCPv4 server. The client transmits a DHCPDISCOVER message on its local network when it boots or senses an active network connection. Because the client has no way of knowing the subnet to which it belongs, the DHCPDISCOVER message is an IPv4 broadcast (destination IPv4 address of 255.255.255.255). The client does not have a configured IPv4 address yet, so the source IPv4 address of 0.0.0.0 is used.

The client IPv4 address (CIADDR), default gateway address (GIADDR), and subnet mask are all marked to indicate that the address 0.0.0.0 is used.

Note: Unknown information is sent as 0.0.0.0.

When the DHCPv4 server receives the DHCPDISCOVER message, it responds with a DHCPOFFER message. This message contains initial configuration information for the client, including the IPv4 address that the server offers, the subnet mask, the lease duration, and the IPv4 address of the DHCPv4 server making the offer.

The DHCPOFFER message can be configured to include other information, such as the lease renewal time and DNS address.

The DHCP server responds to the DHCPDISCOVER by assigning values to the CIADDR and subnet mask. The frame is constructed using the client hardware address (CHADDR) and sent to the requesting client.

The client and server send acknowledgment messages, and the process is complete.

8.1.2 Configuring a Basic DHCPv4 Server

8.1.2.1 Configuring a Basic DHCPv4 Server

A Cisco router running Cisco IOS software can be configured to act as a DHCPv4 server. The Cisco IOS DHCPv4 server assigns and manages IPv4 addresses from specified address pools within the router to DHCPv4 clients.

Step 1. Excluding IPv4 Addresses

The router functioning as the DHCPv4 server assigns all IPv4 addresses in a DHCPv4 address pool unless configured to exclude specific addresses. Typically, some IPv4 addresses in a pool are assigned to network devices that require static address assignments. Therefore, these IPv4 addresses should not be assigned to other devices. To exclude specific addresses, use the **ip dhcp excluded-address** command.

A single address or a range of addresses can be excluded by specifying the low-address and high-address of the range. Excluded addresses should include the addresses assigned to routers, servers, printers, and other devices that have been or will be manually configured.

Step 2. Configuring a DHCPv4 Pool

Configuring a DHCPv4 server involves defining a pool of addresses to assign. The **ip dhcp pool** *pool-name* command creates a pool with the specified name and puts the router in DHCPv4 configuration mode, which is identified by this prompt Router(dhcp-config)#.

Step 3. Configuring Specific Tasks

The address pool and default gateway router must be configured. Use the **network** statement to define the range of available addresses.

Use the **default-router** command to define the default gateway router. Typically, the gateway is the LAN interface of the router closest to the client devices. One gateway is required, but you can list up to eight addresses if there are multiple gateways.

Other DHCPv4 pool commands are optional. For example, the IPv4 address of the DNS server that is available to a DHCPv4 client is configured using the **dns-server** command. The **domain-name** *domain* command is used to define the domain name. The duration of the DHCPv4 lease can be changed using the **lease** command. The default lease value is one day. The **netbios-name-server** command is used to define the NetBIOS WINS server.

DHCPv4 Example

```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
```

A sample configuration with basic DHCPv4 parameters configured on router R1 is shown in Figure above. R1 is configured as a DHCPv4 server for the 192.168.10.0/24 LAN.

Disabling DHCPv4

The DHCPv4 service is enabled, by default. To disable the service, use the **no service dhcp** global configuration mode command. Use the **service dhcp** global configuration mode command to re-enable the DHCPv4 server process. Enabling the service has no effect if the parameters are not configured.

8.1.2.2 Verifying DHCPv4

The **show running-config | section dhcp** command output displays the DHCPv4 commands configured on Router. The **| section** parameter displays only the commands associated with DHCPv4 configuration.

The operation of DHCPv4 can be verified using the **show ip dhcp binding** command. This command displays a list of all IPv4 address to MAC address bindings that have been provided by the DHCPv4 service. The **show ip dhcp server statistics**, is used to verify that messages are being received or sent by the router. This command displays count information regarding the number of DHCPv4 messages that have been sent and received.

8.1.2.3 DHCPv4 Relay

What is DHCP Relay?

In a complex hierarchical network, enterprise servers are usually located in a server farm. These servers may provide DHCP, DNS, TFTP, and FTP services for the network. Network clients are not typically on the same subnet as those servers. In order to locate the servers and receive services, clients often use broadcast messages.

For example, PC1 is attempting to renew its IPv4 address. To do so, the ipconfig /release command is issued. Notice that the IPv4 address is released and the address is shown to be 0.0.0.0. Next, the ipconfig /renew command is issued. This command causes PC1 to broadcast a DHCPDISCOVER message. The output shows that PC1 is unable to locate the DHCPv4 server. Because routers do not forward broadcasts, the request is not successful.

As a solution to this problem, an administrator can add DHCPv4 servers on all the subnets. However, running these services on several computers creates additional cost and administrative overhead.

A better solution is to configure a Cisco IOS helper address. This solution enables a router to forward DHCPv4 broadcasts to the DHCPv4 server. When a router forwards address assignment/parameter requests, it is acting as a DHCPv4 relay agent. In the example topology, PC1 would broadcast a request to locate a DHCPv4 server. If R1 was configured as a DHCPv4 relay agent, it would forward the request to the DHCPv4 server located on subnet 192.168.11.0.

DHCPv4 is not the only service that the router can be configured to relay. By default, the ip helper-address command forwards the following eight UDP services:

- Port 37: Time
- Port 49: TACACS
- Port 53: DNS
- Port 67: DHCP/BOOTP client
- Port 68: DHCP/BOOTP server

- Port 69: TFTP
- Port 137: NetBIOS name service
- Port 138: NetBIOS datagram service

8.1.3 Configure DHCPv4 Client

8.1.3.1 Configuring a Router as DHCPv4 Client

Sometimes, Cisco routers in small office/home office (SOHO) and branch sites have to be configured as DHCPv4 clients in a similar manner to client computers. The method used depends on the ISP. However, in its simplest configuration, the Ethernet interface is used to connect to a cable or DSL modem. To configure an Ethernet interface as a DHCP client, use the **ip address dhcp** interface configuration mode command.

8.1.3.2 Configuring a Wireless Router as a DHCPv4 Client

Typically, wireless routers for home or small office use connect to an ISP using a DSL or cable modem. In most cases, wireless routers are set to receive IPv4 addressing information automatically from the ISP.

Notice that the Internet connection type is set to **Automatic Configuration - DHCP**. This selection is used when the router is connected to a DSL or cable modem and acts as a DHCPv4 client, requesting an IPv4 address from the ISP.

8.1.4 Troubleshoot DHCPv4

8.1.4.1 Troubleshooting Tasks

DHCPv4 problems can arise for a multitude of reasons, such as software defects in operating systems, NIC drivers, or DHCP relay agents, but the most common are configuration issues. Because of the number of potentially problematic areas, a systematic approach to troubleshooting is required.

Troubleshooting Task 1: Resolve IPv4 Address Conflicts

An IPv4 address lease can expire on a client still connected to a network. If the client does not renew the lease, the DHCPv4 server can reassign that IPv4 address to another client. When the client reboots, it requests an IPv4 address. If the DHCPv4 server does not respond quickly, the client uses the last IPv4 address. The situation then arises where two clients are using the same IPv4 address, creating a conflict.

The **show ip dhcp conflict** command displays all address conflicts recorded by the DHCPv4 server. The server uses the **ping** command to detect clients. The client uses Address Resolution Protocol (ARP) to detect conflicts. If an address conflict is detected, the address is removed from the pool and not assigned until an administrator resolves the conflict.

This output displays IP addresses that have conflicts with the DHCP server. It shows the detection method and detection time for conflicting IP addresses that the DHCP server has offered.

Troubleshooting Task 2: Verify Physical Connectivity

First, use the **show interfaces** *interface* command to confirm that the router interface acting as the default gateway for the client is operational. If the state of the interface is anything other than up, the port does not pass traffic, including DHCP client requests.

Troubleshooting Task 3: Test Connectivity using a Static IP Address

When troubleshooting any DHCPv4 issue, verify network connectivity by configuring static IPv4 address information on a client workstation. If the workstation is unable to reach network resources with a statically configured IPv4 address, the root cause of the problem is not DHCPv4. At this point, network connectivity troubleshooting is required.

Troubleshooting Task 4: Verify Switch Port Configuration

If the DHCPv4 client is unable to obtain an IPv4 address from the DHCPv4 server on startup, attempt to obtain an IPv4 address from the DHCPv4 server by manually forcing the client to send a DHCPv4 request.

Note: If there is a switch between the client and the DHCPv4 server, and the client is unable to obtain the DHCP configuration, switch port configuration issues may be the cause. These causes may include issues from trunking and channeling, STP, and RSTP. PortFast and edge port configurations resolve the most common DHCPv4 client issues that occur with an initial installation of a Cisco switch.

Troubleshooting Task 5: Test DHCPv4 Operation on the Same Subnet or VLAN

It is important to distinguish whether DHCPv4 is functioning correctly when the client is on the same subnet or VLAN as the DHCPv4 server. If DHCPv4 is working correctly when the client is on the same subnet or VLAN, the problem may be the DHCP relay agent. If the problem persists even with testing DHCPv4 on the same subnet or VLAN as the DHCPv4 server, the problem may actually be with the DHCPv4 server.

8.1.4.2 Verify Router DHCPv4 Configuration

When the DHCPv4 server is located on a separate LAN from the client, the router interface facing the client must be configured to relay DHCPv4 requests by configuring the IPv4 helper address. If the IPv4 helper address is not configured properly, client DHCPv4 requests are not forwarded to the DHCPv4 server.

Follow these steps to verify the router configuration:

Step 1. Verify that the **ip helper-address** command is configured on the correct interface. It must be present on the inbound interface of the LAN containing the DHCPv4 client workstations and must be directed to the correct DHCPv4 server.

The **show ip interface** command can also be used to verify the DHCPv4 relay on an interface.

Step 2. Verify that the global configuration command **no service dhcp** has not been configured. This command disables all DHCP server and relay functionality on the router. The command **service dhcp** does not appear in the running-config, because it is the default configuration.

The **show running-config | include no service dhcp** command verifies that the DHCPv4 service is enabled since there is no match for the **show running-config | include no service dhcp** command. If the service had been disabled, the **no service dhcp** command would be displayed in the output.

8.1.4.3 Debugging DHCPv4

On routers configured as DHCPv4 servers, the DHCPv4 process fails if the router is not receiving requests from the client. As a troubleshooting task, verify that the router is receiving the DHCPv4 request from the client. This troubleshooting step involves configuring an ACL for debugging output.

The figure shows an extended ACL permitting only packets with UDP destination ports of 67 or 68.The extended ACL is used with the **debug ip packet** command to display only DHCPv4 messages.

Nevertheless, the router did receive a broadcast packet with the source and destination IP and UDP ports that are correct for DHCPv4.

Another useful command for troubleshooting DHCPv4 operation is the **debug ip dhcp server events** command. This command reports server events, like address assignments and database updates.

8.2 DHCPv6

8.2.1 SLAAC and DHCPv6

8.2.1.1 Stateless Address Autoconfiguration (SLAAC)

Similar to IPv4, IPv6 global unicast addresses can be configured manually or dynamically. However, there are two methods in which IPv6 global unicast addresses can be assigned dynamically:

- Stateless Address Autoconfiguration (SLAAC)
- Dynamic Host Configuration Protocol for IPv6 (Stateful DHCPv6)

Introducing SLAAC

SLAAC is a method in which a device can obtain an IPv6 global unicast address without the services of a DHCPv6 server. At the core of SLAAC is ICMPv6. ICMPv6 is similar to ICMPv4 but includes additional functionality and is a much more robust protocol. SLAAC uses ICMPv6 Router Solicitation and Router Advertisement messages to provide addressing and other configuration information that would normally be provided by a DHCP server:

- **Router Solicitation (RS) message** - When a client is configured to obtain its addressing information automatically using SLAAC, the client sends an RS message to the router. The RS message is sent to the IPv6 all-routers multicast address FF02::2.
- **Router Advertisement (RA) message** - RA messages are sent by routers to provide addressing information to clients configured to obtain their IPv6 addresses automatically. The RA message includes the prefix and prefix length of the local segment. A client uses this information to create its own IPv6 global unicast address. A router sends an RA message periodically, or in response to an RS message. By default, Cisco routers send RA messages every 200 seconds. RA messages are always sent to the IPv6 all-nodes multicast address FF02::1.

As the name indicates, SLAAC is stateless. A stateless service means there is no server that maintains network address information. Unlike DHCP, there is no SLAAC server that knows which IPv6 addresses are being used and which ones are available.

8.2.1.2 SLAAC Operation
SLAAC Operation

A router must have IPv6 routing enabled before it can send RA messages:

Router(config)# **ipv6 unicast-routing**

There are two ways PC can create its own unique IID:

- **EUI-64** - Using the EUI-64 process, PC1 will create an IID using its 48-bit MAC address.
- **Randomly generated** - The 64-bit IID can be a random number generated by the client operating system.

As shown in Figure 3, PC1 can create a 128-bit IPv6 global unicast address by combining the 64-bit prefix with the 64-bit IID. PC1 will use the link-local address of the router as its IPv6 default gateway address.

This process is part of ICMPv6 Neighbor Discovery and is known as Duplicate Address Detection (DAD).

8.2.1.3 SLAAC and DHCPv6
The decision of whether a client is configured to obtain its IPv6 address information automatically using SLAAC, DHCPv6, or a combination of both depends on the settings within the RA message.

The two flags are the Managed Address Configuration flag (M flag) and the Other Configuration flag (O flag).

Using different combinations of the M and O flags, RA messages have one of three addressing options for the IPv6 device:

- SLAAC (Router Advertisement only)
- Stateless DHCPv6 (Router Advertisement and DHCPv6)
- Stateful DHCPv6 (DHCPv6 only)

Regardless of the option used, it is recommended by RFC 4861 that all IPv6 devices perform Duplicate Address Detection (DAD) on any unicast address, including addresses configured using SLAAC or DHCPv6. DAD is implemented using ICMPv6, which is specified by RFC 4443.

Note: Although the RA message specifies the process the client should use in obtaining an IPv6 address dynamically, the client operating system may choose to ignore the RA message and use the services of a DHCPv6 server exclusively.

8.2.1.4 SLAAC Option

SLAAC Option (Router Advertisement only)

SLAAC is the default option on Cisco routers. Both the M flag and the O flag are set to 0 in the RA.

This option instructs the client to use the information in the RA message exclusively. This includes prefix, prefix-length, DNS server, MTU, and default gateway information. There is no further information available from a DHCPv6 server. The IPv6 global unicast address is created by combining the prefix from RA and an Interface ID using either EUI-64 or a randomly generated value.

RA messages are configured on an individual interface of a router. To re-enable an interface for SLAAC that might have been set to another option, the M and O flags need to be reset to their initial values of 0. This is done using the following interface configuration mode commands:

```
Router(config-if)# no ipv6 nd managed-config-flag
Router(config-if)# no ipv6 nd other-config-flag
```

8.2.1.5 Stateless DHCPv6 Option

Although DHCPv6 is similar to DHCPv4 in what it provides, the two protocols are independent of each other. DHCPv6 is defined in RFC 3315. There has been a lot of work done on this specification over the years as indicated by the fact that DHCPv6 RFC has the highest revision number of any Internet draft.

Stateless DHCPv6 Option (Router Advertisement and DHCPv6)

The stateless DHCPv6 option informs the client to use the information in the RA message for addressing, but additional configuration parameters are available from a DHCPv6 server.

Using the prefix and prefix length in the RA message, along with EUI-64 or a randomly generated IID, the client creates its IPv6 global unicast address.

The client will then communicate with a stateless DHCPv6 server to obtain additional information not provided in the RA message. This may be a list of DNS server IPv6 addresses, for example. This process is known as stateless DHCPv6 because the server is not maintaining any client state information (i.e., a list of available and allocated IPv6 addresses). The stateless DHCPv6 server is only providing configuration parameters for clients, not IPv6 addresses.

For stateless DHCPv6, the O flag is set to 1 and the M flag is left at the default setting of 0. The O flag value of 1 is used to inform the client that additional configuration information is available from a stateless DHCPv6 server.

To modify the RA message sent on the interface of a router to indicate stateless DHCPv6, use the following command:

Router(config-if)# **ipv6 nd other-config-flag**

8.2.1.6 Stateful DHCPv6 Option

This option is the most similar to DHCPv4. In this case, the RA message informs the client not to use the information in the RA message. All addressing information and configuration information must be

obtained from a stateful DHCPv6 server. This is known as stateful DHCPv6 because the DHCPv6 server maintains IPv6 state information. This is similar to a DHCPv4 server allocating addresses for IPv4.

The M flag indicates whether or not to use stateful DHCPv6. The O flag is not involved. The following command is used to change the M flag from 0 to 1 to signify stateful DHCPv6:

Router(config-if)# **ipv6 nd managed-config-flag**

8.2.1.7 DHCPv6 Operations

Stateless or stateful DHCPv6, or both begin with an ICMPv6 RA message from the router. The RA message might have been a periodic message or solicited by the device using an RS message.

If stateless or stateful DHCPv6 is indicated in the RA message, then the device begins DHCPv6 client/server communications.

DHCPv6 Communications

When stateless DHCPv6 or stateful DHCPv6 is indicated by the RA, DHCPv6 operation is invoked. DHCPv6 messages are sent over UDP. DHCPv6 messages from the server to the client use UDP destination port 546. The client sends DHCPv6 messages to the server using UDP destination port 547.

The client, now a DHCPv6 client, needs to locate a DHCPv6 server. The client sends a DHCPv6 SOLICIT message to the reserved IPv6 multicast all-DHCPv6-servers address FF02::1:2. This multicast address has link-local scope, which means routers do not forward the messages to other networks.

One or more DHCPv6 servers respond with a DHCPv6 ADVERTISE unicast message. The ADVERTISE message informs the DHCPv6 client that the server is available for DHCPv6 service.

The client responds with a DHCPv6 REQUEST or INFORMATION-REQUEST unicast message to the server, depending on whether it is using stateful or stateless DHCPv6.

- **Stateless DHCPv6 client** - The client sends a DHCPv6 INFORMATION-REQUEST message to the DHCPv6 server requesting only configuration parameters, such as DNS server address. The client generated its own IPv6 address using the prefix from the RA message and a self-generated Interface ID.
- **Stateful DHCPv6 client** - The client sends a DHCPv6 REQUEST message to the server to obtain an IPv6 address and all other configuration parameters from the server.

The server sends a DHCPv6 REPLY unicast message to the client containing the information requested in the REQUEST or INFORMATION-REQUEST message.

8.2.2 Stateless DHCPv6

8.2.2.1 Configuring a Router as a Stateless DHCPv6 Server

There are four steps to configure a router as a DHCPv6 server:

Step 1. Enable IPv6 Routing

The **ipv6 unicast-routing** command is required to enable IPv6 routing. This command is not necessary for the router to be a stateless DHCPv6 server, but it is required for the router to source ICMPv6 RA messages.

Step 2. Configure a DHCPv6 Pool

The **ipv6 dhcp pool** *pool-name* command creates a pool and enters the router in DHCPv6 configuration mode, which is identified by the Router(config-dhcpv6)# prompt.

Step 3. Configure Pool Parameters

During the SLAAC process, the client received the information it needed to create an IPv6 global unicast address. The client also received the default gateway information using the source IPv6 address from the RA message, which is the link-local address of the router. However, the stateless DHCPv6 server can be configured to provide other information that might not have been included in the RA message such as DNS server address and the domain name.

Step 4. Configure the DHCPv6 Interface

The **ipv6 dhcp server** *pool-name* interface configuration mode command binds the DHCPv6 pool to the interface. The router responds to stateless DHCPv6 requests on this interface with the information contained in the pool. The O flag needs to be changed from 0 to 1 using the interface command **ipv6 nd other-config-flag**. RA messages sent on this interface indicate that additional information is available from a stateless DHCPv6 server.

8.2.2.2 Configuring a Router as a Stateless DHCPv6 Client

The client router needs an IPv6 link-local address on the interface to send and receive IPv6 messages, such as RS messages and DHCPv6 messages. The link-local address of a router is created automatically when IPv6 is enabled on the interface. This can happen when a global unicast address is configured on the interface or by using the **ipv6 enable** command. After the router receives a link-local address, it can participate in IPv6 neighbor discovery.

In this example, the **ipv6 enable** command is used because the router does not yet have a global unicast address.

The **ipv6 address autoconfig** command enables automatic configuration of IPv6 addressing using SLAAC. By assumption, the server router is configured for stateless DHCPv6 so it sends an RA message to inform the client router to use stateless DHCPv6 to obtain DNS information.

8.2.2.3 Verifying Stateless DHCPv6

Verifying the Stateless DHCPv6 Server

The show ipv6 dhcp pool command verifies the name of the DHCPv6 pool and its parameters. The number of active clients is 0, because there is no state being maintained by the server.

The show running-config command can also be used to verify all the commands that were previously configured.

Verifying the Stateless DHCPv6 Client

The default router information is also from the RA message. This was the source IPv6 address of the packet that contained the RA message and the link-local address of the router.

The debug output displays all the DHCPv6 messages sent between the client and the server including the DNS server and domain name options that were configured on the server.

8.2.3 Stateful DHCPv6 Server

8.2.3.1 Configuring a Router as a Stateful DHCPv6 Server

Configuring a stateful DHCPv6 server is similar to configuring a stateless server. The most significant difference is that a stateful server also includes IPv6 addressing information similar to a DHCPv4 server.

Step 1. Enable IPv6 Routing

The **ipv6 unicast-routing** command is required to enable IPv6 routing. This command is not necessary for the router to be a stateful DHCPv6 server, but it is required for the router to source ICMPv6 RA messages.

Step 2. Configure a DHCPv6 Pool

The **ipv6 dhcp pool** *pool-name* command creates a pool and enters the router in DHCPv6 configuration mode, which is identified by the Router(config-dhcpv6)# prompt.

Step 3. Configure Pool Parameters

With stateful DHCPv6 all addressing and other configuration parameters must be assigned by the DHCPv6 server. The **address prefix** command is used to indicate the pool of addresses to be allocated by the server. The **lifetime** option indicates the valid and preferred lease times in seconds. As with stateless DHCPv6, the client uses the source IPv6 address from the packet that contained the RA message.

Other information provided by the stateful DHCPv6 server typically includes DNS server address and the domain name.

Step 4. Interface Commands

The **ipv6 dhcp server** *pool-name* interface command binds the DHCPv6 pool to the interface. The router responds to stateless DHCPv6 requests on this interface with the information contained in the pool. The M flag needs to be changed from 0 to 1 using the interface command **ipv6 nd managed-config-flag**. This informs the device not to use SLAAC but to obtain IPv6 addressing and all configuration parameters from a stateful DHCPv6 server.

8.2.3.2 Configuring a Router as a Stateful DHCPv6 Client

Use the **ipv6 enable** interface configuration mode command to allow the router to receive a link-local address to send RS messages and participate in DHCPv6.

The **ipv6 address dhcp** interface configuration mode command enables the router to behave as a DHCPv6 client on this interface.

8.2.3.3 Verifying Stateful DHCPv6

Verifying the Stateful DHCPv6 Server

The show ipv6 dhcp pool command verifies the name of the DHCPv6 pool and its parameters. The number of active clients is 1, which reflects client R3 receiving its IPv6 global unicast address from this server.

Verifying the Stateful DHCPv6 Client

The output from the show ipv6 interface command verifies the IPv6 global unicast address on DHCPv6 client R3 that was assigned by the DHCPv6 server. The default router information is not from the DHCPv6 server, but was determined by using the source IPv6 address from the RA message. Although the client does not use the information contained in the RA message, it is able to use the source IPv6 address for its default gateway information.

8.2.3.4 Configuring a Router as a DHCPv6 Relay Agent

If the DHCPv6 server is located on a different network than the client, then the IPv6 router can be configured as a DHCPv6 relay agent. The configuration of a DHCPv6 relay agent is similar to the configuration of an IPv4 router as a DHCPv4 relay.

Note: Although the configuration of a DHCPv6 relay agent is similar to DHCPv4, IPv6 router or relay agents forward DHCPv6 messages slightly differently than DHCPv4 relays. The messages and the process are beyond the scope of this curriculum.

Configuring the DHCPv6 Relay Agent

DHCPv6 relay agent is configured using the **ipv6 dhcp relay destination** command. This command is configured on the interface facing the DHCPv6 client using the address of the DHCPv6 server as the destination.

The **show ipv6 dhcp interface** command verifies the G0/0 interface is in relay mode with 2001:DB8:CAFE:1::6 configured as the DHCPv6 server.

8.2.4 Troubleshoot DHCPv6

8.2.4.1 Troubleshooting Tasks

Troubleshooting DHCPv6 is similar to troubleshooting DHCPv4.

Troubleshooting Task 1: Resolve Conflicts

Similar to IPv4 addresses, an IPv6 address lease can expire on a client that still needs to connect to the network. The **show ipv6 dhcp conflict** command displays any address conflicts logged by the stateful DHCPv6 server. If an IPv6 address conflict is detected, the client typically removes the address and generates a new address using either SLAAC or stateful DHCPv6.

Troubleshooting Task 2: Verify Allocation Method

The **show ipv6 interface** *interface* command can be used to verify the method of address allocation indicated in the RA message as indicated by the settings of the M and O flags. This information is displayed in the last lines of the output. If a client is not receiving its IPv6 address information from a stateful DHCPv6 server, it could be due to incorrect M and O flags in the RA message.

Troubleshooting Task 3: Test with a Static IPv6 Address

When troubleshooting any DHCP issue, whether it is DHCPv4 or DHCPv6, network connectivity can be verified by configuring a static IP address on a client workstation. In the case of IPv6, if the workstation is unable to reach network resources with a statically configured IPv6 address, the root cause of the problem is not SLAAC or DHCPv6. At this point, network connectivity troubleshooting is required.

Troubleshooting Task 4: Verify Switch Port Configuration

If the DHCPv6 client is unable to obtain information from a DHCPv6 server, verify that the switch port is enabled and is operating correctly.

Note: If there is a switch between the client and the DHCPv6 server, and the client is unable to obtain the DHCP configuration, switch port configuration issues may be the cause. These causes may include issues related to trunking, channeling, or spanning tree. PortFast and edge port configurations resolve the most common DHCPv6 client issues that occur with an initial installation of a Cisco switch.

Troubleshooting Task 5: Test DHCPv6 Operation on the Same Subnet or VLAN

If the stateless or stateful DHCPv6 server is functioning correctly, but is on a different IPv6 network or VLAN than the client, the problem may be with the DHCPv6 relay agent. The client facing interface on the router must be configured with the **ipv6 dhcp relay destination** command.

8.2.4.2 Verify Router DHCPv6 Configuration

The router configurations for stateless and stateful DHCPv6 services have many similarities but also include significant differences.

Stateful DHCPv6

A router configured for stateful DHCPv6 services has the **address prefix** command to provide addressing information.

For stateful DHCPv6 services the **ipv6 nd managed-config-flag** interface configuration mode command is used. In this instance, the client ignores the addressing information in the RA message and communicates with a DHCPv6 server for both addressing and other information.

CHAPTER 9: NAT FOR IPV4

All public IPv4 addresses that transverse the Internet must be registered with a Regional Internet Registry (RIR). Organizations can lease public addresses from a service provider. The registered holder of a public IP address can assign that address to a network device.

With a theoretical maximum of 4.3 billion addresses, IPv4 address space is severely limited. When Bob Kahn and Vint Cerf first developed the suite of TCP/IP protocols including IPv4 in 1981, they never envisioned what the Internet would become. At the time, the personal computer was mostly a curiosity for hobbyists and the World Wide Web was still more than a decade away.

With the proliferation of personal computing and the advent of the World Wide Web, it soon became obvious that 4.3 billion IPv4 addresses would not be enough. The long term solution was IPv6, but more immediate solutions to address exhaustion were required. For the short term, several solutions were implemented by the IETF including Network Address Translation (NAT) and RFC 1918 private IPv4 addresses. The chapter discusses how NAT, combined with the use of private address space, is used to both conserve and more efficiently use IPv4 addresses to provide networks of all sizes access to the Internet.

9.1 NAT Operation

9.1.1 NAT Characteristics

9.1.1.1 IPv4 Private Address Space

There are not enough public IPv4 addresses to assign a unique address to each device connected to the Internet. Networks are commonly implemented using private IPv4 addresses, as defined in RFC 1918.

These private addresses are used within an organization or site to allow devices to communicate locally. However, because these addresses do not identify any single company or organization, private IPv4 addresses cannot be routed over the Internet. To allow a device with a private IPv4 address to access devices and resources outside of the local network, the private address must first be translated to a public address.

NAT provides the translation of private addresses to public addresses. This allows a device with a private IPv4 address to access resources outside of their private network, such as those found on the Internet. NAT combined with private IPv4 addresses, has proven to be a useful method of preserving public IPv4 addresses. A single, public IPv4 address can be shared by hundreds, even thousands of devices, each configured with a unique private IPv4 address.

Without NAT, the exhaustion of the IPv4 address space would have occurred well before the year 2000. However, NAT has certain limitations, which will be explored later in this chapter. The solution to the exhaustion of IPv4 address space and the limitations of NAT is the eventual transition to IPv6.

9.1.1.2 What is NAT?

What is NAT?

NAT has many uses, but its primary use is to conserve public IPv4 addresses. It does this by allowing networks to use private IPv4 addresses internally and providing translation to a public address only when needed. NAT has an added benefit of adding a degree of privacy and security to a network, because it hides internal IPv4 addresses from outside networks.

NAT-enabled routers can be configured with one or more valid public IPv4 addresses. These public addresses are known as the NAT pool. When an internal device sends traffic out of the network, the NAT-enabled router translates the internal IPv4 address of the device to a public address from the NAT pool.

To outside devices, all traffic entering and exiting the network appears to have a public IPv4 address from the provided pool of addresses.

A NAT router typically operates at the border of a stub network. A stub network is a network that has a single connection to its neighboring network, one way in and one way out of the network.

When a device inside the stub network wants to communicate with a device outside of its network, the packet is forwarded to the border router. The border router performs the NAT process, translating the internal private address of the device to a public, outside, routable address.

Note: The connection to the ISP may use a private address or a public address that is shared among customers. For the purposes of this chapter, a public address is shown.

9.1.1.3 NAT Terminology

In NAT terminology, the inside network is the set of networks that is subject to translation. The outside network refers to all other networks.

When using NAT, IPv4 addresses have different designations based on whether they are on the private network, or on the public network (Internet), and whether the traffic is incoming or outgoing.

NAT includes four types of addresses:

- Inside local address
- Inside global address
- Outside local address
- Outside global address

When determining which type of address is used, it is important to remember that NAT terminology is always applied from the perspective of the device with the translated address:

- **Inside address** - The address of the device which is being translated by NAT.
- **Outside address** - The address of the destination device.

NAT also uses the concept of local or global with respect to addresses:

- **Local address** - A local address is any address that appears on the inside portion of the network.
- **Global address** - A global address is any address that appears on the outside portion of the network.

Note: The use of the outside local address is outside the scope of this course.

9.1.1.4 How NAT Works

In this example, PC1 with private address 192.168.10.10 wants to communicate with an outside web server with public address 209.165.201.1.

PC1 sends a packet addressed to the web server. The packet is forwarded by R1 to R2.

When the packet arrives at R2, the NAT-enabled router for the network, R2 reads the source IPv4 address of the packet to determine if the packet matches the criteria specified for translation.

In this case, the source IPv4 address does match the criteria and is translated from 192.168.10.10 (inside local address) to 209.165.200.226 (inside global address). R2 adds this mapping of the local to global address to the NAT table.

R2 sends the packet with the translated source address toward the destination.

The web server responds with a packet addressed to the inside global address of PC1 (209.165.200.226).

R2 receives the packet with destination address 209.165.200.226. R2 checks the NAT table and finds an entry for this mapping. R2 uses this information and translates the inside global address (209.165.200.226) to the inside local address (192.168.10.10), and the packet is forwarded toward PC1.

9.1.2 Types of NAT

9.1.2.1 Static NAT

There are three types of NAT translation:

- **Static address translation (static NAT)** - One-to-one address mapping between local and global addresses.
- **Dynamic address translation (dynamic NAT)** - Many-to-many address mapping between local and global addresses. Translations are made on an as-available basis; for example, if there are 100 inside local addresses and 10 inside global addresses, then at any given time only 10 of the 100 inside local addresses can be translated. This limitation of dynamic NAT makes it much less useful for production networks than port address translation.
- **Port Address Translation (PAT)** - Many-to-one address mapping between local and global addresses. This method is also known as overloading (NAT overloading). For example, if there are 100 inside local addresses and 10 inside global addresses, PAT uses ports as an additional parameter to provide a multiplier effect, making it possible to reuse any one of the 10 inside global addresses up to 65,536 times (depending on whether the flow is based on UDP, TCP, or ICMP).

Static NAT

Static NAT uses a one-to-one mapping of local and global addresses. These mappings are configured by the network administrator and remain constant.

Static NAT is particularly useful for web servers or devices that must have a consistent address that is accessible from the Internet, such as a company web server. It is also useful for devices that must be accessible by authorized personnel when offsite, but not by the general public on the Internet.

Static NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

9.1.2.2 Dynamic NAT

Dynamic NAT uses a pool of public addresses and assigns them on a first-come, first-served basis. When an inside device requests access to an outside network, dynamic NAT assigns an available public IPv4 address from the pool.

Similar to static NAT, dynamic NAT requires that enough public addresses are available to satisfy the total number of simultaneous user sessions.

9.1.2.3 Port Address Translation (PAT)

Port Address Translation (PAT), also known as NAT overload, maps multiple private IPv4 addresses to a single public IPv4 address or a few addresses. This is what most home routers do. The ISP assigns one address to the router, yet several members of the household can simultaneously access the Internet. This is the most common form of NAT.

With PAT, multiple addresses can be mapped to one or to a few addresses, because each private address is also tracked by a port number. When a device initiates a TCP/IP session, it generates a TCP or UDP source port value or a specially assigned query ID for ICMP, to uniquely identify the session. When the NAT router receives a packet from the client, it uses its source port number to uniquely identify the specific NAT translation.

PAT ensures that devices use a different TCP port number for each session with a server on the Internet. When a response comes back from the server, the source port number, which becomes the destination port number on the return trip, determines to which device the router forwards the packets. The PAT process also validates that the incoming packets were requested, thus adding a degree of security to the session.

9.1.2.4 Next Available Port

PAT attempts to preserve the original source port. However, if the original source port is already used, PAT assigns the first available port number starting from the beginning of the appropriate port group 0–511, 512–1,023, or 1,024–65,535. When there are no more ports available and there is more than one external address in the address pool, PAT moves to the next address to try to allocate the original source port. This process continues until there are no more available ports or external IPv4 addresses.

9.1.2.5 Comparing NAT and PAT

Summarizing the differences between NAT and PAT helps your understanding of each.

NAT		PAT	
Inside Global Address Pool	Inside Local Address	Inside Global Address	Inside Local Address
209.165.200.226	192.168.10.10	209.165.200.226:1444	192.168.10.10:1444
209.165.200.227	192.168.10.11	209.165.200.226:1445	192.168.10.11:1444
209.165.200.228	192.168.10.12	209.165.200.226:1555	192.168.10.12:1555
209.165.200.229	192.168.10.13	209.165.200.226:1556	192.168.10.13:1555

As the figure shows, NAT translates IPv4 addresses on a 1:1 basis between private IPv4 addresses and public IPv4 addresses. However, PAT modifies both the address and the port number.

NAT forwards incoming packets to their inside destination by referring to the incoming source IPv4 address given by the host on the public network. With PAT, there is generally only one or a very few publicly exposed IPv4 addresses. Incoming packets from the public network are routed to their destinations on the private network by referring to a table in the NAT router. This table tracks public and private port pairs. This is called connection tracking.

Packets without a Layer 4 Segment

What about IPv4 packets carrying data other than a TCP or UDP segment? These packets do not contain a Layer 4 port number. PAT translates most common protocols carried by IPv4 that do not use TCP or UDP as a transport layer protocol. The most common of these is ICMPv4. Each of these types of protocols is handled differently by PAT. For example, ICMPv4 query messages, echo requests, and echo replies include a Query ID. ICMPv4 uses the Query ID to identify an echo request with its corresponding echo reply. The Query ID is incremented with each echo request sent. PAT uses the Query ID instead of a Layer 4 port number.

Note: Other ICMPv4 messages do not use the Query ID. These messages and other protocols that do not use TCP or UDP port numbers vary and are beyond the scope of this curriculum.

9.1.3 NAT Advantages

9.1.3.1 Advantages of NAT

NAT provides many benefits, including:

- NAT conserves the legally registered addressing scheme by allowing the privatization of intranets. NAT conserves addresses through application port-level multiplexing. With NAT overload, internal hosts can share a single public IPv4 address for all external communications. In this type of configuration, very few external addresses are required to support many internal hosts.
- NAT increases the flexibility of connections to the public network. Multiple pools, backup pools, and load-balancing pools can be implemented to ensure reliable public network connections.

- NAT provides consistency for internal network addressing schemes. On a network not using private IPv4 addresses and NAT, changing the public IPv4 address scheme requires the readdressing of all hosts on the existing network. The costs of readdressing hosts can be significant. NAT allows the existing private IPv4 address scheme to remain while allowing for easy change to a new public addressing scheme. This means an organization could change ISPs and not need to change any of its inside clients.
- NAT hides user IPv4 addresses. Using RFC 1918 IPv4 addresses, NAT provides the side effect of hiding users and other devices' IPv4 addresses. Some people consider this a security feature, however most experts agree that NAT does not provide security. A stateful firewall is what provides security on the edge of the network.

9.1.3.2 Disadvantages of NAT

NAT does have some drawbacks. The fact that hosts on the Internet appear to communicate directly with the NAT-enabled device, rather than with the actual host inside the private network, creates a number of issues.

One disadvantage of using NAT is related to network performance, particularly for real time protocols such as VoIP. NAT increases forwarding delays because the translation of each IPv4 address within the packet headers takes time. The first packet is always process-switched going through the slower path. The router must look at every packet to decide whether it needs translation. The router must alter the IPv4 header, and possibly alter the TCP or UDP header. The IPv4 header checksum, along with the TCP or UDP checksum must be recalculated each time a translation is made. Remaining packets go through the fast-switched path if a cache entry exists; otherwise, they too are delayed.

Another disadvantage of using NAT is that end-to-end addressing is lost. Many Internet protocols and applications depend on end-to-end addressing from the source to the destination. Some applications do not work with NAT. For example, some security applications, such as digital signatures, fail because the source IPv4 address changes before reaching the destination. Applications that use physical addresses, instead of a qualified domain name, do not reach destinations that are translated across the NAT router. Sometimes, this problem can be avoided by implementing static NAT mappings.

End-to-end IPv4 traceability is also lost. It becomes much more difficult to trace packets that undergo numerous packet address changes over multiple NAT hops, making troubleshooting challenging.

Using NAT also complicates the use of tunneling protocols, such as IPsec, because NAT modifies values in the headers, causing integrity checks to fail.

Services that require the initiation of TCP connections from the outside network, or stateless protocols, such as those using UDP, can be disrupted. Unless the NAT router has been configured to support such protocols, incoming packets cannot reach their destination. Some protocols can accommodate one instance of NAT between participating hosts (passive mode FTP, for example), but fail when both systems are separated from the Internet by NAT.

9.2 Configure NAT

9.2.1 Configuring Static NAT

9.2.1.1 Configure Static NAT

Static NAT is a one-to-one mapping between an inside address and an outside address. Static NAT allows external devices to initiate connections to internal devices using the statically assigned public address. For instance, an internal web server may be mapped to a specific inside global address so that it is accessible from outside networks.

There are two basic tasks when configuring static NAT translations.

Step 1. The first task is to create a mapping between the inside local address and the inside global addresses.

Step 2. After the mapping is configured, the interfaces participating in the translation are configured as inside or outside relative to NAT.

9.2.1.2 Verifying Static NAT

A useful command to verify NAT operation is **show ip nat translations**. This command shows active NAT translations. Static translations, unlike dynamic translations, are always in the NAT table.

Another useful command is **show ip nat statistics**. The **show ip nat statistics** command displays information about the total number of active translations, NAT configuration parameters, the number of addresses in the pool, and the number of addresses that have been allocated.

To verify that the NAT translation is working, it is best to clear statistics from any past translations using the **clear ip nat statistics** command before testing.

Prior to any communications with the web server, the **show ip nat statistics** command shows no current hits.

9.2.2 Configure Dynamic NAT

9.2.2.1 Dynamic NAT Operation

While static NAT provides a permanent mapping between an inside local address and an inside global address, dynamic NAT allows the automatic mapping of inside local addresses to inside global addresses. These inside global addresses are typically public IPv4 addresses. Dynamic NAT uses a group, or pool of public IPv4 addresses for translation.

Dynamic NAT, like static NAT, requires the configuration of the inside and outside interfaces participating in NAT. However, where static NAT creates a permanent mapping to a single address, dynamic NAT uses a pool of addresses.

Note: Translating between public and private IPv4 addresses is by far the most common use of NAT. However, NAT translations can occur between any pair of addresses.

The pool of public IPv4 addresses (inside global address pool) is available to any device on the inside network on a first-come first-served basis. With dynamic NAT, a single inside address is translated to a single outside address. With this type of translation there must be enough addresses in the pool to accommodate all the inside devices needing access to the outside network at the same time. If all of the addresses in the pool have been used, a device must wait for an available address before it can access the outside network.

9.2.2.2 Configuring Dynamic NAT

The steps and the commands used to configure dynamic NAT.

Step 1. Define the pool of addresses that will be used for translation using the **ip nat pool** command. This pool of addresses is typically a group of public addresses. The addresses are defined by indicating the starting IPv4 address and the ending IPv4 address of the pool. The **netmask** or **prefix-length** keyword indicates which address bits belong to the network and which bits belong to the host for the range of addresses.

Step 2. Configure a standard ACL to identify (permit) only those addresses that are to be translated. An ACL that is too permissive can lead to unpredictable results. Remember there is an implicit **deny all** statement at the end of each ACL.

Step 3. Bind the ACL to the pool. The **ip nat inside source list** *access-list-number* **pool** *pool name* command is used to bind the ACL to the pool. This configuration is used by the router to identify which devices (**list**) receive which addresses (**pool**).

Step 4. Identify which interfaces are inside, in relation to NAT; that is, any interface that connects to the inside network.

Step 5. Identify which interfaces are outside, in relation to NAT; that is, any interface that connects to the outside network.

9.2.2.3 Verifying Dynamic NAT

The output of the **show ip nat translations** command displays the details of the two previous NAT assignments. The command displays all static translations that have been configured and any dynamic translations that have been created by traffic.

Adding the **verbose** keyword displays additional information about each translation, including how long ago the entry was created and used.

By default, translation entries time out after 24 hours, unless the timers have been reconfigured with the **ip nat translation timeout** *timeout-seconds* command in global configuration mode.

To clear dynamic entries before the timeout has expired, use the **clear ip nat translation** privileged EXEC mode command. It is useful to clear the dynamic entries when testing the NAT configuration. As shown in the table, this command can be used with keywords and variables to control which entries are cleared. Specific entries can be cleared to avoid disrupting active sessions. Use the **clear ip nat translation *** privileged EXEC command to clear all translations from the table.

Note: Only the dynamic translations are cleared from the table. Static translations cannot be cleared from the translation table.

The **show ip nat statistics** command displays information about the total number of active translations, NAT configuration parameters, the number of addresses in the pool, and how many of the addresses have been allocated.

Alternatively, use the **show running-config** command and look for NAT, ACL, interface, or pool commands with the required values. Examine these carefully and correct any errors discovered.

9.2.3 Configure PAT

9.2.3.1 Configuring PAT: Address Pool

PAT (also called NAT overload) conserves addresses in the inside global address pool by allowing the router to use one inside global address for many inside local addresses. In other words, a single public IPv4 address can be used for hundreds, even thousands of internal private IPv4 addresses. When this type of translation is configured, the router maintains enough information from higher-level protocols, TCP or UDP port numbers, for example, to translate the inside global address back into the correct inside local address. When multiple inside local addresses map to one inside global address, the TCP or UDP port numbers of each inside host distinguish between the local addresses.

Note: The total number of internal addresses that can be translated to one external address could theoretically be as high as 65,536 per IPv4 address. However, the number of internal addresses that can be assigned a single IPv4 address is around 4,000.

There are two ways to configure PAT, depending on how the ISP allocates public IPv4 addresses. In the first instance, the ISP allocates more than one public IPv4 address to the organization, and in the other, it allocates a single public IPv4 address that is required for the organization to connect to the ISP.

Configuring PAT for a Pool of Public IPv4 Addresses

If a site has been issued more than one public IPv4 address, these addresses can be part of a pool that is used by PAT. This is similar to dynamic NAT, except that there are not enough public addresses for a one-to-one mapping of inside to outside addresses. The small pool of addresses is shared among a larger number of devices.

Step 1	Define a pool of global addresses to be used for overload translation. `ip nat pool name start-ip end-ip {netmask netmask | prefix-length prefix-length}`
Step 2	Define a standard access list permitting the addresses that should be translated. `access-list access-list-number permit source [source-wildcard]`
Step 3	Establish overload translation, specifying the access list and pool defined in prior steps. `ip nat inside source list access-list-number pool name overload`

Step 4	Identify the inside interface.
	
	`interface type number`
	`ip nat inside`

Step 5	Identify the outside interface.
	
	`interface type number`
	`ip nat outside`

Figure above shows the steps to configure PAT to use a pool of addresses. The primary difference between this configuration and the configuration for dynamic, one-to-one NAT is that the **overload** keyword is used. The **overload** keyword enables PAT.

9.2.3.2 Configuring PAT: Single Address

In the example, all hosts from network 192.168.0.0/16 (matching ACL 1) that send traffic through router R2 to the Internet will be translated to IPv4 address 209.165.200.225. The traffic flows will be identified by port numbers in the NAT table because the overload keyword was used.

If only a single public IPv4 address is available, the overload configuration typically assigns the public address to the outside interface that connects to the ISP. All inside addresses are translated to the single IPv4 address when leaving the outside interface.

Step 1. Define an ACL to permit the traffic to be translated.

Step 2. Configure source translation using the interface and overload keywords. The interface keyword identifies which interface IPv4 address to use when translating inside addresses. The overload keyword directs the router to track port numbers with each NAT entry.

Step 3. Identify which interfaces are inside in relation to NAT. That is any interface that connects to the inside network.

Step 4. Identify which interface is outside in relation to NAT. This should be the same interface identified in the source translation statement from Step 2.

The configuration is similar to dynamic NAT, except that instead of a pool of addresses, the interface keyword is used to identify the outside IPv4 address. Therefore, no NAT pool is defined.

9.2.3.3 Verifying PAT

Router R2 has been configured to provide PAT to the 192.168.0.0/16 clients. When the internal hosts exit router R2 to the Internet, they are translated to an IPv4 address from the PAT pool with a unique source port number.

The same commands used to verify static and dynamic NAT are used to verify PAT. The **show ip nat translations** command displays the translations from two different hosts to different web servers. Notice that two different inside hosts are allocated the same IPv4 address of 209.165.200.226 (inside global address). The source port numbers in the NAT table differentiate the two transactions.

The **show ip nat statistics** command verifies that NAT-POOL2 has allocated a single address for both translations. Included in the output is information about the number and type of active translations, NAT configuration parameters, the number of addresses in the pool, and how many have been allocated.

9.2.4 Configure Port Forwarding

9.2.4.1 Port Forwarding

Port forwarding is the act of forwarding traffic addressed to a specific network port from one network node to another. This technique allows an external user to reach a port on a private IPv4 address (inside a LAN) from the outside, through a NAT-enabled router.

Typically, peer-to-peer file-sharing programs and operations, such as web serving and outgoing FTP, require that router ports be forwarded or opened to allow these applications to work. Because NAT hides internal addresses, peer-to-peer only works from the inside out where NAT can map outgoing requests against incoming replies.

The problem is that NAT does not allow requests initiated from the outside. This situation can be resolved with manual intervention. Port forwarding can be configured to identify specific ports that can be forwarded to inside hosts.

Recall that Internet software applications interact with user ports that need to be open or available to those applications. Different applications use different ports. This makes it predictable for applications and routers to identify network services. For example, HTTP operates through the well-known port 80. When someone enters the **http://cisco.com** address, the browser displays the Cisco Systems, Inc. website. Notice that they do not have to specify the HTTP port number for the page request, because the application assumes port 80.

If a different port number is required, it can be appended to the URL separated by a colon (:). For example, if the web server is listening on port 8080, the user would type **http://www.example.com:8080**.

Port forwarding allows users on the Internet to access internal servers by using the WAN port address of the router and the matched external port number. The internal servers are typically configured with RFC 1918 private IPv4 addresses. When a request is sent to the IPv4 address of the WAN port via the Internet, the router forwards the request to the appropriate server on the LAN. For security reasons, broadband routers do not by default permit any external network request to be forwarded to an inside host.

9.2.4.2 Configuring Port Forwarding with IOS

Implementing port forwarding with IOS commands is similar to the commands used to configure static NAT. Port forwarding is essentially a static NAT translation with a specified TCP or UDP port number.

Figure below shows the static NAT command used to configure port forwarding using IOS.

```
ip nat inside source {static {tcp | udp local-ip local-port
global-ip global-port} [extendable]
```

Parameter	Description
tcp or udp	Indicates if this is a TCP or UDP port number.
local-ip	This is the IPv4 address assigned to the host on the inside network, typically from RFC 1918 private address space.
local-port	Sets the local TCP/UDP port in a range from 1 – 65,535. This is the port number the server is listening on.
global-ip	This is the IPv4 globally unique IP address of an inside host. This is the IP address the outside clients will use to reach the internal server.
global-port	Sets the global TCP/UDP port in a range from 1 – 65,535. This is the port number the outside client will use to reach the internal server.
extendable	The extendable option is applied automatically. The extendable keyword allows the user to configure several ambiguous static translations, where ambiguous translations are translations with the same local or global address. It allows the

When a well-known port number is not being used, the client must specify the port number in the application.

Like other types of NAT, port forwarding requires the configuration of both the inside and outside NAT interfaces.

Similar to static NAT, the **show ip nat translations** command can be used to verify the port forwarding.

9.2.5 NAT and IPv6

9.2.5.1 NAT for IPv6?

Since the early 1990s, the concern about the depletion of IPv4 address space has been a priority of the IETF. The combination of RFC 1918 private IPv4 addresses and NAT has been instrumental in slowing this depletion. NAT has significant disadvantages, and in January of 2011, IANA allocated the last of its IPv4 addresses to RIRs.

One of the unintentional benefits of NAT for IPv4 is that it hides the private network from the public Internet. NAT has the advantage of providing a perceived level of security by denying computers in the public Internet from accessing internal hosts. However, it should not be considered a substitute for proper network security, such as that provided by a firewall.

In RFC 5902, the Internet Architecture Board (IAB) included the following quote concerning IPv6 network address translation:

“It is commonly perceived that a NAT box provides one level of protection because external hosts cannot directly initiate communication with hosts behind a NAT. However, one should not confuse NAT boxes with firewalls. As discussed Section 2.2 in RFC4864, the act of translation does not provide security in itself. The stateful filtering function can provide the same level of protection without requiring a translation function.”

IPv6, with a 128-bit address, provides 340 undecillion addresses. Therefore, address space is not an issue. IPv6 was developed with the intention of making NAT for IPv4 with its translation between public and private IPv4 addresses unnecessary. However, IPv6 does implement a form of NAT. IPv6 includes both its own IPv6 private address space and NAT, which are implemented differently than they are for IPv4.

9.2.5.2 IPv6 Unique Local Addresses

IPv6 unique local addresses (ULA) are similar to RFC 1918 private addresses in IPv4, but there are significant differences as well. The intent of ULA is to provide IPv6 address space for communications within a local site; it is not meant to provide additional IPv6 address space, nor is it meant to provide a level of security.

Unique local addresses are defined in RFC 4193. ULAs are also known as local IPv6 addresses (not to be confused with IPv6 link-local addresses) and have several characteristics including:

- Allows sites to be combined or privately interconnected, without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- Independent of any ISP and can be used for communications within a site without having any Internet connectivity.
- Not routable across the Internet, however, if accidentally leaked by routing or DNS, there is not conflict with other addresses.

ULA is not quite as straight-forward as RFC 1918 addresses. Unlike private IPv4 addresses, it has not been the intention of the IETF to use a form of NAT to translate between unique local addresses and IPv6 global unicast addresses.

The implementation and potential uses for IPv6 unique local addresses are still being examined by the Internet community. For example, the IETF is considering allowing the option of having the 40-bit global ID centrally assigned when using the FC00::/8 ULA prefix, and the 40-bit global ID randomly generated, or perhaps manually assigned, when using the ULA prefix FD00::/8. The rest of the address remains the same. We still use 16 bits for the subnet ID and 64 bits for the interface ID.

Note: The original IPv6 specification allocated address space for site-local addresses, defined in RFC 3513. Site-local addresses have been deprecated by the IETF in RFC 3879 because the term "site" was somewhat ambiguous. Site-local addresses had the prefix range of FEC0::/10 and may still be found in some older IPv6 documentation.

9.2.5.3 NAT for IPv6

NAT for IPv6 is used in a much different context than NAT for IPv4. The varieties of NAT for IPv6 are used to transparently provide access between IPv6-only and IPv4-only networks. It is not used as a form of private IPv6 to global IPv6 translation.

Ideally, IPv6 should be run natively wherever possible. This means IPv6 devices communicating with each other over IPv6 networks. However, to aid in the move from IPv4 to IPv6, the IETF has developed several transition techniques to accommodate a variety of IPv4-to-IPv6 scenarios, including dual-stack, tunneling, and translation.

Dual-stack is when the devices are running protocols associated with both the IPv4 and IPv6. Tunneling for IPV6 is the process of encapsulating an IPv6 packet inside an IPv4 packet. This allows the IPv6 packet to be transmitted over an IPv4-only network.

NAT for IPv6 should not be used as a long term strategy, but as a temporary mechanism to assist in the migration from IPv4 to IPv6. Over the years, there have been several types of NAT for IPv6 including Network Address Translation-Protocol Translation (NAT-PT). NAT-PT has been deprecated by IETF in favor of its replacement, NAT64. NAT64 is beyond the scope of this curriculum.

9.3 Troubleshoot NAT

9.3.1 NAT Troubleshooting Commands

9.3.1.1 The show ip nat Commands

When there are IPv4 connectivity problems in a NAT environment, it is often difficult to determine the cause of the problem. The first step in solving the problem is to rule out NAT as the cause. Follow these steps to verify that NAT is operating as expected:

Step 1. Based on the configuration, clearly define what NAT is supposed to achieve. This may reveal a problem with the configuration.

Step 2. Verify that correct translations exist in the translation table using the **show ip nat translations** command.

Step 3. Use the **clear** and **debug** commands to verify that NAT is operating as expected. Check to see if dynamic entries are recreated after they are cleared.

Step 4. Review in detail what is happening to the packet, and verify that routers have the correct routing information to move the packet.

In a simple network environment, it is useful to monitor NAT statistics with the **show ip nat statistics** command. The **show ip nat statistics** command displays information about the total number of active translations, NAT configuration parameters, the number of addresses in the pool, and the number that have been allocated. However, in a more complex NAT environment, with several translations taking place, this command may not clearly identify the issue. It may be necessary to run **debug** commands on the router.

9.3.1.2 The debug ip nat Command

Use the **debug ip nat** command to verify the operation of the NAT feature by displaying information about every packet that is translated by the router. The **debug ip nat detailed** command generates a description of each packet considered for translation. This command also provides information about certain errors or exception conditions, such as the failure to allocate a global address. The **debug ip nat detailed** command generates more overhead than the **debug ip nat** command, but it can provide the detail that may be needed to troubleshoot the NAT problem. Always turn off debugging when finished.

When decoding the debug output, note what the following symbols and values indicate:

- ***(asterisk)** - The asterisk next to NAT indicates that the translation is occurring in the fast-switched path. The first packet in a conversation is always process-switched, which is slower. The remaining packets go through the fast-switched path if a cache entry exists.
- **s=** - This symbol refers to the source IPv4 address.
- **a.b.c.d--->w.x.y.z** - This value indicates that source address a.b.c.d is translated to w.x.y.z.
- **d=** - This symbol refers to the destination IPv4 address.
- **[xxxx]** - The value in brackets is the IPv4 identification number. This information may be useful for debugging in that it enables correlation with other packet traces from protocol analyzers.

Note: Verify that the ACL referenced in the NAT command is permitting all of the necessary networks.

CHAPTER 10: DEVICE DISCOVERY, MANAGEMENT, AND MAINTENANCE

In this chapter, you will explore the tools network administrators can use for device discovery, device management, and device maintenance. Cisco Discovery Protocol (CDP) and Link Layer Discover Protocol (LLDP) are both capable of discovering information about directly connected devices.

Network Time Protocol (NTP) can be effectively used to synchronize the time across all your networking devices, which is especially important when trying to compare log files from different devices. Those log files are generated by the syslog protocol. Syslog messages can be captured and sent to a syslog server to aid in device management tasks.

Device maintenance includes ensuring that Cisco IOS images and configuration files are backed up in a safe location in the event that the device memory is corrupted or erased, either maliciously or inadvertently. Maintenance also includes keeping the IOS image up to date. The device maintenance section of the chapter includes topics for file maintenance, image management, and software licensing.

10.1 Device Discovery
10.1.1 Device Discovery with CDP

10.1.1.1 CDP Overview
Cisco Discovery Protocol (CDP) is a Cisco proprietary Layer 2 protocol that is used to gather information about Cisco devices which share the same data link. CDP is media and protocol independent and runs on all Cisco devices, such as routers, switches, and access servers.

The device sends periodic CDP advertisements to connected devices. These advertisements share information about the type of device that is discovered, the name of the devices, and the number and type of the interfaces.

Because most network devices are connected to other devices, CDP can assist in network design decisions, troubleshooting, and making changes to equipment. CDP can also be used as a network discovery tool to determine the information about the neighboring devices. This information gathered from CDP can help build a logical topology of a network when documentation is missing or lacking in detail.

10.1.1.2 Configure and Verify CDP
For Cisco devices, CDP is enabled by default. For security reasons, it may be desirable to disable CDP on a network device globally, or per interface. With CDP, an attacker can gather valuable insight about the network layout, such as IP addresses, IOS versions, and types of devices.

To verify the status of CDP and display information about CDP, enter the **show cdp** command.

To enable CDP globally for all the supported interfaces on the device, enter **cdp run** in the global configuration mode. CDP can be disabled for all the interfaces on the device with the **no cdp run** command in the global configuration mode.

To disable CDP on a specific interface, such as the interface facing an ISP, enter **no cdp enable** in the interface configuration mode. CDP is still enabled on the device; however, no more CDP advertisements will be sent out that interface. To enable CDP on the specific interface again, enter **cdp enable**.

To verify the status of CDP and display a list of neighbors, use the **show cdp neighbors** command in the privileged EXEC mode. The **show cdp neighbors** command displays important information about the CDP neighbors. Currently, this device does not have any neighbors because it is not physically connected to any devices, as indicated by the results of the **show cdp neighbors** command.

Use the **show cdp interface** command to display the interfaces that are CDP enabled on a device. The status of each interface is also displayed.

10.1.1.3 Discover Devices Using CDP

With CDP enabled on the network, the **show cdp neighbors** command can be used to determine the network layout.

No information is available regarding the rest of the network. The **show cdp neighbors** command provides helpful information about each CDP neighbor device, including the following:

- **Device identifiers** - The host name of the neighbor device (S1)
- **Port identifier** - The name of the local and remote port (Gig 0/1 and Fas 0/5, respectively)
- **Capabilities list** - Whether the device is a router or a switch (S for switch; I for IGMP is beyond scope for this course)
- **Platform** - The hardware platform of the device (WS-C2960 for Cisco 2960 switch)

If more information is needed, the **show cdp neighbors** detail command can also provide information, such as the neighbors' IOS version and IPv4 address.

10.1.2 Device Discovery with LLDP

10.1.2.1 LLDP Overview

Cisco devices also support Link Layer Discovery Protocol (LLDP), which is a vendor-neutral neighbor discovery protocol similar to CDP. LLDP works with network devices, such as routers, switches, and wireless LAN access points. This protocol advertises its identity and capabilities to other devices and receives the information from a physically connected Layer 2 device.

10.1.2.2 Configure and Verify LLDP

Depending on the device, LLDP may be enabled by default. To enable LLDP globally on a Cisco network device, enter the **lldp run** command in the global configuration mode. To disable LLDP, enter the **no lldp run** command in the global configuration mode.

Similar to CDP, LLDP can be configured on specific interfaces. However, LLDP must be configured separately to transmit and receive LLDP packets.

To verify LLDP has been enabled on the device, enter the **show lldp** command in the privileged EXEC mode.

10.1.2.3 Discover Devices Using LLDP

With LLDP enabled, device neighbors can be discovered using the **show lldp neighbors** command.

When more details about the neighbors are needed, the **show lldp neighbors detail** command can provide information, such as the neighbors' IOS version, IP address, and device capability.

10.2 Device Management

10.2.1 NTP

10.2.1.1 Setting the System Clock

The software clock on a router or switch starts when the system boots and is the primary source of time for the system. It is important to synchronize the time across all devices on the network because all aspects of managing, securing, troubleshooting, and planning networks require accurate timestamping. When the time is not synchronized between devices, it will be impossible to determine the order of the events and the cause of an event.

Typically, the date and time settings on a router or switch can be set using one of two methods:

- Manually configure the date and time
- Configure the Network Time Protocol (NTP)

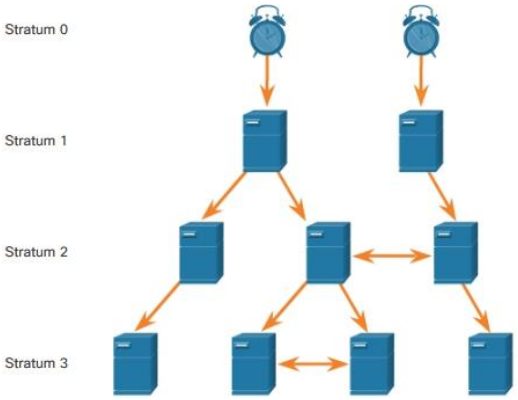
As a network grows, it becomes difficult to ensure that all infrastructure devices are operating with synchronized time. Even in a smaller network environment, the manual method is not ideal. If a router reboots, how will it get an accurate date and timestamp?

A better solution is to configure the NTP on the network. This protocol allows routers on the network to synchronize their time settings with an NTP server. A group of NTP clients that obtain time and date information from a single source have more consistent time settings. When NTP is implemented in the network, it can be set up to synchronize to a private master clock or it can synchronize to a publicly available NTP server on the Internet.

NTP uses UDP port 123 and is documented in RFC 1305.

10.2.1.2 NTP Operation

NTP networks use a hierarchical system of time sources. Each level in this hierarchical system is called a stratum. The stratum level is defined as the number of hop counts from the authoritative source. The synchronized time is distributed across the network using NTP. The figure displays a sample NTP network.



NTP servers arranged in three levels showing the three strata. Stratum 1 is connected to Stratum 0 clocks.

Stratum 0

An NTP network gets the time from authoritative time sources. These authoritative time sources, also referred to as stratum 0 devices, are high-precision timekeeping devices assumed to be accurate and with little or no delay associated with them. Stratum 0 devices are represented by the clock in the figure.

Stratum 1

The stratum 1 devices are directly connected to the authoritative time sources. They act as the primary network time standard.

Stratum 2 and Lower

The stratum 2 servers are connected to stratum 1 devices through network connections. Stratum 2 devices, such as NTP clients, synchronize their time using the NTP packets from stratum 1 servers. They could also act as servers for stratum 3 devices.

Smaller stratum numbers indicate that the server is closer to the authorized time source than larger stratum numbers. The larger the stratum number, the lower the stratum level. The max hop count is 15. Stratum 16, the lowest stratum level, indicates that a device is unsynchronized. Time servers on the same stratum level can be configured to act as a peer with other time servers on the same stratum level for backup or verification of time.

10.2.1.3 Configure and Verify NTP

Before NTP is configured on the network, the **show clock** command displays the current time on the software clock. With the **detail** option, the time source is also displayed. Use the **ntp server** *ip-address*

command in global configuration mode to configure 209.165.200.225 as the NTP server for R1. To verify the time source is set to NTP, use the **show clock detail** command again.

se the **show ip ntp associations** and **show ntp status** commands to verify that R1 is synchronized with the NTP server at 209.165.200.225. Notice that R1 is synchronized with a stratum 1 NTP server at 209.165.200.225, which is synchronized with a GPS clock. The **show ntp status** command displays that R1 is now a stratum 2 device synchronized with the NTP server at 209.165.220.225.

The clock on S1 is configured to synchronize to R1, as shown in Figure 3. Output from the **show ntp associations** command verifies that the clock on S1 is now synchronized with R1 at 192.168.1.1 via NTP. R1 is a stratum 2 device and NTP server to S1. Now S1 is a stratum 3 device that can provide NTP service to other devices in the network, such as end devices.

10.2.2 Syslog Operation

10.2.2.1 Introduction to Syslog

When certain events occur on a network, networking devices have trusted mechanisms to notify the administrator with detailed system messages. These messages can be either non-critical or significant. Network administrators have a variety of options for storing, interpreting, and displaying these messages, and for being alerted to those messages that could have the greatest impact on the network infrastructure.

The most common method of accessing system messages is to use a protocol called syslog.

Syslog is a term used to describe a standard. It is also used to describe the protocol developed for that standard. The syslog protocol was developed for UNIX systems in the 1980s, but was first documented as RFC 3164 by IETF in 2001. Syslog uses UDP port 514 to send event notification messages across IP networks to event message collectors.

Many networking devices support syslog, including: routers, switches, application servers, firewalls, and other network appliances. The syslog protocol allows networking devices to send their system messages across the network to syslog servers.

There are several different syslog server software packages for Windows and UNIX. Many of them are freeware.

The syslog logging service provides three primary functions:

- The ability to gather logging information for monitoring and troubleshooting
- The ability to select the type of logging information that is captured
- The ability to specify the destinations of captured syslog messages

10.2.2.2 Syslog Operation

On Cisco network devices, the syslog protocol starts by sending system messages and **debug** output to a local logging process internal to the device. How the logging process manages these messages and outputs is based on device configurations. For example, syslog messages may be sent across the network to an external syslog server. These messages can be retrieved without the need of accessing the actual device. Log messages and outputs stored on the external server can be pulled into various reports for easier reading.

Alternatively, syslog messages may be sent to an internal buffer. Messages sent to the internal buffer are only viewable through the CLI of the device.

Finally, the network administrator may specify that only certain types of system messages are sent to various destinations. For example, the device may be configured to forward all system messages to an external syslog server. However, debug-level messages are forwarded to the internal buffer and are only accessible by the administrator from the CLI.

Popular destinations for syslog messages include:

- Logging buffer (RAM inside a router or switch)
- Console line
- Terminal line
- Syslog server

It is possible to remotely monitor system messages by viewing the logs on a syslog server, or by accessing the device through Telnet, SSH, or through the console port.

10.2.2.3 Syslog Message Format

Cisco devices produce syslog messages as a result of network events. Every syslog message contains a severity level and a facility.

The smaller numerical levels are the more critical syslog alarms. The severity level of the messages can be set to control where each type of message is displayed (i.e. on the console or the other destinations). The complete list of syslog levels is shown in Figure below.

Severity Name	Severity Level	Explanation
Emergency	Level 0	System Unusable
Alert	Level 1	Immediate Action Needed
Critical	Level 2	Critical Condition
Error	Level 3	Error Condition
Warning	Level 4	Warning Condition
Notification	Level 5	Normal, but Significant Condition
Informational	Level 6	Informational Message
Debugging	Level 7	Debugging Message

Each syslog level has its own meaning:

- **Warning Level 4 - Emergency Level 0**: These messages are error messages about software or hardware malfunctions; these types of messages mean that the functionality of the device is affected. The severity of the issue determines the actual syslog level applied.
- **Notification Level 5**: The notifications level is for normal, but significant events. For example, interface up or down transitions, and system restart messages are displayed at the notifications level.
- **Informational Level 6**: A normal information message that does not affect device functionality. For example, when a Cisco device is booting, you might see the following informational message: %LICENSE-6-EULA_ACCEPT_ALL: The Right to Use End User License Agreement is accepted.
- **Debugging Level 7**: This level indicates that the messages are output generated from issuing various **debug** commands.

In addition to specifying the severity, syslog messages also contain information on the facility. Syslog facilities are service identifiers that identify and categorize system state data for error and event message reporting. The logging facility options that are available are specific to the networking device. For example, Cisco 2960 Series switches running Cisco IOS Release 15.0(2) and Cisco 1941 routers running Cisco IOS Release 15.2(4) support 24 facility options that are categorized into 12 facility types.

Some common syslog message facilities reported on Cisco IOS routers include:

- IP
- OSPF protocol
- SYS operating system
- IP security (IPsec)
- Interface IP (IF)

By default, the format of syslog messages on the Cisco IOS Software is as follows:

seq no: timestamp: %facility-severity-MNEMONIC: description

The fields contained in the Cisco IOS Software syslog message are explained in Figure below.

Field	Explanation
seq no	Stamps log messages with a sequence number only if the `service sequence-numbers` global configuration command is configured.
timestamp	Date and time of the message or event, which appears only if the `service timestamps` global configuration command is configured.
facility	The facility to which the message refers.
severity	Single-digit code from 0 to 7 that is the severity of the message.
MNEMONIC	Text string that uniquely describes the message.
description	Text string containing detailed information about the event being reported.

For example, sample output on a Cisco switch for an EtherChannel link changing state to up is:

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up

Here the facility is LINK and the severity level is 3, with a MNEMONIC of UPDOWN.

The most common messages are link up and down messages, and messages that a device produces when it exits from configuration mode. If ACL logging is configured, the device generates syslog messages when packets match a parameter condition.

10.2.2.4 Service Timestamp

By default, log messages are not timestamped. For example, in the figure, the R1 GigabitEthernet 0/0 interface is shutdown. The message logged to the console does not identify when the interface state was changed. Log messages should be timestamped so that when they are sent to another destination, such as a Syslog server, there is record of when the message was generated.

```
R1# conf t
R1(config)# interface g0/0
R1(config-if)# shutdown
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0,
changed state to down
R1(config-if)# exit
R1(config)# service timestamps log datetime
R1(config)# interface g0/0
R1(config-if)# no shutdown
*Mar  1 11:52:42: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to down
*Mar  1 11:52:45: %LINK-3-UPDOWN: Interface GigabitEthernet0/0,
changed state to up
*Mar  1 11:52:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/0, changed state to up
R1(config-if)#
```

Use the command **service timestamps log datetime** to force logged events to display the date and time. As shown in the figure, when the R1 GigabitEthernet 0/0 interface is reactivated, the log messages now contain the date and time.

Note: When using the **datetime** keyword, the clock on the networking device must be set, either manually or through NTP, as previously discussed.

10.2.3 Syslog Configuration

10.2.3.1 Syslog Server

To view syslog messages, a syslog server must be installed on a workstation in the network. There are several freeware and shareware versions of syslog, as well as enterprise versions for purchase.

The syslog server provides a relatively user-friendly interface for viewing syslog output. The server parses the output and places the messages into pre-defined columns for easy interpretation. If timestamps are configured on the networking device sourcing the syslog messages, then the date and time of each message displays in the syslog server output.

Network administrators can easily navigate the large amount of data compiled on a syslog server. One advantage of viewing syslog messages on a syslog server is the ability to perform granular searches through the data. Also, a network administrator can quickly delete unimportant syslog messages from the database.

10.2.3.2 Default Logging

By default, Cisco routers and switches send log messages for all severity levels to the console. On some IOS versions, the device also buffers log messages by default. To enable these two settings, use the **logging console** and **logging buffered** global configuration commands, respectively.

The **show logging** command displays the default logging service settings on a Cisco router. The first lines of output list information about the logging process, with the end of the output listing log messages.

10.2.3.3 Router and Switch Commands for Syslog Clients

There are three steps to configuring the router to send system messages to a syslog server where they can be stored, filtered, and analyzed:

Step 1. In global configuration mode, use the **logging** command toconfigure the destination hostname or IPv4 address of the syslog.

Step 2. Control the messages that will be sent to the syslog server with the **logging trap** *level* global configuration mode command. For example, to limit the messages to levels 4 and lower (0 to 4), use one of the two equivalent commands.

Step 3. Optionally, configure the source interface with the **logging source-interface** *interface-type interface-number* global configuration mode command. This specifies that syslog packets contain the IPv4 or IPv6 address of a specific interface, regardless of which interface the packet uses to exit the router.

10.2.3.4 Verifying Syslog

You can use the **show logging** command to view any messages that are logged. When the logging buffer is large, it is helpful to use the pipe option (|) with the **show logging** command. The pipe option allows the administrator to specifically state which messages should be displayed. For example, you can use the pipe to filter only messages that **include changed state to up**.

10.3 Device Maintenance

10.3.1 Router and Switch File Management

10.3.1.1 Router File Systems

The Cisco IOS File System (IFS) allows the administrator to navigate to different directories and list the files in a directory, and to create subdirectories in flash memory or on a disk. The directories available depend on the device.

The **show file systems** command lists all of the available file systems on a Cisco 1941 router. This command provides useful information such as the amount of available and free memory, the type of file system, and its permissions. Permissions include read only (ro), write only (wo), and read and write (rw), shown in the Flags column of the command output.

Although there are several file systems listed, of interest to us will be the tftp, flash, and nvram file systems.

Notice that the flash file system also has an asterisk preceding it. This indicates that flash is the current default file system. The bootable IOS is located in flash; therefore, the pound symbol (#) is appended to the flash listing, indicating that it is a bootable disk.

The Flash File System

The **dir** command lists the contents of flash. Several files are located in flash, but of specific interest is the last listing. This is the name of the current Cisco IOS file image that is running in RAM.

The NVRAM File System

To view the contents of NVRAM, you must change the current default file system using the **cd** (change directory) command. The **pwd** (present working directory) command verifies that we are viewing the NVRAM directory. Finally, the **dir** command lists the contents of NVRAM. Although there are several configuration files listed, of specific interest is the startup-configuration file.

10.3.1.2 Switch File Systems

With the Cisco 2960 switch flash file system, you can copy configuration files, and archive (upload and download) software images.

The command to view the file systems on a Catalyst switch is the same as on a Cisco router: **show file systems**.

10.3.1.3 Backing Up and Restoring Using Text Files

Backup Configurations with Text Capture (Tera Term)

Configuration files can be saved/archived to a text file using Tera Term.

The steps are:

Step 1. On the File menu, click Log.

Step 2. Choose the location to save the file. Tera Term will begin capturing text.

Step 3. After capture has been started, execute the show running-config or show startup-config command at the privileged EXEC prompt. Text displayed in the terminal window will be directed to the chosen file.

Step 4. When the capture is complete, select Close in the Tera Term: Log window.

Step 5. View the file to verify that it was not corrupted.

Restoring Text Configurations

A configuration can be copied from a file to a device. When copied from a text file and pasted into a terminal window, the IOS executes each line of the configuration text as a command. This means that the file will require editing to ensure that encrypted passwords are in plain text and that non-command text such as "--More--" and IOS messages are removed. This process is discussed in the lab.

Further, at the CLI, the device must be set at the global configuration mode to receive the commands from the text file being pasted into the terminal window.

When using Tera Term, the steps are:

Step 1. On the File menu, click Send file.

Step 2. Locate the file to be copied into the device and click Open.

Step 3. Tera Term will paste the file into the device.

The text in the file will be applied as commands in the CLI and become the running configuration on the device. This is a convenient method for manually configuring a router.

10.3.1.4 Backing up and Restoring TFTP

Backup Configurations with TFTP

Copies of configuration files should be stored as backup files in the event of a problem. Configuration files can be stored on a Trivial File Transfer Protocol (TFTP) server or a USB drive. A configuration file should also be included in the network documentation.

To save the running configuration or the startup configuration to a TFTP server, use either the copy running-config tftp or copy startup-config tftp command. Follow these steps to backup the running configuration to a TFTP server:

Step 1. Enter the copy running-config tftp command.

Step 2. Enter the IP address of the host where the configuration file will be stored.

Step 3. Enter the name to assign to the configuration file.

Step 4. Press Enter to confirm each choice.

Restoring Configurations with TFTP

To restore the running configuration or the startup configuration from a TFTP server, use either the copy tftp running-config or copy tftp startup-config command. Use these steps to restore the running configuration from a TFTP server:

Step 1. Enter the copy tftp running-config command.

Step 2. Enter the IP address of the host where the configuration file is stored.

Step 3. Enter the name to assign to the configuration file.

Step 4. Press Enter to confirm each choice.

10.3.1.5 Using USB Ports on a Cisco Router

The Universal Serial Bus (USB) storage feature enables certain models of Cisco routers to support USB flash drives. The USB flash feature provides an optional secondary storage capability and an additional boot device. Images, configurations, and other files can be copied to or from the Cisco USB flash memory with the same reliability as storing and retrieving files using the Compact Flash card. In addition, modular integrated services routers can boot any Cisco IOS Software image saved on USB flash memory. Ideally, USB flash can hold multiple copies of the Cisco IOS and multiple router configurations.

Use the **dir** command to view the contents of the USB flash drive.

10.3.1.6 Backing Up and Restoring Using a USB

Backup Configurations with a USB Flash Drive

When backing up to a USB port, it is a good idea to issue the show file systems command to verify that the USB drive is there and confirm the name.

Next, use the copy run usbflash0:/ command to copy the configuration file to the USB flash drive. Be sure to use the name of the flash drive, as indicated in the file system. The slash is optional but indicates the root directory of the USB flash drive.

The IOS will prompt for the filename. If the file already exists on the USB flash drive, the router will prompt to overwrite.

Use the dir command to see the file on the USB drive and use the more command to see the contents.

Restore Configurations with a USB Flash Drive

In order to copy the file back, it will be necessary to edit the USB R1-Config file with a text editor. Assuming the file name is R1-Config, use the command copy usbflash0:/R1-Config *running-config* to restore a running configuration.

10.3.1.7 Password Recovery

Passwords on devices are used to prevent unauthorized access. For encrypted passwords, such as the enable secret passwords, the passwords must be replaced after recovery. Depending on the device, the detailed procedure for password recovery varies; however, all the password recovery procedures follow the same principle:

Step 1. Enter the ROMMON mode.

Step 2. Change the configuration register to 0x2142 to ignore the startup config file.

Step 3. Make necessary changes to the original startup config file.

Step 4. Save the new configuration.

Console access to the device through a terminal or terminal emulator software on a PC is required for password recovery. The terminal settings to access the device are:

- 9600 baud rate
- No parity
- 8 data bits
- 1 stop bit
- No flow control

With console access, a user can access the ROMMON mode by using a break sequence during the boot up process or removing the external flash memory when the device is powered off.

After the device has finished reloading, copy the startup config to the running config.

CAUTION: Do *not* enter **copy running-config startup-config**. This command erases your original startup configuration.

Because you are in privileged EXEC mode, you can now configure all the necessary passwords. After the new passwords are configured, change the configuration register back to 0x2102 using the **config-register 0x2102** command in the global configuration mode. Save the running-config to startup-config and reload the device.

Note: The password **cisco** is not a strong password and is used here only as an example.

The device now uses the newly configured passwords for authentication. Be sure to use **show** commands to verify that all the configurations are still in place. For example, verify that the appropriate interfaces are not shut down after password recovery.

10.3.2 IOS System Files

10.3.2.1 IOS 15 System Image Packaging

Cisco Integrated Services Routers Generation Two (ISR G2) 1900, 2900, and 3900 Series support services on demand through the use of software licensing. The Services on Demand process enables customers to realize operational savings through ease of software ordering and management. When an order is placed for a new ISR G2 platform, the router is shipped with a single universal Cisco IOS Software image and a license is used to enable the specific feature set packages.

There are two types of universal images supported in ISR G2:

- **Universal images with the "universalk9" designation in the image name** - This universal image offers all of the Cisco IOS Software features, including strong payload cryptography features, such as IPsec VPN, SSL VPN, and Secure Unified Communications.
- **Universal images with the "universalk9_npe" designation in the image name** - The strong enforcement of encryption capabilities provided by Cisco Software Activation satisfies

requirements for the export of encryption capabilities. However, some countries have import requirements that require that the platform does not support any strong cryptography functionality, such as payload cryptography. To satisfy the import requirements of those countries, the npe universal image does not support any strong payload encryption.

10.3.2.2 IOS Image Filenames

When selecting or upgrading a Cisco IOS router, it is important to choose the proper IOS image with the correct feature set and version. The Cisco IOS image file is based on a special naming convention. The name for the Cisco IOS image file contains multiple parts, each with a specific meaning. It is important to understand this naming convention when upgrading and selecting a Cisco IOS Software.

The **show flash** command displays the files stored in flash memory, including the system image files.

Different parts of an IOS 15 system image file on an ISR G2 device:

- **Image Name (c1900)** - Identifies the platform on which the image runs. In this example, the platform is a Cisco 1900 router.
- **universalk9** - Specifies the image designation. The two designations for an ISR G2 are universalk9 and universalk9_npe. Universalk9_npe does not contain strong encryption and is meant for countries with encryption restrictions. Features are controlled by licensing and can be divided into four technology packages. These are IP Base, Security, Unified Communications, and Data.
- **mz** - Indicates where the image runs and if the file is compressed. In this example, mz indicates that the file runs from RAM and is compressed.
- **SPA** - Designates that file is digitally signed by Cisco.
- **152-4.M3** - Specifies the filename format for the image 15.2(4)M3. This is the version of IOS, which includes the major release, minor release, maintenance release, and maintenance rebuild numbers. The M indicates this is an extended maintenance release.
- **bin** - The file extension. This extension indicates that this file is a binary executable file.

The most common designation for memory location and compression format is mz. The first letter indicates the location where the image is executed on the router. The locations can include:

- **f** - flash
- **m** - RAM
- **r** - ROM
- **l** - relocatable

The compression format can be either z for zip or x for mzip. Zipping is a method Cisco uses to compress some run-from-RAM images that is effective in reducing the size of the image. It is self-unzipping, so when the image is loaded into RAM for execution, the first action is to unzip.

Note: The Cisco IOS Software naming conventions, field meaning, image content, and other details are subject to change.

Memory Requirements

On most Cisco routers including the integrated services routers, the IOS is stored in compact flash as a compressed image and loaded into DRAM during boot-up. The Cisco IOS Software Release 15.0 images available for the Cisco 1900 and 2900 ISR require 256MB of flash and 512MB of RAM. The 3900 ISR requires 256MB of flash and 1GB of RAM. This does not include additional management tools such as Cisco Configuration Professional (Cisco CP). For complete details, refer to the product data sheet for the specific router.

10.3.3 IOS Image Management

10.3.3.1 TFTP Servers as a Backup Location

As a network grows, Cisco IOS Software images and configuration files can be stored on a central TFTP server. This helps to control the number of IOS images and the revisions to those IOS images, as well as the configuration files that must be maintained.

Production internetworks usually span wide areas and contain multiple routers. For any network, it is good practice to keep a backup copy of the Cisco IOS Software image in case the system image in the router becomes corrupted or accidentally erased.

Widely distributed routers need a source or backup location for Cisco IOS Software images. Using a network TFTP server allows image and configuration uploads and downloads over the network. The network TFTP server can be another router, a workstation, or a host system.

10.3.3.2 Steps to Backup IOS Image to TFTP Server

To maintain network operations with minimum down time, it is necessary to have procedures in place for backing up Cisco IOS images. This allows the network administrator to quickly copy an image back to a router in case of a corrupted or erased image.

To create a backup of the Cisco IOS image to a TFTP server, perform the following three steps:

Step 1. Ensure that there is access to the network TFTP server. Ping the TFTP server to test connectivity.

Step 2. Verify that the TFTP server has sufficient disk space to accommodate the Cisco IOS Software image. Use the **show flash0:** command on the router to determine the size of the Cisco IOS image file. The file in the example is 68831808 bytes long.

Step 3. Copy the image to the TFTP server using the **copy** *source-url destination-url* command.

After issuing the command using the specified source and destination URLs, the user is prompted for the source file name, IP address of the remote host, and destination file name. The transfer will then begin.

10.3.3.3 Steps to Copy an IOS Image to a Device

Cisco consistently releases new Cisco IOS software versions to resolve caveats and provide new features. This example uses IPv6 for the transfer to show that TFTP can also be used across IPv6 networks.

Follow these steps to upgrade the software on the Cisco router:

Step 1. Select a Cisco IOS image file that meets the requirements in terms of platform, features, and software. Download the file from cisco.com and transfer it to the TFTP server.

Step 2. Verify connectivity to the TFTP server. Ping the TFTP server from the router.

Step 3. Ensure that there is sufficient flash space on the router that is being upgraded. The amount of free flash can be verified using the **show flash0:** command. Compare the free flash space with the new image file size.

Step 4. Copy the IOS image file from the TFTP server to the router using the **copy** command. After issuing this command with specified source and destination URLs, the user will be prompted for IP address of the remote host, source file name, and destination file name. The transfer of the file will begin.

10.3.3.4 The boot system Command

To upgrade to the copied IOS image after that image is saved on the router's flash memory, configure the router to load the new image during bootup using the **boot system** command. Save the configuration. Reload the router to boot the router with new image. After the router has booted, to verify the new image has loaded, use the **show version** command.

During startup, the bootstrap code parses the startup configuration file in NVRAM for the **boot system** commands that specify the name and location of the Cisco IOS Software image to load. Several **boot system** commands can be entered in sequence to provide a fault-tolerant boot plan.

If there are no **boot system** commands in the configuration, the router defaults to loading the first valid Cisco IOS image in flash memory and running it.

10.3.4 Software Licensing

10.3.4.1 Licensing Overview

Beginning with Cisco IOS Software release 15.0, Cisco modified the process to enable new technologies within the IOS feature sets. Cisco IOS Software release 15.0 incorporates cross-platform feature sets to simplify the image selection process. It does this by providing similar functions across platform boundaries. Each device ships with the same universal image. Technology packages are enabled in the universal image via Cisco Software Activation licensing keys. The Cisco IOS Software Activation feature allows the user to enable licensed features and register licenses. The Cisco IOS Software Activation feature is a collection of processes and components used to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.

The technology packages that are available:

- IP Base
- Data
- Unified Communications (UC)
- Security (SEC)

Note: The IP Base license is a prerequisite for installing the Data, Security, and Unified Communications licenses. For earlier router platforms that can support Cisco IOS Software release 15.0, a universal image is not available. It is necessary to download a separate image that contains the desired features.

Technology Package Licenses

Technology package licenses are supported on Cisco ISR G2 platforms (Cisco 1900, 2900, and 3900 Series routers). The Cisco IOS universal image contains all packages and features in one image. Each package is a grouping of technology-specific features. Multiple technology package licenses can be activated on the Cisco 1900, 2900, and 3900 series ISR platforms.

Note: Use the **show license feature** command to view the technology package licenses and feature licenses supported on the router.

10.3.4.2 Licensing Process

When a new router is shipped, it comes preinstalled with the software image and the corresponding permanent licenses for the customer-specified packages and features.

The router also comes with the evaluation license, known as a temporary license, for most packages and features supported on the specified router. This allows customers to try a new software package or feature by activating a specific evaluation license. If customers want to permanently activate a software package or feature on the router, they must get a new software license.

10.3.4.3 Step 1. Purchase the Software Package or Feature to Install

The first step is to purchase the software package or feature needed. This may be adding a package to IP Base, such as Security.

Software Claim Certificates are used for licenses that require software activation. The claim certificate provides the Product Activation Key (PAK) for the license and important information regarding the Cisco End User License Agreement (EULA). In most instances, Cisco or the Cisco channel partner will have

already activated the licenses ordered at the time of purchase and no Software Claim Certificate is provided.

In either instance, customers receive a PAK with their purchase. The PAK serves as a receipt and is used to obtain a license. A PAK is an 11 digit alpha numeric key created by Cisco manufacturing. It defines the Feature Set associated with the PAK. A PAK is not tied to a specific device until the license is created. A PAK can be purchased that generates any specified number of licenses.

10.3.4.4 Step 2. Obtain a License

The second step is to obtain the license, which is actually a license file. A license file, also known as a Software Activation License, is obtained using one of the following options:

- **Cisco License Manager (CLM)** - This is a free software application available at http://www.cisco.com/go/clm. Cisco License Manager is a standalone application from Cisco that helps network administrators rapidly deploy multiple Cisco software licenses across their networks. Cisco License Manager can discover network devices, view their license information, and acquire and deploy licenses from Cisco. The application provides a GUI that simplifies installation and helps automate license acquisition, as well as perform multiple licensing tasks from a central location. CLM is free of charge and can be downloaded from CCO.
- **Cisco License Registration Portal** - This is the web-based portal for getting and registering individual software licenses, available at http://www.cisco.com/go/license.

Both of these processes require a PAK number and a Unique Device Identifier (UDI).

The PAK is received during purchase.

The UDI is a combination of the Product ID (PID), the Serial Number (SN), and the hardware version. The SN is an 11 digit number which uniquely identifies a device. The PID identifies the type of device. Only the PID and SN are used for license creation. This UDI can be displayed using the **show license udi** command. This information is also available on a pull-out label tray found on the device.

After entering the appropriate information, the customer receives an email containing the license information to install the license file. The license file is an XML text file with a .lic extension.

10.3.4.5 Step 3. Install the License

After the license has been purchased, the customer receives a license file. Installing a permanent license requires two steps:

Step 1. Use the **license install** *stored-location-url* privileged exec mode command to install a license file.

Step 2. Reload the router using the privileged exec command **reload**. A reload is not required if an evaluation license is active.

Note: Unified Communications is not supported on 1941 routers.

A permanent license is a license that never expires. After a permanent license is installed on a router, it is good for that particular feature set for the life of the router, even across IOS versions. For example, when a UC, SEC, or Data license is installed on a router, the subsequent features for that license are activated even if the router is upgraded to a new IOS release. A permanent license is the most common license type used when a feature set is purchased for a device.

Note: Cisco manufacturing preinstalls the appropriate permanent license on the ordered device for the purchased feature set. No customer interaction with the Cisco IOS Software Activation processes is required to enable that license on new hardware.

10.3.5 License Verification and Management

10.3.5.1 License Verification

After a new license has been installed the router must be rebooted using the **reload** command. The **show version** command is used after the router is reloaded to verify that license has been installed.

The **show license** command is used to display additional information about Cisco IOS software licenses. This command displays license information used to help with troubleshooting issues related to Cisco IOS software licenses. This command displays all the licenses installed in the system. In this example, both the IP Base and Security licenses have been installed. This command also displays the features that are available, but not licensed to execute, such as the Data feature set. Output is grouped according to how the features are stored in license storage.

The following is a brief description of the output:

- **Feature** - Name of the feature
- **License Type** - Type of license; such as Permanent or Evaluation
- **License State** - Status of the license; such as Active or In Use
- **License Count** - Number of licenses available and in use, if counted. If non-counted is indicated, the license is unrestricted.
- **License Priority** - Priority of the license; such as high or low

10.3.5.2 Activate an Evaluation Right-To-Use License

Evaluation licenses are replaced with Evaluation Right-To-Use licenses (RTU) after 60 days. An Evaluation license is good for a 60 day evaluation period. After the 60 days, this license automatically transitions into an RTU license. These licenses are available on the honor system and require the customer's acceptance of the EULA. The EULA is automatically applied to all Cisco IOS software licenses.

The **license accept end user agreement** global configuration mode command is used to configure a one-time acceptance of the EULA for all Cisco IOS software packages and features. After the command is issued and the EULA accepted, the EULA is automatically applied to all Cisco IOS software licenses and the user is not prompted to accept the EULA during license installation.

Here below shows how to configure a one-time acceptance of the EULA:

Router(config)# **license accept end user agreement**

In addition, here below shows the command to activate an Evaluation RTU license:

Router# **license boot module** *module-name* **technology-package** *package-name*

Use the **?** in place of the arguments to determine which module names and supported software packages are available on the router. Technology package names for Cisco ISR G2 platforms are:

- **ipbasek9** - IP Base technology package
- **securityk9** - Security technology package
- **datak9** - Data technology package
- **uck9** - Unified Communications package (not available on 1900 series)

Note: A reload using the **reload** command is required to activate the software package.

Evaluation licenses are temporary, and are used to evaluate a feature set on new hardware. Temporary licenses are limited to a specific usage period (for example, 60 days).

Reload the router after a license is successfully installed using the **reload** command. The **show license** command verifies that the license has been installed.

10.3.5.3 Back up the License

The **license save** command is used to copy all licenses in a device and store them in a format required by the specified storage location. Saved licenses are restored by using the **license install** command.

The command to back up a copy of the licenses on a device is:

Router# **license save** *file-sys://lic-location*

Use the **show flash0:** command to verify that the licenses have been saved.

The license storage location can be a directory or a URL that points to a file system. Use the **?** command to see the storage locations supported by a device.

10.3.5.4 Uninstall the License

To clear an active permanent license from the Cisco 1900 series, 2900 series, and 3900 series routers, perform the following steps:

Step 1. Disable the technology package.

- Disable the active license with the command:
- Router(config)# **license boot module** *module-name* **technology-package** *package-name* **disable**
- Reload the router using the **reload** command. A reload is required to make the software package inactive.

Step 2. Clear the license.

- Clear the technology package license from license storage.

Router# **license clear** *feature-name*

- Clear the **license boot module** command used for disabling the active license:

Router(config)# no license boot module *module-name* technology-package *package-name* disable

Note: Some licenses, such as built-in licenses, cannot be cleared. Only licenses that have been added by using the **license install** command are removed. Evaluation licenses are not removed.

Figure below shows an example of clearing an active license.

Step 1. Disable the technology package.

```
R1(config)# license boot module c1900 technology-package
seck9 disable
R1(config)# exit
R1# reload
```

Step 2. Clear the license.

```
R1# license clear seck9
R1# configure terminal
R1(config)# no license boot module c1900 technology-package
seck9 disable
R1(config)# exit
R1# reload
```

REFERENCES