Module 1: Building a Simple Network

This module provides a high-level overview of basic networking components and their functions. The need for a communication module is explained, followed by an overview of the TCP/IP protocol stack. Cisco IOS Software is introduced, and its basic functions and features are described. Basic switch configuration is described, with configuration examples so that learners can perform switch startup and initial configuration in the associated lab. LANs are introduced, as well as the Ethernet standard. The operation and role of switches within LANs is described. Finally, the module provides an overview of common switch media issues and lists recommended troubleshooting steps.

Lesson 1: Exploring the Functions of Networking

Introduction

You have succeeded in getting a job interview with a company named CCS. It is an IT services firm that specializes in providing managed IT and software services to law firms, among other companies. CCS provides networking design, implementation, and support services. You are interviewing for an entry-level network engineer position on the implementation team.

The hiring manager, Maya, explains the interview process with you. The first phase is a two-part phone interview. The first part focuses on your work history, and the second part is more technical. She wants to conduct the first part of the interview right away.

You can either schedule the second part later or move forward with it immediately. She suggests that before you decide, she can provide you with more information about the technical portion of the interview. You understand that if you do well, you will be scheduled for a more in-depth technical interview, so you want to make sure that you are fully prepared. She explains what the interview will require you to do:

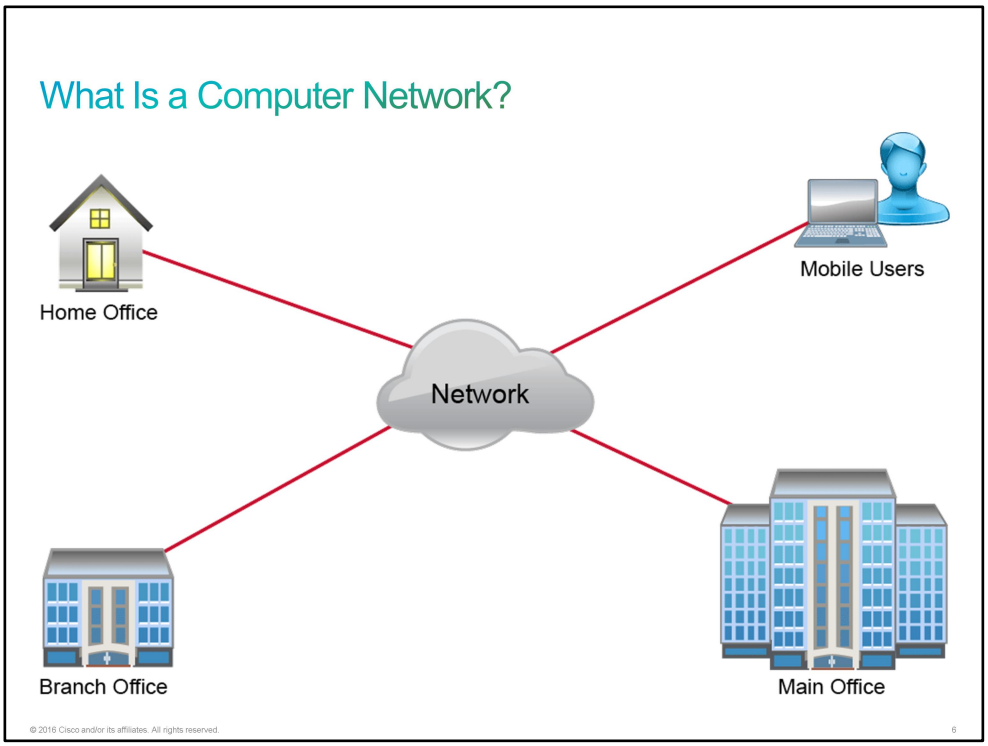
- Explain the functions, characteristics, and common components of a network
- Read a network diagram, including comparing and contrasting the logical and physical topologies
- Describe the impact of user applications on the network

Maya asks if you would like to continue the interview now or take some time to prepare.

What Is a Computer Network?

The term *network* is used in many different arenas. There are social networks, phone networks, television networks, and, of course, computer networks. In all cases, a network is a means of connecting various components together. A computer network connects PCs, printers, servers, phones, cameras, and other types of devices so that they can communicate with each other.

Networks carry data in many types of environments, including homes, small businesses, and large enterprises. Large enterprise networks may have a number of locations that need to communicate with each other. Based on where workers are situated, these locations are as follows:

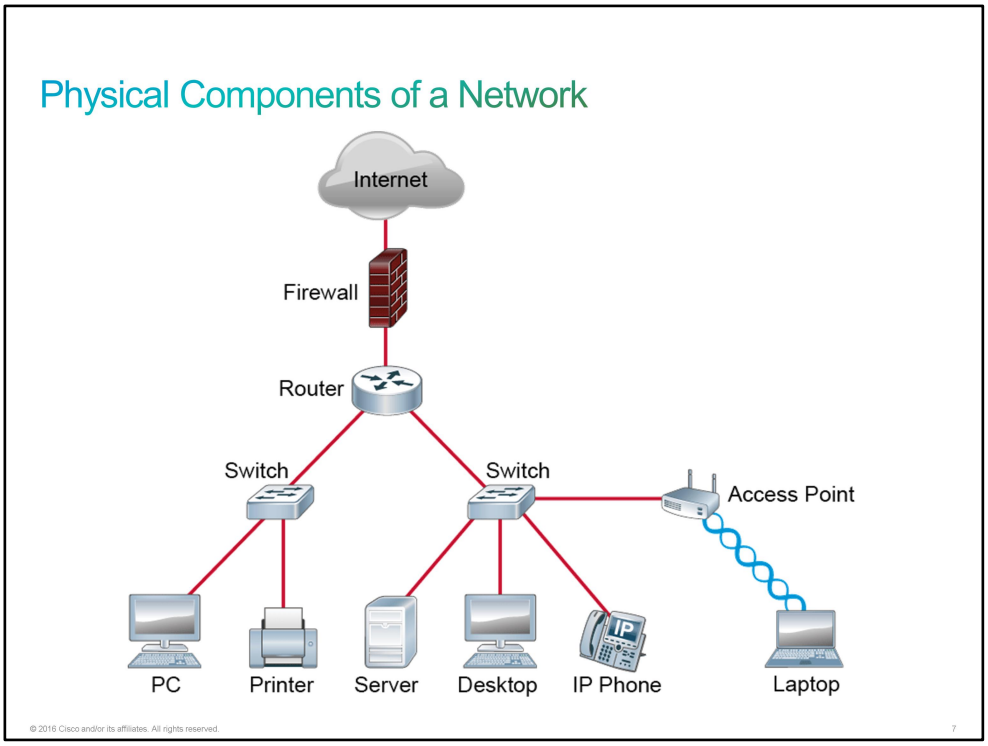


- **Main office:** A main office is a site where everyone is connected via a network and where most corporate information is located. A main office can have hundreds or even thousands of people who depend on network access to do their jobs. A main office may use several connected networks that can span many floors in an office building or cover a campus that contains several buildings.
- **Remote locations:** Various remote access locations use networks to connect to the main office or to each other.
  - **Branch offices:** In branch offices, smaller groups of people work and communicate with each other via a network. Although some corporate information may be stored at a branch office, it is more likely that branch offices have local network resources, such as printers, but must access information directly from the main office.
  - **Home offices:** When individuals work from home, their location is called a home office. Home-office workers often require on-demand connections to the main office or branch offices to access information or to use network resources such as file servers.
  - **Mobile users:** Mobile users connect to the main office network while at the main office, at the branch office, or while traveling. The location of mobile users determines their network access requirements.

You can use a network in your home office to communicate via the Internet in order to locate information, place orders for merchandise, and send messages to friends. You can also have a small office that is set up with a network that connects other computers and printers in the office. Similarly, you can work in a large enterprise with many computers, printers, storage devices, and servers that are used to communicate and store information from many departments over large geographic areas.

Physical Components of a Network

Man different types of devices can be part of a network. A network can be as simple as two PCs that are connected by a wire or as complex as several thousands of devices that are connected through many different types of media.



Take a look at the five major components that you may find in a network:

- **Endpoints:** These elements include devices such as PCs, file servers, printers, tablets, sensors, cameras, and manufacturing robots.
- **Interconnections:** These are the components that connect the devices on the network. They provide a means for data to travel from one point to another in the network. This category includes the following components:
  - NICs translate computer data into a format that can be transmitted over the network. A NIC is the device on a PC into which you plug a network cable.
  - Network media, such as cables or wireless media, provide the means by which signals are transmitted from one network device to another.
  - Connectors provide connection points for the media. The most common type of connector is the plug on the end of a network cable that looks like an analog phone connector. This is called an RJ-45 connector. Compared to an analog phone connector the RJ-45 connector is slightly larger and contains eight wires instead of four.
- **Switches:** These are devices that endpoints such as PCs, file servers, printers, sensors, cameras, and manufacturing robots typically connect to. Usually, all the computers that are connected to the same switch can communicate directly with one another. They share what can be called a *common network*. If a computer wants to communicate with a device that is on a separate network, then it needs a device that is known as a *router*, which connects the two networks together.

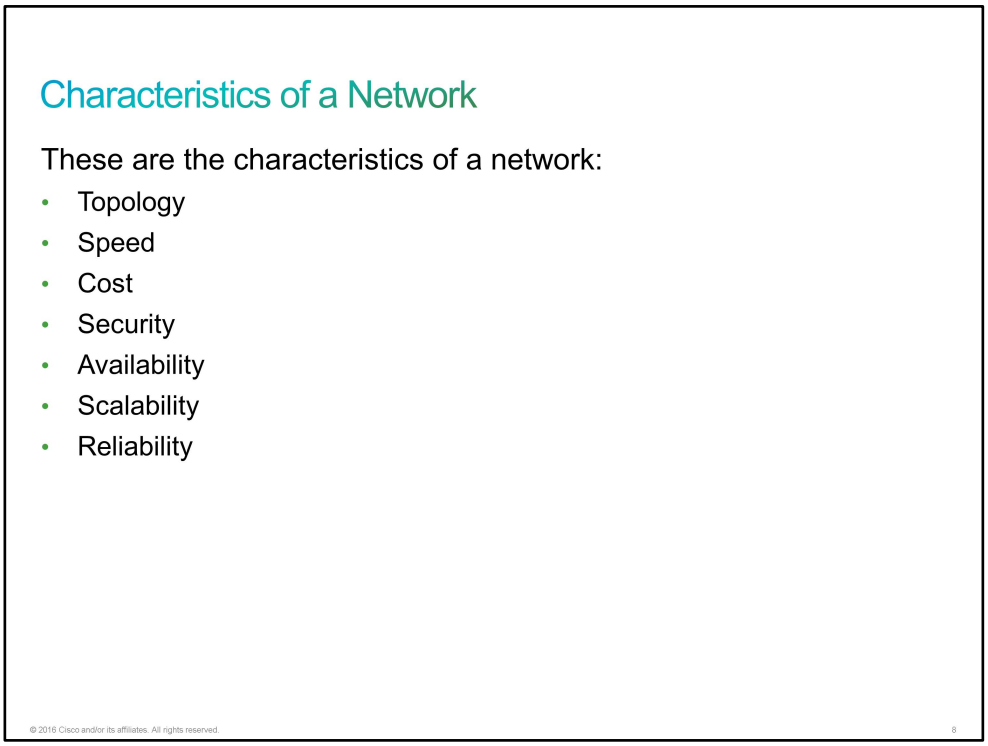
Note	Before switches became affordable, many networks used devices that were called hubs. Hubs serve functions similar to switches, but they have a number of limitations. These limitations include lower speed and poor performance.

- **Routers:** These devices connect networks and intelligently choose the best paths between networks. Their main function is to route traffic from one network to another. For example, you need a router to connect your office network to the Internet. An analogy that may help you understand the function of switches and routers is that of a neighborhood. Think of the devices that are plugged into a switch as the houses on a city block. From a house on that block, you can go to any other house on the block without having to cross a street. However, if you want to travel to a house that is on another block, you must cross a street intersection. In networking, when you want to connect to a device on a different network from your own, you need to cross a router. Just like an intersection connects blocks in a neighborhood, a router connects computer networks.
- **WLAN devices:** These devices connect wireless devices such as computers, printers, and tablets to the network. The minimum requirement for wireless access to the network is a device with a WLAN NIC and a wireless AP that is connected to a traditional wired network.
- **Access Points or APs**: These devices allow wireless devices to connect to a wired network. An AP usually connects to a router as a standalone device, but it can also be an integral component of the router itself.
- **WLAN Controllers**: These are the devices that network administrators or network operations centers use to manage access points in large quantities. The WLAN controller automatically handles the configuration of wireless access points.
- **Firewalls:** These devices are network security systems that monitor and control the incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted, secure internal network and another outside network, such as the Internet, that is assumed not to be secure or trusted.

Homes may have a single device that provides connectivity for wired and wireless devices as well as providing access to the Internet. You may be wondering which kind of device that is. It seems to have characteristics of a switch, a router, and a WLAN AP. The answer is that it is actually all three of those devices in a single package. It acts as a switch by providing physical ports to plug local devices into. It acts as a WLAN AP by allowing wireless devices to connect to it. And it acts as a router by connecting the local network to the Internet.

Characteristics of a Network

When you purchase a PC, one of the things that you examine is the specifications list. It tells you the important characteristics of the PC. As with the specifications list for a PC, the specific characteristics of a network help to describe its performance and structure. Understanding what each of the characteristics of a network means, enables you to better understand how the network is designed and what type of performance you should expect from it.



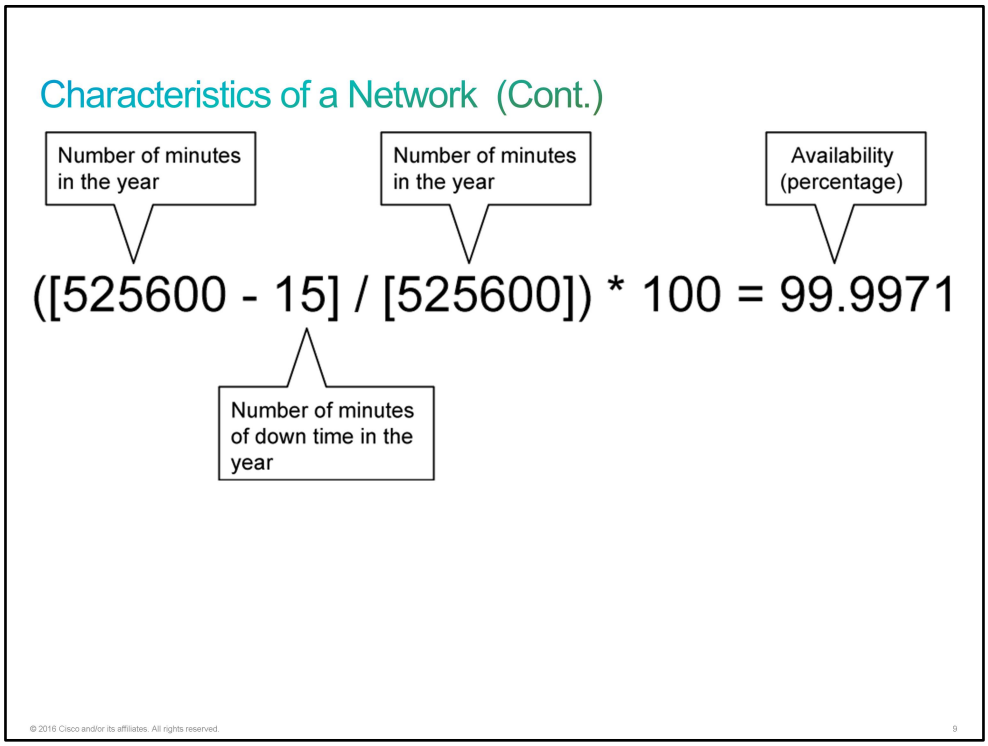
You can describe a network according to performance and structure:

- **Topology:** In networks, there are physical and logical topologies. The physical topology is the arrangement of cables, network devices, and end systems. The logical topology is the path over which the data is transferred in a network. For example, a physical topology describes how the network devices are actually interconnected with wires and cables. A logical topology describes how the network devices appear connected to network users.
- **Speed:** Speed is a measure of the data rate in bits per second of a given link in the network.
- **Cost:** Cost indicates the general expense for the purchasing of network components and the 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. You should consider security whenever you take actions that affect the network.
- **Availability:** Availability is a measure of the probability that the network will be available for use when it is required. For networks that are meant to be used 24 hours per day, 7 days per week, 365 days per year, availability is calculated by dividing the time that it is actually available by the total time in a year and then multiplying by 100 to get a percentage.

For example, if a network is unavailable for 15 minutes per year because of network outages, you can calculate its percentage availability as follows:

([Number of minutes in a year – down time] / [number of minutes in a year]) * 100 = percentage availability

([525600 – 15] / [525600]) * 100 = 99.9971

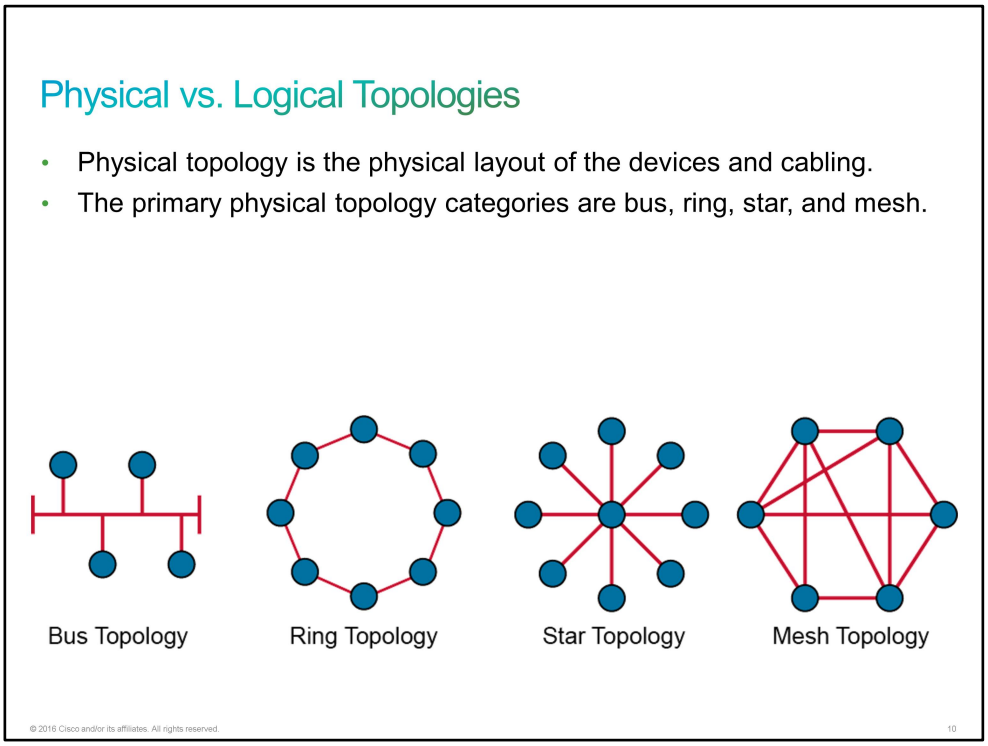


- **Scalability:** Scalability indicates how easily the network can accommodate more users and data transmission requirements. If you design and optimize a network for only the current requirements, it can be very expensive and difficult to meet new needs when the network grows.
- **Reliability:** Reliability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

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

Physical vs. Logical Topologies

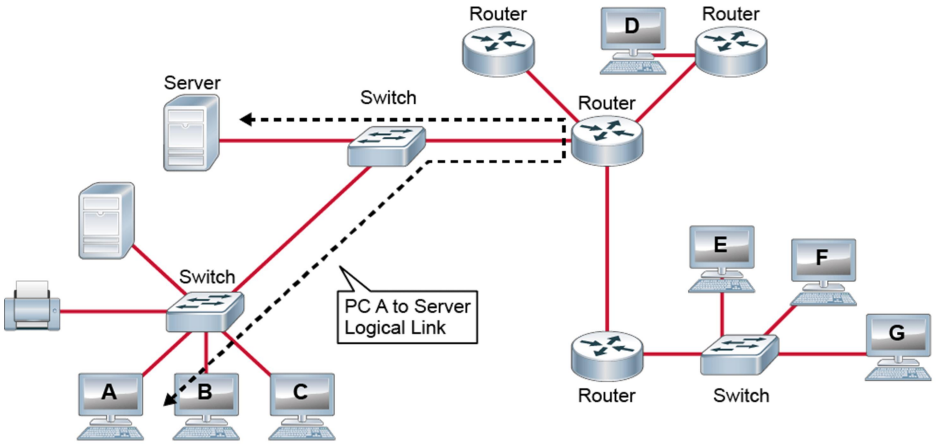
Each type of network has both a physical and a logical topology. The physical topology of a network refers to the physical layout of the devices and cabling. You must match the appropriate physical topology to the type of cabling that you will install, such as twisted pair, coaxial, or fiber. Therefore, understanding the type of cabling that is used is important in understanding each type of physical topology. The logical topology defines the logical path on which data will travel from one point to another. First, take a look at some of the types of physical topologies that you may encounter.



- **Bus:** In a bus topology, every workstation is connected to the main cable. Therefore, each workstation is directly connected to every other workstation in the network. In early bus topologies, computers and other network devices were cabled together in a line using coaxial cable. Modern bus topologies establish the bus in a hardware device and connect the host devices to the bus using twisted-pair wiring.
- **Ring:** In a ring topology, computers and other network devices are cabled together in a way that the last device is connected to the first to form a circle or ring. Each device is connected to exactly two neighbors and has no direct connection to a third. Physical connection can be made using either coaxial or fiber wiring.
- **Star:** The most common physical topology is a star topology. In this topology, a central cabling device connects the computers and other network devices. This category includes star and extended star topologies. Physical connection is commonly made using twisted-pair wiring.
- **Mesh:** In a mesh topology, every network device is cabled together with many others. Redundant links increase reliability and self-healing. The physical connection is commonly made using fiber or twisted-pair wiring.

Physical vs. Logical Topologies (Cont.)

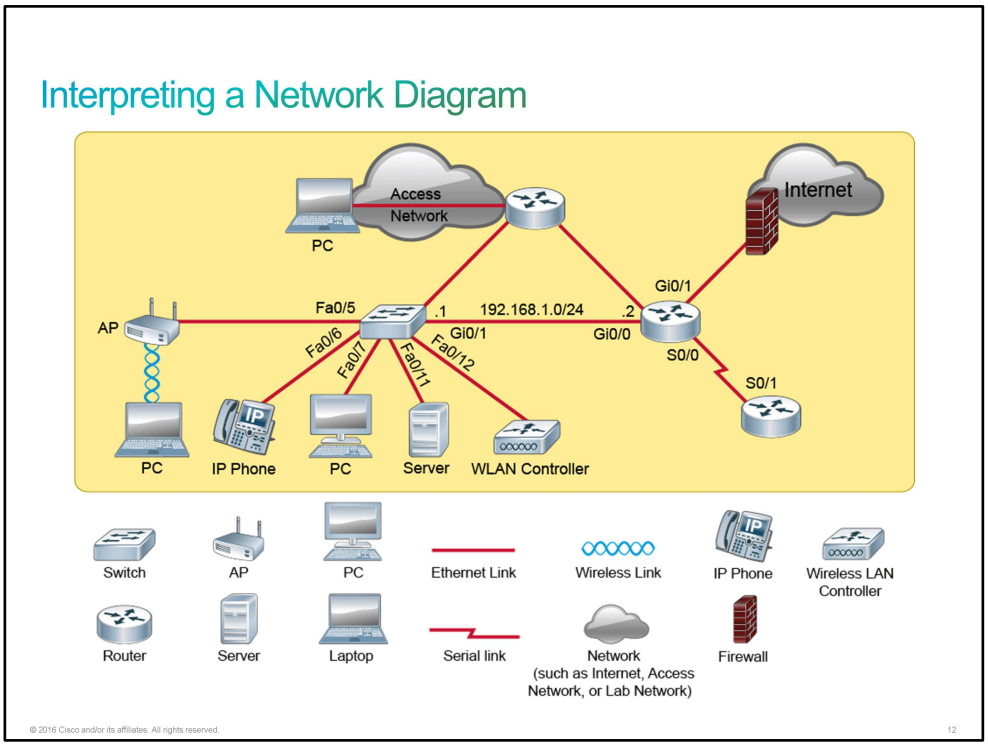
The logical topology is the path along which data travels from one point in the network to another.



It is possible for the logical and physical topology of a network to be of the same type. However, physical and logical topologies often differ. For example, an Ethernet hub is an example of a physical star topology with a logical bus topology. A physical star topology is by far the most common implementation of LANs today. Ethernet uses a logical bus topology in either a physical bus or a physical star topology.

Interpreting a Network Diagram

One of the most important tasks that you must complete when designing a network is to create a network diagram. It is basically a map of the network. It illustrates the logical representation of all devices in the network and clarifies how they are interconnected. In addition, a proper diagram provides information such as the interface IDs and network addressing. The figure shows a network diagram and Cisco icons that are commonly used to represent network devices in network diagrams.



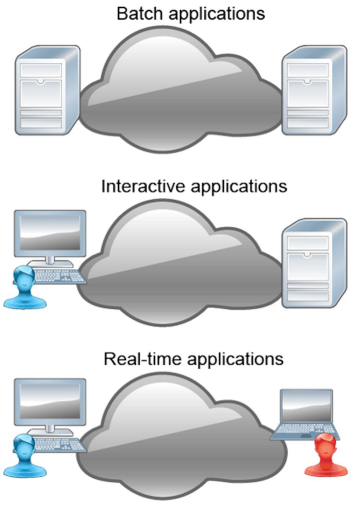
Other information may be included in the network diagram as space allows. For example, it is common to identify the interface on a device in the S0/0/0 format for a serial interface. For Ethernet interfaces, Fa0/0 identifies a Fast Ethernet interface and Gi0/1 identifies a Gigabit Ethernet interface. It is also common to include the network address of the segment in the 192.168.1.0/24 format. In the example that is shown in the figure, 192.168.1.0 indicates the network address, /24 indicates the subnet mask, and .1 and .2 at the device ends indicate IP addresses of the interfaces (.1 corresponds to 192.168.1.1).

Impact of User Applications on the Network

Applications can affect network performance and, conversely, network performance can affect applications. Here you will learn about common interactions between user applications and the network.

Impact of User Applications on the Network

- Batch applications:
  - FTP, TFTP, inventory updates
  - No direct human interaction
  - Bandwidth important, but not critical
- Interactive applications:
  - Inventory inquiry
  - Human-to-machine interaction
  - Human waiting for response, response time important but not critical, unless waiting becomes excessive
- Real-time applications:
  - VoIP, video
  - Human-to-human interaction
  - End-to-end latency critical



Batch Applications

Applications such as FTP and TFTP are considered batch applications. Both are used to send and receive files. Typically, a user selects a group of files that need to be retrieved and then starts the transfer. Once the download starts, no additional human interaction is required. The amount of available bandwidth determines the speed at which the download occurs. While bandwidth is important for batch applications, it is not critical. Even with low bandwidth, the download is completed eventually.

Interactive Applications

Interactive applications are applications in which the user waits for a response—for example, inventory lookup. Because these applications require human interaction, response times are more important than for batch applications, but they are still not critical. The transaction will still be completed if the appropriate amount of bandwidth is not available, but it may take longer. Examples of the interactive applications are used in plant automation, physical security, sport, and retail.

Real-Time Applications

Real-time applications such as voice and video applications also involve human interaction. Because of the amount of information that is transmitted, bandwidth is critical. In addition, because these applications are time-critical, a delay on the network can cause a problem. Timely delivery of the data is crucial. It is also important that data is not lost during transmission, because real-time applications, unlike other applications, do not retransmit lost data. Therefore, sufficient bandwidth is mandatory and the quality of the transmission must be ensured by implementing QoS. QoS is a way of granting higher priority to certain types of data, such as VoIP.

Challenge

1. What are the varieties of remote access locations? (Choose three.)

   A. branch offices
   B. head office
   C. mobile users
   D. main office
   E. home offices
   F. Internet

2. What is a function of the WLAN Controller?

   A. to monitor and control the incoming and outgoing network traffic
   B. to automatically handle the configuration of wireless access point
   C. to allow wireless devices to connect to a wired network
   D. to connect networks and intelligently choose the best paths between networks

3. What is a function of the firewall?

   A. to automatically handle the configuration of wireless access points
   B. to allow wireless devices to connect to a wired network
   C. to connect networks and intelligently choose the best paths between networks
   D. to monitor and control the incoming and outgoing network traffic

4. What is the percentage availability of the network that is not available for 15 minutes each month?

   A. 99.9657%
   B. 0.9996%
   C. 99.8457%
   D. 0.0342%

5. What network characteristic indicates the general expense for the purchasing of network components and installation and maintenance?

   A. speed
   B. security
   C. cost
   D. availability

6. What network characteristic indicates the dependability of the components that make up the network?

   A. reliability
   B. scalability
   C. security
   D. availability

7. In what type of applications does the user need to wait for a response when performing actions such as inventory lookup or a database update?

   A. batch applications
   B. real-time applications
   C. interactive applications
   D. human-to-human applications

Answer Key

Challenge

1. A, C, E
2. B
3. D
4. A
5. C
6. A
7. C

Lesson 2: Understanding the Host-to-Host Communications Model

Introduction

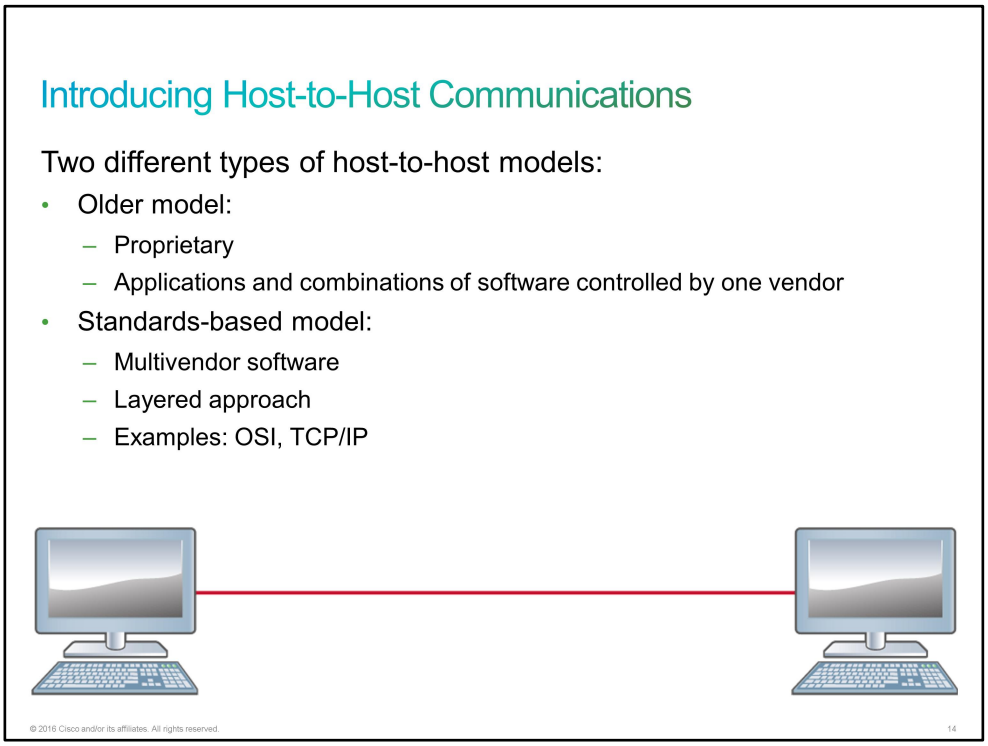
You just received a phone call from Maya. She has reviewed the technical portion of your phone interview with the lead engineer, Bob, and she was very impressed. Bob would like to interview you. Maya tells you that, in addition to the typical interview questions, Bob will also want to test your knowledge of topics that are more technical than the ones that were covered in the phone interview.

Maya says that Bob has time now for the interview. Otherwise, you can set up an appointment for later this week after you have had time to prepare.

You will need to demonstrate that you are familiar with the host-to-host communications model. Are you familiar with the TCP/IP model, its layers, and functions? Can you explain the terms bit, frame, packet, segment, and so on? You should also review the OSI reference model, as it is an alternative to the TCP/IP stack. Lastly, to show your understanding of the operation of the communications model, you should review the encapsulation and de-encapsulation processes.

Introducing Host-to-Host Communications

Host-to-host communications require a consistent model. The model addresses hardware, software, and data transmission.



The network devices that people are most familiar with are called *end devices*. End devices form the interface between the human network and the underlying communications network. In the context of a network, end devices are called *hosts*. A host device is either the source or the destination of a message that is transmitted over the network. Communication begins with a message, or information, that must be sent from one device to another device. The message then flows through the network and arrives at the end device.

Successful communication between hosts on a network requires the interaction of many different protocols. A protocol is a set of rules that govern communications. Networking protocols describe the functions that occur during network communications. Protocols are implemented in the software and hardware of each host and other devices.

Original host-to-host communications models were proprietary. Each vendor controlled its own application and embedded communications software. An application that was written by one vendor would not function on a network that was developed by another vendor. In the computer industry, *proprietary* is the opposite of *open*. Proprietary means that one company or a small group of companies control all use of the technology. Open means that the use of the technology is available and is free to the public.

Business drivers and technology advances led to a multivendor solution. The first step was to separate application software from communications software, which allowed new communications technologies to be implemented without requiring new applications. However, it still requires a single-vendor solution for communications software and hardware.

It became apparent that a multivendor solution for communications software and hardware would require a layered approach with clearly defined rules for interlayer interaction. Within a layered model, various vendors provide solutions for separate layers. Hardware vendors could design hardware and software to support emerging physical-level technologies (that is, Ethernet, Token Ring, Frame Relay, and so on). Other vendors could write software that network operating systems that control host communications would use.

Standards-based layered models provide several benefits:

- Reducing complexity by breaking network communications into smaller, simpler parts
- Standardizing network components to allow different vendors to provide solutions for separate layers
- Facilitating modular engineering, allowing different types of network hardware and software to communicate with one another
- Ensuring interoperable technology and preventing changes in one layer from affecting the other layers
- Accelerating evolution, providing for effective updates and improvements to individual components without affecting other components or having to rewrite the entire protocol
- Simplifying teaching and learning

Examples of such standards-based models are TCP/IP and OSI.

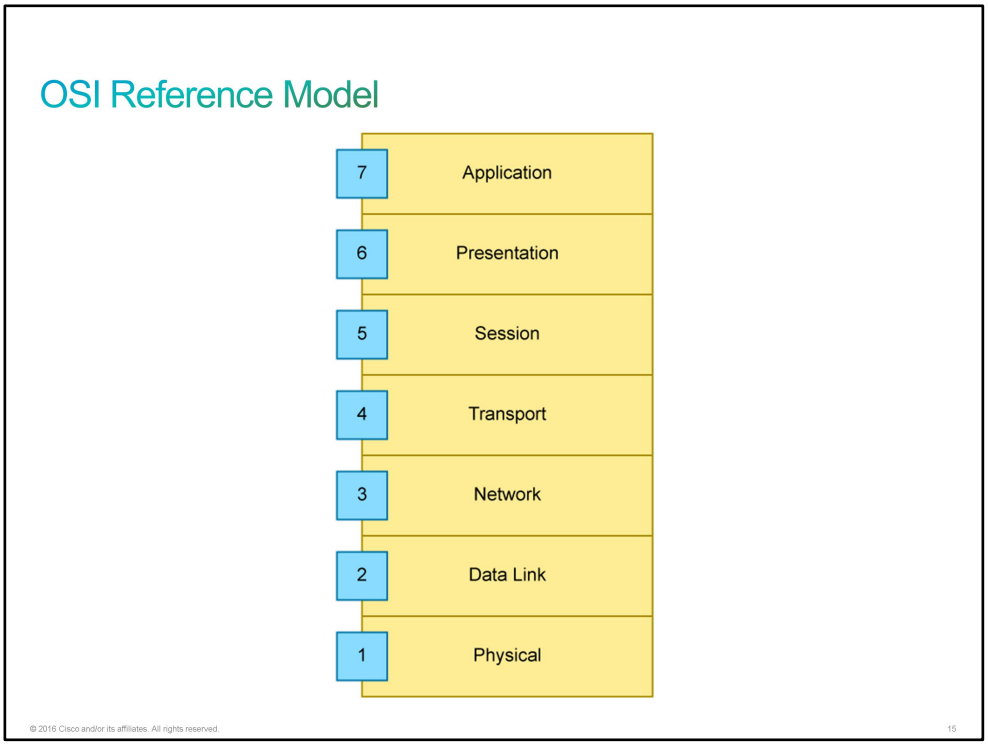
OSI Reference Model

To address the problem of networks being incompatible and unable to communicate with each other, the ISO researched different network schemes. As a result of this research, the ISO created a model to serve as a framework on which to build a suite of open systems protocols. The vision was that this set of protocols would be used to develop an international network that would not depend on proprietary systems.

Note	ISO, the International Organization for Standardization, is an independent, nongovernmental organization. It is the world's largest developer of voluntary international standards. Those standards help businesses to increase productivity while minimizing errors and waste.

The OSIreference model provides a means of describing how data is transmitted over a network. The model addresses hardware, software, and data transmission.

As a reference, the OSI model provides an extensive list of functions and services that can occur at each layer. It also describes the interaction of each layer with the layers directly above and below it. More importantly, the OSI model facilitates an understanding of how information travels throughout the network. It provides vendors with a set of standards that ensure compatibility and interoperability between the various types of network technologies that companies produce around the world. You also use the OSI model for data network design, operation specifications, and troubleshooting.



The OSI reference model separates network functions into seven categories. This separation of networking functions is called layering. The OSI reference model has seven numbered layers, each one illustrating a particular network function.

Physical Layer (Layer 1)

The physical layer defines certain specifications (for example, electrical, mechanical, procedural, and functional). These specifications are needed for activating, maintaining, and deactivating the physical link between end devices. This physical link enables bit transmission between end devices. Physical layer specifications are defining characteristics—for example, voltage levels, timing of voltage changes, physical data rates, maximum transmission distances, physical connectors, and other similar attributes.

Data Link Layer (Layer 2)

The data link layer defines how data is formatted for transmission and how access to physical media is controlled. This layer also typically includes error detection and correction to ensure a reliable delivery of the data.

Network Layer (Layer 3)

The network layer provides connectivity and path selection between two host systems that may be located on geographically separated networks. With the growth of the Internet, the number of users that access information from sites around the world has increased. The network layer is the layer that manages the connectivity of these users by providing logical addressing.

Transport Layer (Layer 4)

The transport layer defines services to segment, transfer, and reassemble the data for individual communications between the end devices. For example, business users in large corporations often transfer large files from field locations to a corporate site. Reliable delivery of the files is important, so the transport layer breaks down large files into smaller segments that are less likely to incur transmission problems. TCP and UDP are the protocols that operate at this layer. TCP is used when data loss cannot be tolerated (file transfer), and UDP is used when some data loss is acceptable (when speed is more important than accuracy, for example in video streaming).

Session Layer (Layer 5)

The session layer establishes, manages, and terminates sessions between two communicating hosts. The session layer also synchronizes dialog between the presentation layers of the two hosts and manages their data exchange. For example, web servers have many users, so there are many communication processes open at a given time. As a result, it is important to keep track of which user communicates on which path. In addition to session regulation, the session layer offers provisions for efficient data transfer, CoS, and exception reporting of session layer, presentation layer, and application layer problems.

Presentation Layer (Layer 6)

The presentation layer ensures that the information that is sent at the application layer of one system is readable by the application layer of another system. For example, when a PC program is communicating with another computer, each computer might be using a different encoding scheme. The presentation layer has to translate among multiple data formats using a common format.

Application Layer (Layer 7)

The application layer is the OSI layer that is closest to the user. This layer provides network services to the applications of the user, such as email, file transfer, and terminal emulation. The application layer differs from the other layers in that it does not provide services to any other OSI layer. It provides services only to applications outside the OSI model. The application layer establishes the availability of intended communication partners. It then synchronizes and establishes agreement on procedures for error recovery and control of data integrity.

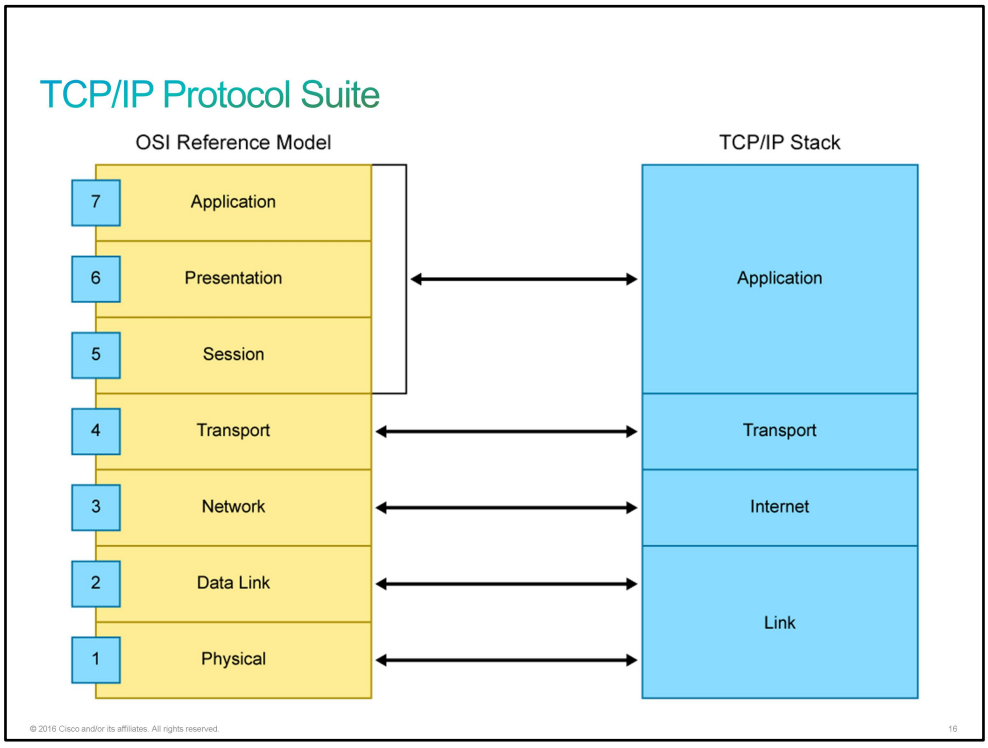
TCP/IP Protocol Suite

TCP/IP stands for Transmission Control Protocol/Internet Protocol. It defines how devices should be connected over the Internet, and how data should be transmitted between those devices.

TCP/IP is actually two protocols, but they are used together so often that many people think of them as a single protocol. TCP operates at Layer 4 and is responsible for making sure that the data that the source device sends arrives at its destination. IP operates at Layer 3 and is responsible for the transmission of data. It does not do any error correction itself.

The TCP/IP reference model is similar to the OSImodel. It also separates data communication into layers. However, it predates the OSI model and consists of only four layers. The TCP/IP model provides a common reference for maintaining consistency within all types of network protocols and services. It is not intended to be an implementation specification or to provide sufficient detail to precisely define the services of the network architecture. The primary purpose of this reference model is to help you understand the functions and processes that are involved in data communication.

Note	Although this course refers to the TCP/IP stack, it has become common in the industry to shorten this term to "IP stack."



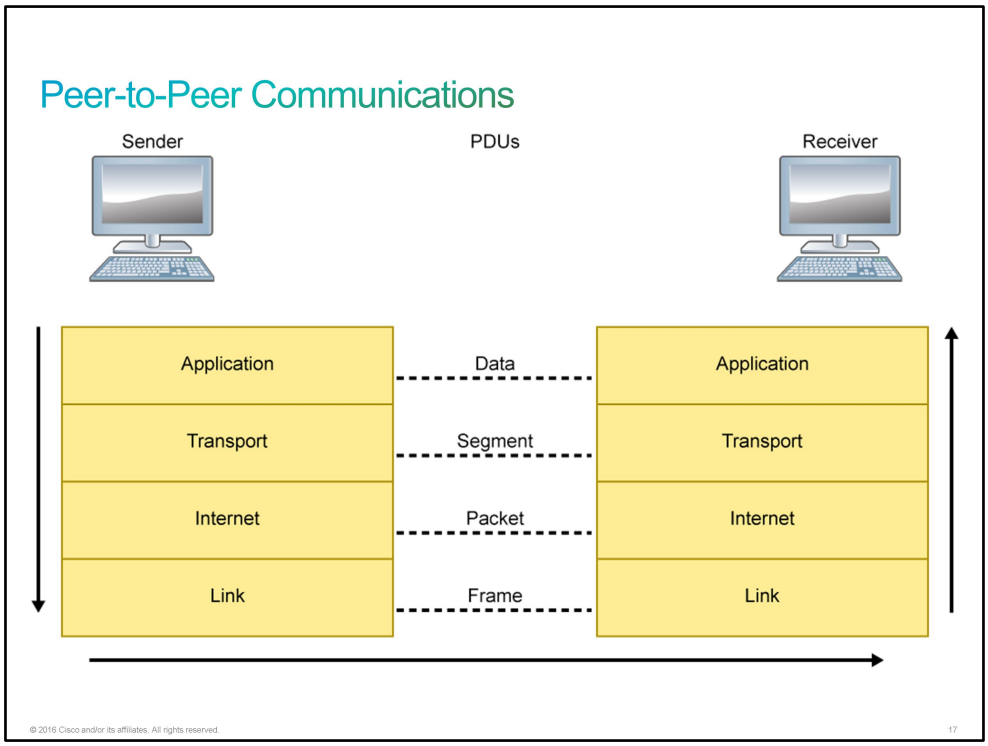
Take a look at the four layers of the TCP/IP model:

- **Link layer:** This layer is also known as the network access layer and is the equivalent of both the physical and data link layers of the OSI model. It deals with components such as cables, connectors, and network cards, like OSI Layer 1. Like Layer 2 of the OSI model, the link layer of the TCP/IP model is concerned with hardware addresses.
- **Internet layer—aligns directly with Layer 3 of the OSI model:** You may also know this layer as the Internet layer. It routes data from the source to the destination by defining the packet and the addressing scheme, moving data between the link and transport layers, routing packets of data to remote hosts, and performing fragmentation and reassembly of data packets. This is the layer where IP operates.
- **Transport layer—directly aligned with Layer 4 of the OSI model:** This layer is the core of the TCP/IP architecture. It is the layer where TCP and UDP operate. This layer provides communication services directly to the application processes that are running on network hosts.
- **Application layer—corresponds to Layers 5, 6, and 7 of the OSI model:** It provides applications for file transfer, network troubleshooting, and Internet activities. It also supports network APIs, which allow programs that have been created for a particular operating system to access the network.

Peer-to-Peer Communications

The term *peer* means the equal of a person or object. Therefore, peer-to-peer communication means communications between equals. In other words, each layer must be able to communicate with its equal (peer) on the other side. So each source layer must be able to communicate with its corresponding destination layer. One way that this understanding is accomplished is by packing data in a format that the peer layer will understand.

During the process of peer-to-peer communication, the protocols at each layer exchange packets between peer layers. These packets of information are called PDUs. At each stage of the process, a PDU has a different name to reflect its new appearance.



Although there is no universal naming convention for PDUs, here, the PDUs are named as follows:

- **Data:** The general term for the PDU used at the application layer
- **Segment:** A transport layer PDU
- **Packet:** An Internet layer PDU
- **Frame:** A link layer PDU

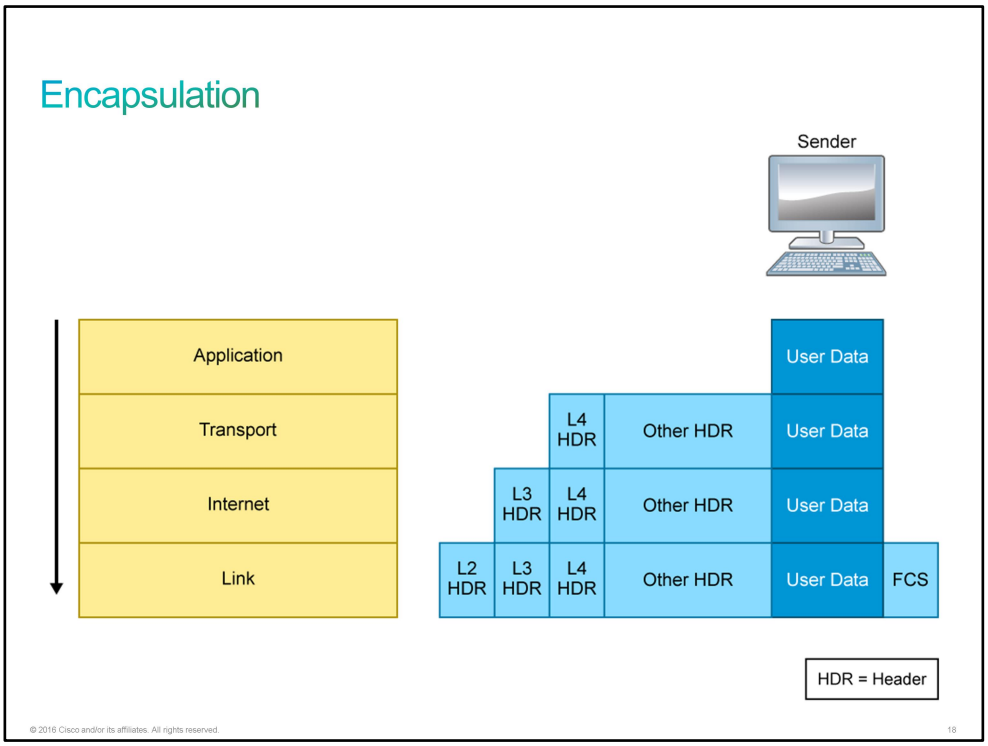
To look into PDUs from peer-to-peer communication, you can use a program for analyzing packets—for example, Wireshark. Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education.

Encapsulation and De-Encapsulation

Information that is transmitted over a network must undergo a process of conversion at the sending and receiving ends of the communication. The conversion process is known as encapsulation and de-encapsulation of data.

Encapsulation

Have you ever opened a very large present and found a smaller box inside? And then an even smaller box inside that one, until you get to the smallest box and, finally, to your present? Encapsulation operates similarly in the OSI model. The application layer receives the user data and adds a header before sending it to the presentation layer, like putting it into a box. The presentation layer then adds its own header before sending to the session layer, placing the small box into a larger one. This process continues at each layer. At the data link layer, a trailer is also added. The data is then sent across the physical layer. When it reaches the destination, each layer removes the header that its peer layer added, equivalent to unpacking the boxes. The layer reads the information in the header to determine what to do with the data and then hands the PDU up to the next layer for processing.

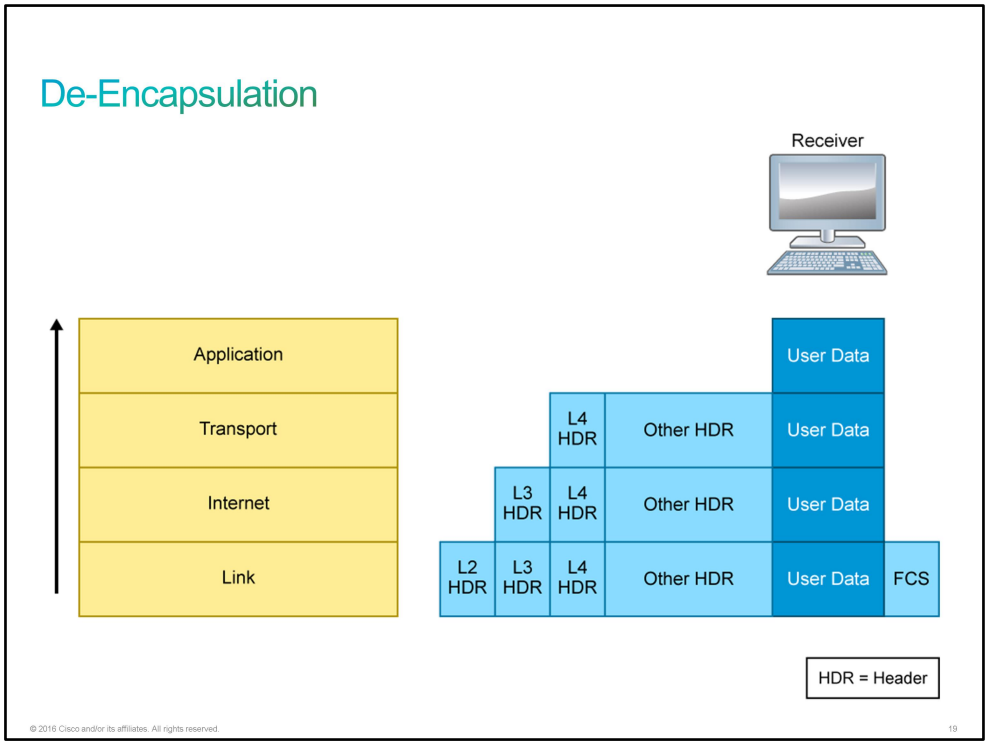


The information that is sent on a network is referred to as data or data packets. As application data is passed down the protocol stack on its way to be transmitted across the network media, various protocols add information to it at each level. This process is commonly known as the encapsulation process. Each layer adds a header (and a trailer, if applicable) to the data before passing it down to a lower layer. The headers and trailers contain control information for the network devices and the receiver. This information ensures that the data is properly delivered and that the receiver can correctly interpret the data.

The figure shows how encapsulation occurs. It shows how data travels through the layers. The data is encapsulated as follows:

1. The user data is sent from an application to the application layer.
2. The transport layer adds the transport layer header (Layer 4 header) to the data. The Layer 4 header and the previous data become the data that is passed down to the Internet layer.
3. The Internet layer adds the Internet layer header (Layer 3 header) to the data. The Layer 3 header and the previous data become the data that is passed down to the link layer.
4. The link layer adds the Layer 2 header and trailer to the data. A Layer 2 trailer is usually the FCS, which the receiver uses to detect whether the data is in error.

De-Encapsulation



When receiving messages on a network, the protocol stack on a host operates from the bottom to the top. The process of encapsulation is reversed at the receiving host. When the data reaches the destination, each layer removes the header that its peer layer added, equivalent to unpacking the boxes. The layer reads the information in the header to determine what to do with the data and then hands the PDU up to the next layer for processing. The data is de-encapsulated as it moves up the stack toward the end-user application.

When the remote device receives a sequence of bits, the data is de-encapsulated as follows:

1. The link layer checks the trailer (the FCS) to see if the data is in error. The frame may be discarded or the link layer may ask for the data to be retransmitted.
2. If the data is not in error, the link layer reads and interprets the control information in the Layer 2 header.
3. The link layer strips the Layer 2 header and trailer and then passes the remaining data up to the Internet layer, which is based on the control information in the link layer header.

Each subsequent layer performs a similar de-encapsulation process.

The de-encapsulation process is like reading the address on a package to see if it is addressed to you and then removing the contents of the package.

Challenge

1. Which OSI layer defines services to segment the data?

   A. presentation layer
   B. session layer
   C. transport layer
   D. network layer

2. Which OSI layer manages sessions between two communicating hosts?

   A. transport layer
   B. presentation layer
   C. session layer
   D. network layer

3. Which OSI layer ensures that the application layer of the receiving system will be able to read the information that the application layer of another system has sent?

   A. transport layer
   B. presentation layer
   C. session layer
   D. network layer

4. To which OSI model layer or layers is the TCP/IP model transport layer aligned?

   A. network
   B. transport
   C. session
   D. session, presentation, and application

5. To which OSI model layer or layers does the TCP/IP model application layer correspond?

   A. network
   B. transport
   C. session
   D. session, presentation, and application

6. Match the PDUs with the correct descriptions.

segment	the general term for the PDU used at the application layer
packet	a transport layer PDU
frame	an Internet layer PDU
data	a link layer PDU

7. Align TCP/IP layers with the corresponding PDUs.

packet	application layer
segment	transport layer
data	Internet layer
frame	link layer

Answer Key

Challenge

1. C
2. C
3. B
4. B
5. D
6.

data	the general term for the PDU used at the application layer
segment	a transport layer PDU
packet	an Internet layer PDU
frame	a link layer PDU

7.

data	application layer
segment	transport layer
packet	Internet layer
frame	link layer

Lesson 3: Introducing LANs

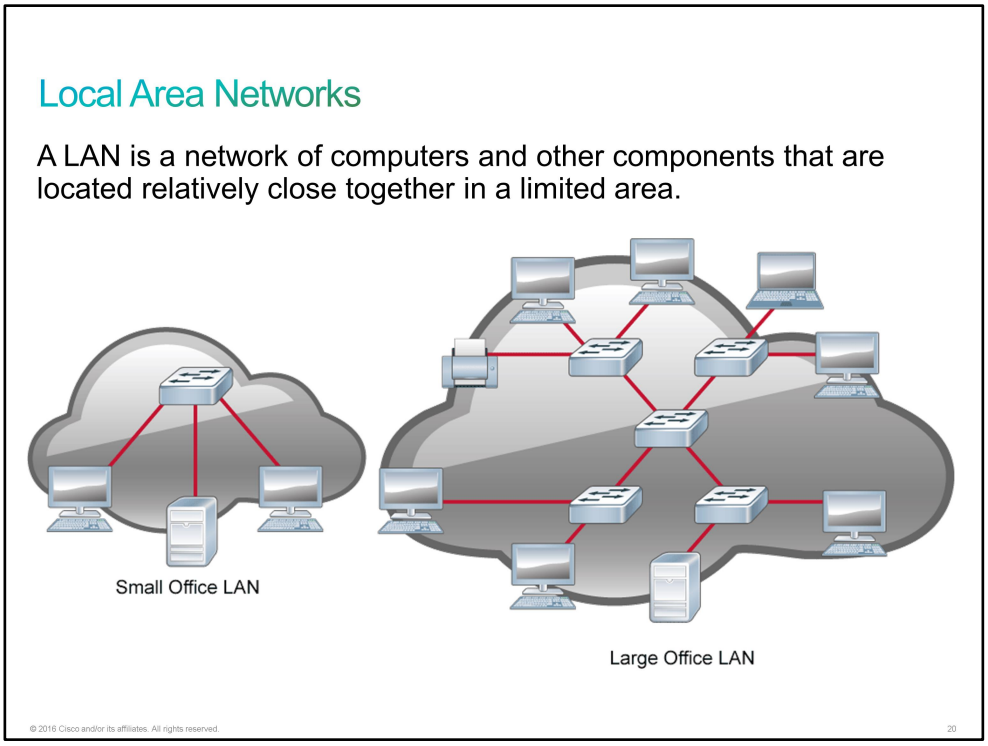
Introduction

Your first interview with Bob was a success. Maya calls the following afternoon to set up a second interview and provide you information about what you will need to know. She says that this time, you will be asked about LANs and their components, especially switches. She asks if you would like to come in for the interview now, because Bob is available all afternoon.

You need to know what a LAN is and be able to identify LAN components. You should also be able to explain why you need switches and list important switch features and characteristics.

Local Area Networks

The LAN emerged to serve the needs of a disconnected collective. While there have been many types of LAN transports, Ethernet became the favorite of businesses starting in the early 1990s. Since its introduction, Ethernet bandwidth has scaled from the original shared-media 10 Mb/s to 100 Gb/s in Cisco Nexus 7000 Series Switches for the data center.



LANs can vary widely in size. A LAN may consist of only two computers in a home office or small business, or it may include hundreds of computers in a large corporate office or multiple buildings.

The defining characteristics of LANs, in contrast to WANs, include their typically higher data transfer rates, smaller geographic area, and the lack of need for leased telecommunication lines.

Examples: A Small-Office LAN and a Large-Office LAN

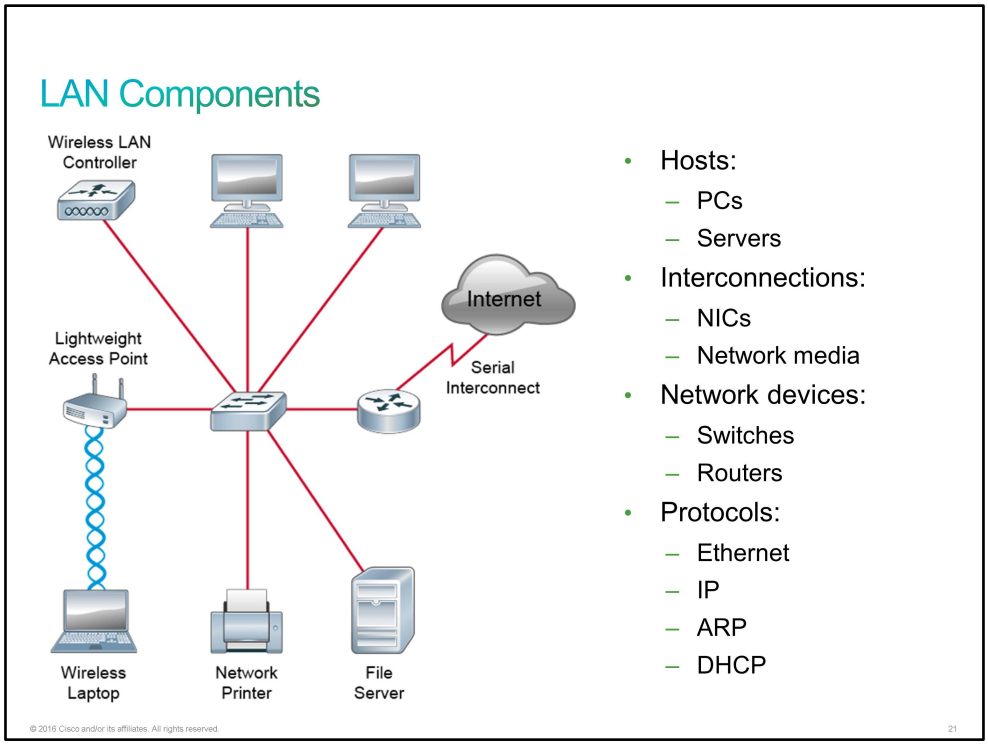
A small home business or a small-office environment can use a small LAN to connect two or more computers and to connect the computers to one or more shared peripheral devices, such as printers.

A large corporate office can use multiple LANs to accommodate hundreds of computers and shared peripheral devices, spanning many floors in an office complex.

LAN Components

On the first LANs, devices with Ethernet connectivity were mostly limited to PCs, file servers, print servers, hubs, and bridges.

Today, a typical small office will include a router for Internet connectivity, wireless capabilities, network printers, file servers, PCs, and laptops.



Regardless of its size, a LAN requires these fundamental components for its operation:

- **Hosts:** Hosts include any device that can send or receive data on the LAN.
- **Interconnections:** Interconnections allow data to travel from one point to another in the network. Interconnections include these components:
  - **NICs:** NICs translate the data that is produced by the computer into a format that can be transmitted over the LAN. NICs connect a station to the LAN over copper cable, fiber-optic cable, or wireless communication.
  - **Network media:** In traditional LANs, data was transmitted mostly over copper and fiber-optic cables. Modern LANs (even small home LANs) generally include wireless connectivity.
- **Network devices:** Network devices, like switches and routers, are responsible for data delivery between hosts.
- **Protocols:** Protocols are rules that govern how data is transmitted over a LAN. Here are some commonly used LAN protocols:
  - Ethernet protocols (IEEE 802.2 and IEEE 802.3)
  - IP
  - TCP
  - UDP

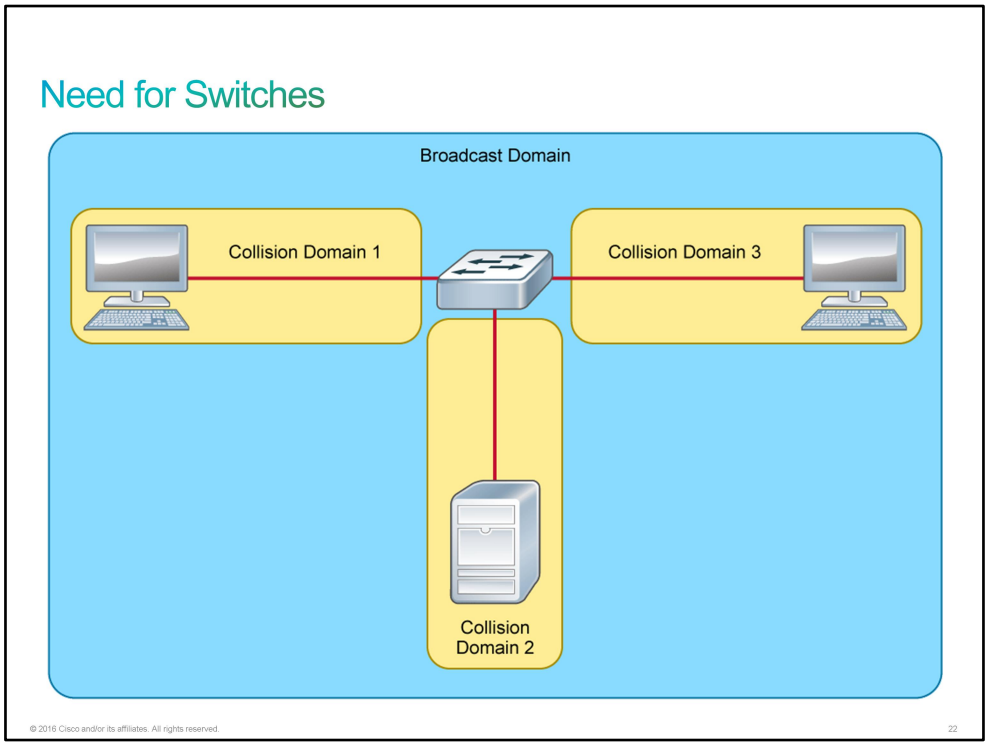
- ARP and RARP
- CIFS
- DHCP

Need for Switches

When you connect three or more devices together, you need a dedicated network device to enable communication between these hosts.

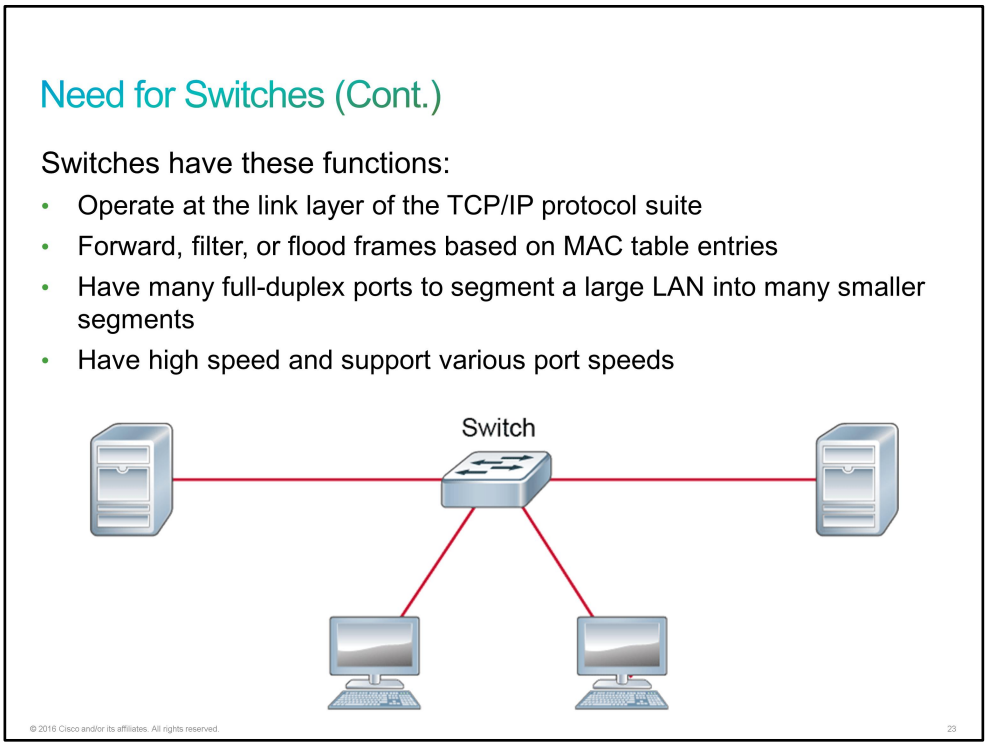
Historically, when network devices had few Ethernet segments, end host devices had to compete for the same bandwidth, and only one device was able to transmit data at a time. Network segments that share the same bandwidth are known as *collision domains*, because when two or more devices within that segment try to communicate at the same time, collisions may occur.

Today, it is common to use switches as network devices, operating at the link layer of the TCP/IP protocol suite, to divide a network into segments and reduce the number of devices that compete for bandwidth. Each new segment, then, results in a new collision domain. More bandwidth is available to the devices on a segment, and collisions in one collision domain do not interfere with the working of the other segments.



As shown in the figure, each switch port connects to a single PC or server. Each switch port represents a unique collision domain.

The *broadcast domain* is another key concept. The filtering of frames by switches, which is done based on their MAC addresses, does not extend to filtering broadcast frames. By their nature, broadcast frames must be forwarded. Therefore, a port on a switch forms a single broadcast domain. It takes a Layer 3 entity, such as a router, to terminate a Layer 2 broadcast domain.



Ethernet switches selectively forward individual frames from a receiving port to the port where the destination node is connected. This selective forwarding process can be thought of as establishing a momentary point-to-point connection between the transmitting and receiving nodes. The connection is made only long enough to forward a single frame. During this instant, the two nodes have a full-bandwidth connection between them and represent a logical point-to-point connection.

The switch builds and maintains a table, which is called a MAC table. This table matches a destination MAC address with the port that is used to connect to a node. For each incoming frame, the destination MAC address in the frame header is compared to the list of addresses in the MAC table. Switches then use MAC addresses as they decide whether to filter, forward, or flood frames.

The table shows how switches process unicast frames.

How Switches Process Unicast Frames on an Ethernet LAN

Step	Action
1	When a unicast frame is received on a port, the switch compares the destination MAC address to the MAC addresses that it has listed in its table.
2	If the switch determines that the destination MAC address of the frame resides on the same network segment as the source, it does not forward the frame. This process is called filtering. By performing this process, switches can significantly reduce the amount of traffic going between network segments by eliminating the unnecessary frames.
3	If the switch determines that the destination MAC address of the frame is not in the same network segment as the source, it forwards the frame to the appropriate segment.

Step	Action
4	If the switch does not have an entry in its table for the destination address, it transmits the frame out of all ports except the port on which it received the frame. This process is called *flooding*.

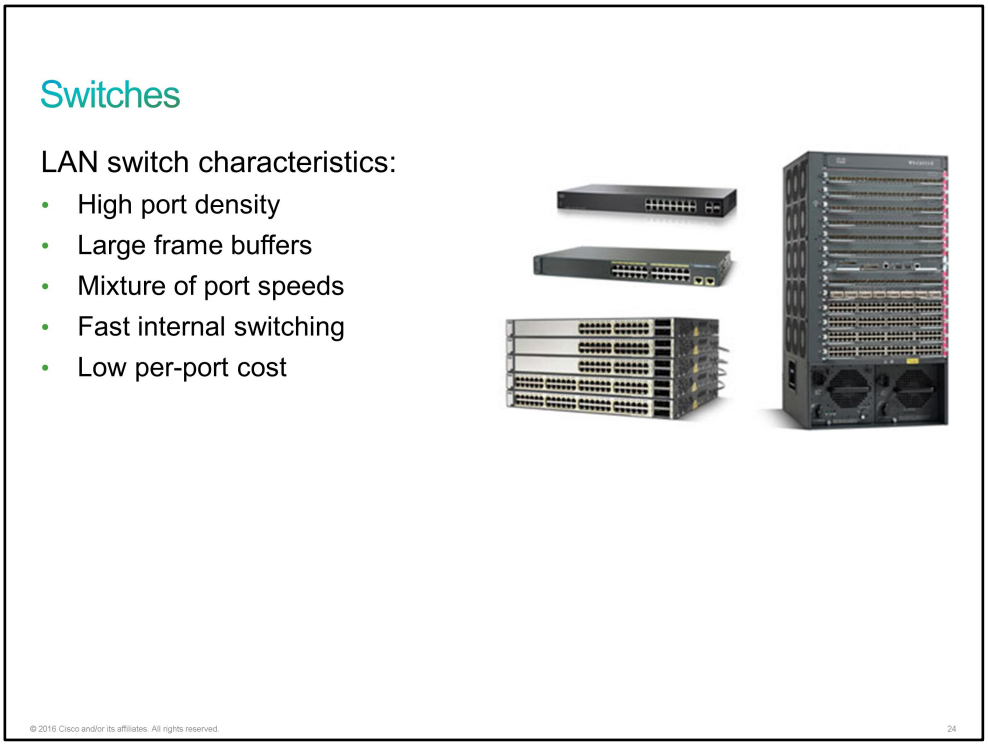
Switches

Switches have become a fundamental part of most networks. They allow the segmentation of a LAN into separate collision domains. Each port of the switch represents a separate collision domain and provides the full media to the node or nodes that are connected on that port. The introduction of full-duplex communications (a connection that can carry transmitted and received signals at the same time) has enabled 1 Gbps Ethernet and beyond.

Instead of Ethernet stations connecting to shared media, stations connect directly to a port on the LAN switch, providing two features that are unavailable in shared Ethernet LANs:

- **Dedicated bandwidth:** Because only one station is connected to a LAN switch port, the station does not compete for access to the media with other Ethernet stations. So, it receives the full bandwidth that is configured on the port.
- **Full-duplex operation:** In classical shared Ethernet, a station can either transmit or receive at a given time, which is referred to as half-duplex operation. Because the Ethernet station now directly connects its transmit and receive wires to the switch port, it can simultaneously transmit and receive. This mode is termed full duplex. Half-duplex mode is available for legacy 10BASE-T NICs that do not support full-duplex operation.

LAN switches have special characteristics that make them effective in alleviating network congestion.



Switches connect LAN segments, use a table of MAC addresses to determine the segment to which it will send the data, and reduce network traffic. The following are some important characteristics of switches:

- **High port density:** Switches have high port densities: 24- and 48-port switches operate at speeds of 100 Mbps, 1 Gbps, and 10 Gbps. Large enterprise switches may support 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, they may support a mixture of speeds. You would expect ports of 100 Mbps, but switches offering ports that support 1 or 10 Gbps are more common.
- **Fast internal switching:** Having fast internal switching allows many speeds: 100 Mbps, 1 Gbps, and 10 Gbps. The method that is used may be a fast internal bus, shared memory, or integrated crossbar switch fabric, which affects the overall performance of the switch.
- **Low per-port cost:** Switches provide high port density at a lower cost. For this reason, LAN switches can accommodate network designs which feature fewer users per segment. This feature, therefore, increases the average available bandwidth per user.

Challenge

1. What are the two defining characteristics of a LAN? (Choose two.)

   A. higher data transfer rates in contrast to WAN
   B. lower data transfer rates in contrast to WAN
   C. smaller geographic area in contrast to WAN
   D. larger geographic area in contrast to WAN
   E. need for leased telecommunication lines

2. What are typical LAN host components? (Choose two.)

   A. switches
   B. routers
   C. servers
   D. hosts
   E. firewalls

3. What are the two typical LAN network components? (Choose two.)

   A. switches
   B. routers
   C. servers
   D. hosts
   E. firewalls

4. Which devices or protocols are responsible for sending or receiving data on the LAN?

   A. hosts
   B. interconnections
   C. network devices
   D. TCP

5. Which devices or protocols allow data to travel from one point to another in the network?

   A. hosts
   B. interconnections
   C. network devices
   D. TCP

6. What are the three functions of switches? (Choose three.)

   A. have high speed and support single port speeds
   B. operate at the link layer of the TCP/IP protocol suite
   C. operate at the network layer of the TCP/IP protocol suite
   D. forward, filter, or flood frames based on MAC table entries
   E. forward, filter, or flood packets based on IP routing table entries
   F. have many full-duplex ports to segment a large LAN into many smaller segments

7. Which switch characteristic can accommodate network designs that feature fewer users per segment?

   A. high port density
   B. large frame buffers
   C. port speed
   D. fast internal switching
   E. low per-port cost

Answer Key

Challenge

1. A, C
2. C, D
3. A, B
4. A
5. B
6. B, D, F
7. E

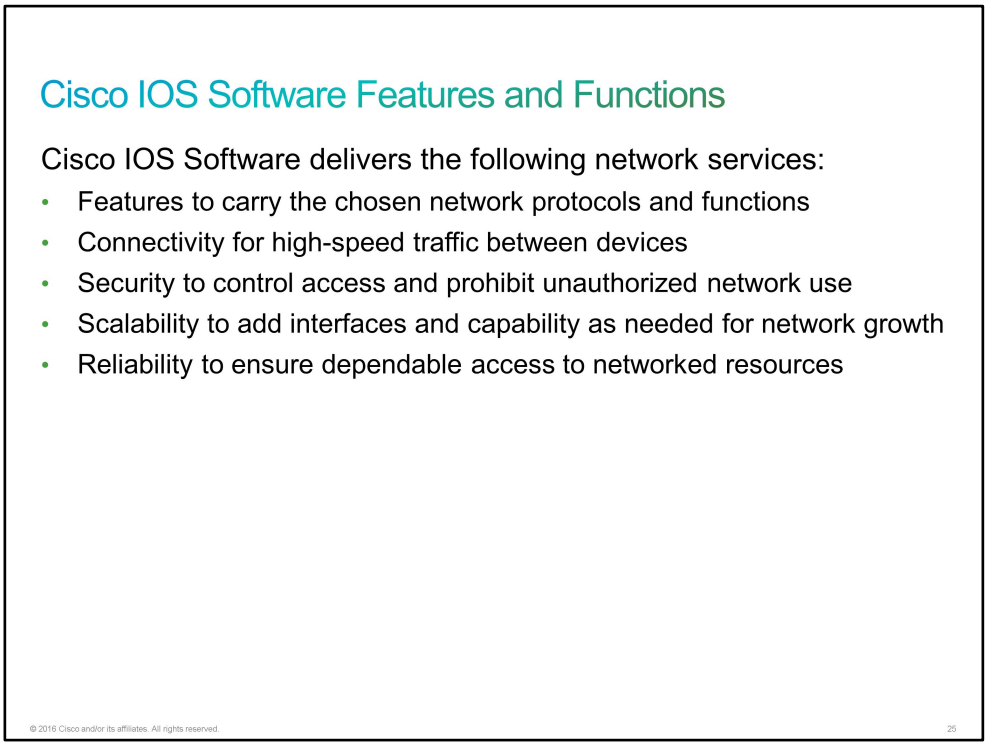
Lesson 4: Operating Cisco IOS Software

Introduction

Maya calls to set up the final technical interview with Bob. She says that, in the final phase, you will need to demonstrate your ability to perform essential tasks in Cisco IOS Software, such as managing Cisco IOS configurations, using the built-in help functionality, and improving the user experience in the CLI. She asks if you would like some time to practice.

Cisco IOS Software Features and Functions

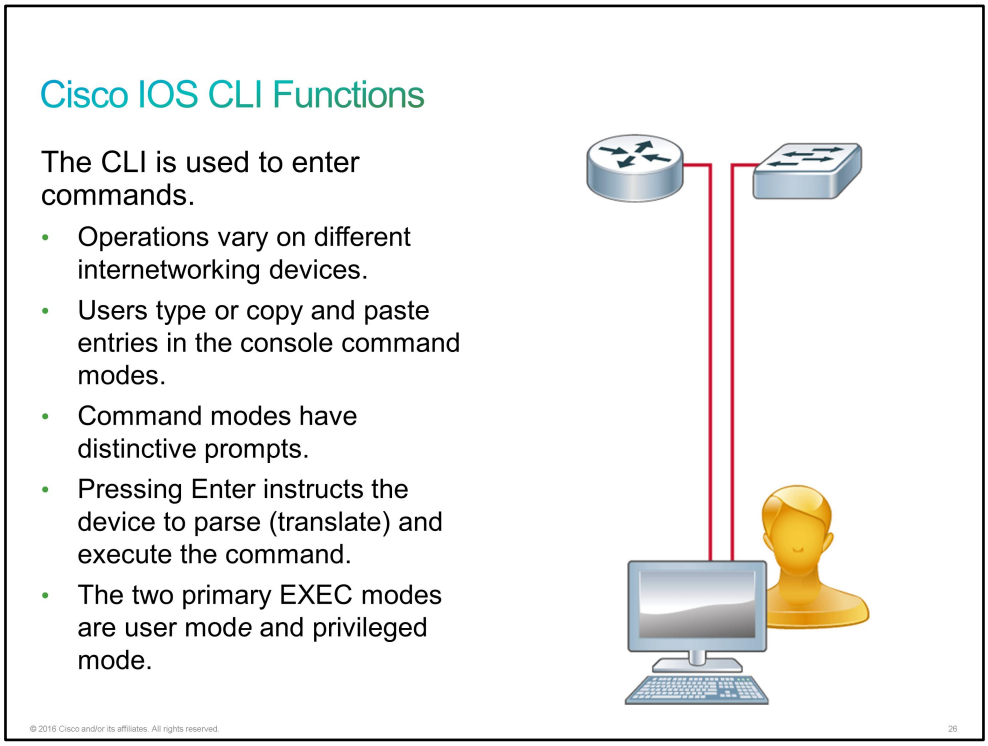
Like a computer, a switch and a router also need an operating system to function. An operating system is the software that manages how various hardware components of a device function together. Just as an office needs a manager to supervise workers, a computing device needs an operating system. Many Cisco devices use Cisco IOS Software as their operating system. It is the core technology that extends across most Cisco devices, regardless of the size and type of the device. Many devices use the Cisco IOS Software—for example, routers, LAN switches, small wireless APs, large routers with numerous interfaces, and so on.



The services that are provided by Cisco IOS Software are generally accessed using a CLI. The CLI is a text-based interface that is similar to the old Microsoft operating system that is called MS-DOS. The CLI is accessed through a direct cabled connection that is called the *console connection*, a modem connection, or a Telnet or SSH session. Regardless of which connection method you use, access to the Cisco IOS CLI is generally referred to as an EXEC session. The features that you can access via the CLI vary—it depends on the version of the Cisco IOS Software and the type of device.

Cisco IOS CLI Functions

Cisco IOS Software uses a CLI via the console as its traditional environment to enter commands. While Cisco IOS Software is a core technology that extends across many products, the details of its operation vary on different internetworking devices.



Cisco IOS Software is designed as a modal operating system. The term *modal* describes a system which has various modes of operation, each having its own domain of operation. The CLI uses a hierarchical structure for the modes.

To enter commands into the CLI, type or copy and paste the entries within one of the several console command modes. Each command mode is indicated with a distinctive prompt. The term *prompt* is used because the system is prompting you to make an entry. By default, every prompt begins with the device name. Following the name, the remainder of the prompt indicates the mode. As you use commands and change mode, the prompt changes to reflect the current context. Pressing Enter instructs the device to parse and execute the command.

Note	It is important to remember that as soon as you enter a command, the command is executed. If you enter an incorrect command in a production router, it can negatively affect the network.

Cisco IOS Software uses a hierarchy of commands in its command-mode structure. Each command mode supports specific Cisco IOS commands that are related to the type of operation on the device.

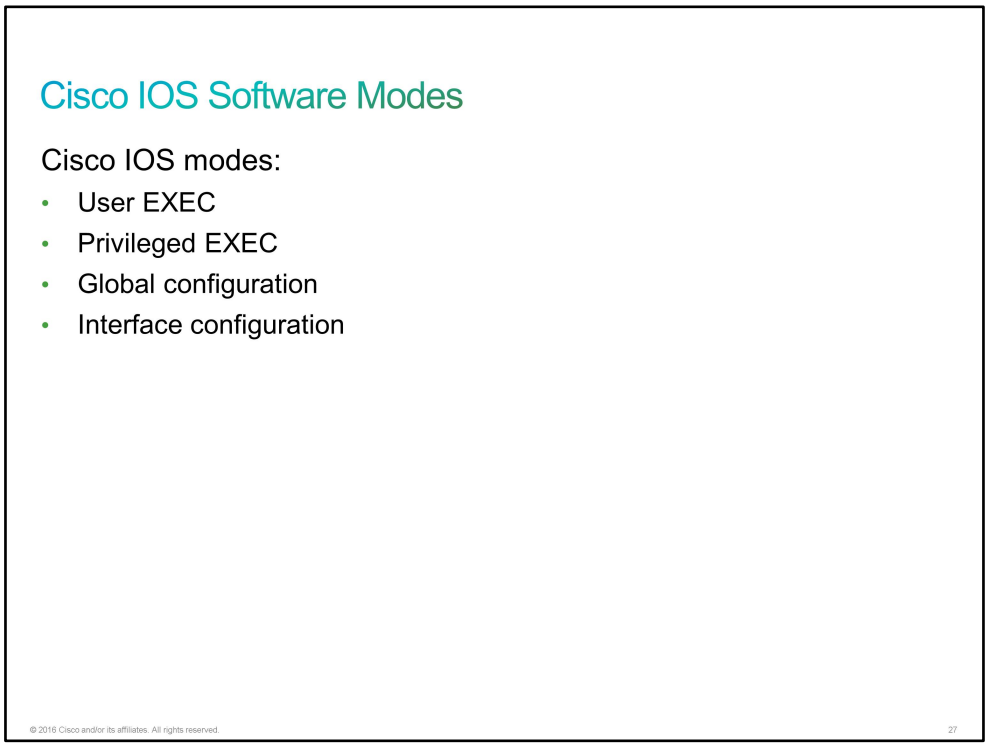
As a security feature, Cisco IOS Software separates EXEC sessions into two access levels:

- **User EXEC:** Allows a person to access only a limited number of basic monitoring commands.
- **Privileged EXEC:** Allows a person to access all device commands, such as those that you use for configuration and management. This level can be password-protected to allow only authorized users to access the device.

Cisco IOS Software Modes

Cisco IOS Software on a Cisco switch has various configuration modes that are hierarchically structured. The major modes are user EXEC, privileged EXEC, global configuration, and interface configuration.

Because these modes have a hierarchy, you can only access a lower-level mode from a higher-level mode. For example, in order to access global configuration mode, you must be in the privileged EXEC mode. Each mode is used to accomplish particular tasks and has a specific set of commands that are available in that mode. For example, to configure a switch interface, you must be in interface configuration mode. All configurations that you enter in interface configuration mode apply only to that interface.



The following table offers more detail on each mode:

Mode	Access Method	Prompt	Exit Method	About This Mode
User EXEC	Begin a session with your switch.	Switch>	Enter **logout** or **quit**.	Use this mode to change terminal settings, perform basic tests, or display system information.
Privileged EXEC	While in user EXEC mode, enter the **enable** command.	Switch#	Enter **disable** or **exit**.	Use this mode to verify commands that you have entered. Use a password to protect access to this mode.

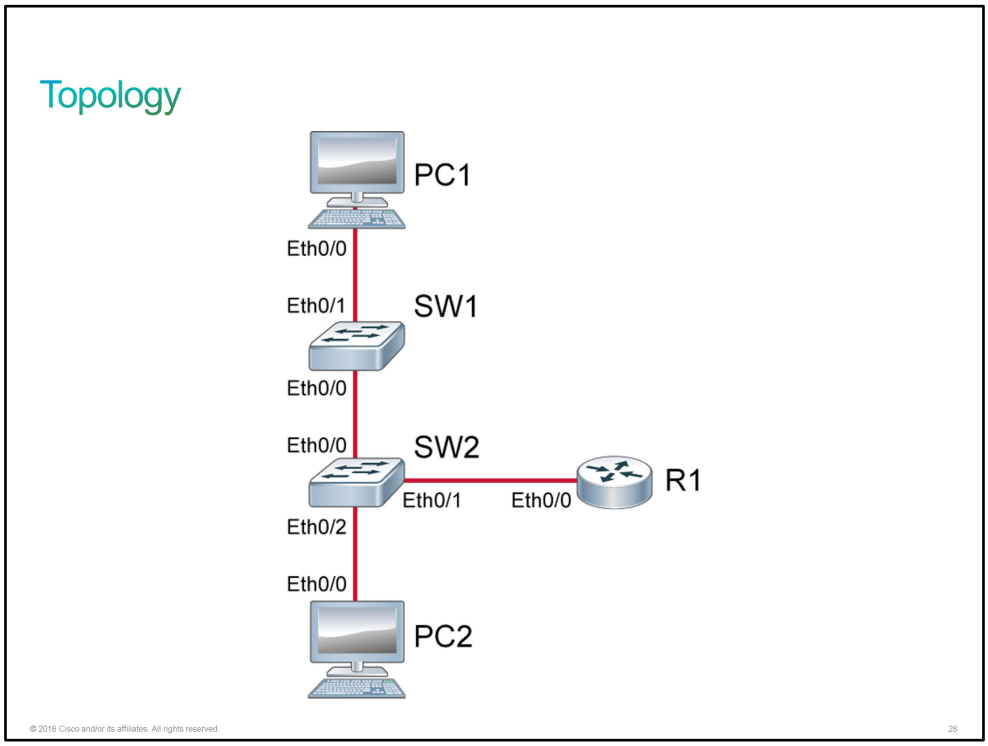
Mode	Access Method	Prompt	Exit Method	About This Mode
Global configuration	While in privileged EXEC mode, enter the **configure** command.	Switch(config)#	To exit to privileged EXEC mode, enter **exit** or **end**, or press **Ctrl-Z**.	Use this mode to configure parameters that apply to the entire switch.
Interface configuration	While in global configuration mode, enter the **interface** command (with a specific interface).	Switch(config-if)#	To exit to global configuration mode, enter **exit**. To return to privileged EXEC mode, press **Ctrl-Z** or enter **end**.	Use this mode to configure parameters for the Ethernet interfaces.

Discovery 1: Get Started with Cisco CLI

Introduction

In this discovery lab, you will learn about EXEC modes, CLI help, and the CLI error message. You will also learn how to manage Cisco IOS configuration and how to improve user experience in CLI.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2
PC2	IP address	10.10.1.20/24

Device	Characteristic	Value
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Navigate Between EXEC Modes

This session will guide you through the navigation between user EXEC and privileged EXEC on the Cisco IOS command line. The lab is prepared with the devices that are represented in the topology, but for this session you will only be using SW2.

Activity

Step 1 Access the console of SW2.

The greater than symbol (>) at the end of the prompt is an indication that you are accessing the user EXEC.


```
SW2>
```

Step 2 Use the question mark (**?**) to view the list of commands that are available in user EXEC.

When the display output pauses with the --More-- prompt, you can use the space bar to display the next page of the output.

```
SW2> ?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  clear            Reset functions
  connect          Open a terminal connection
  crypto           Encryption related commands.
  disable          Turn off privileged commands
  <... output omitted ...>
  mtrace           Trace reverse multicast path from destination to source
  name-connection  Name an existing network connection
 --More-- <space bar>
  pad              Open a X.29 PAD connection
  ping             Send echo messages
  <... output omitted ...>
  where            List active connections
  x3               Set X.3 parameters on PAD
```

Note	You have to press the space bar twice to scroll through the complete command list under the user EXEC. Have this information in mind, because you will soon contrast it to what you will be able to see in the privileged EXEC mode.

Note	The commands are listed in alphabetical order. Note that the **configure** command is not available under user EXEC.

Note	In the outputs, like in the previous one, many lines are omitted, due to space preservation. Omitted lines are indicated with <... output omitted ...> string.

Step 3 As you just saw, when you are presented with the --More-- prompt, you can use the space bar to scroll through the output page by page.

You can also use the Enter key to scroll forward just one line. You can also cancel the remaining output. The method to cancel the remaining output is device and operating system-version dependent. Sometimes you need to press Ctrl-C and sometimes you need to press "Q." On SW2, you can press any key other than the space bar or the Enter key. Give it a try!

```
SW2> ?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  clear            Reset functions
  connect          Open a terminal connection
  crypto           Encryption related commands.
  disable          Turn off privileged commands
  <... output omitted ...>
  mtrace           Trace reverse multicast path from destination to source
  name-connection  Name an existing network connection
--More--  <Enter>
  pad              Open a X.29 PAD connection
--More--  <Enter>
  ping             Send echo messages
--More--  <Ctrl-C>
SW2>
```

Entering EXEC Mode

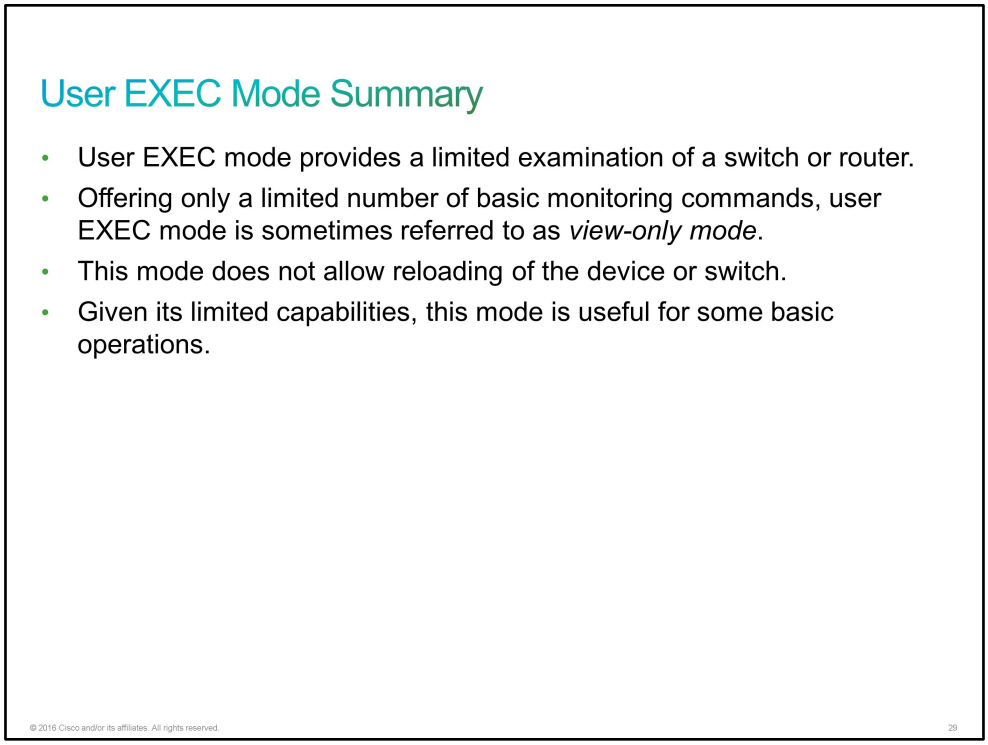
As a security feature, Cisco IOS Software separates EXEC sessions into the following two access levels:

- **User EXEC:** Allows you to access only a limited number of basic monitoring commands. When in EXEC mode, the prompt ends with the greater than or right angle bracket (>) symbol. For example, when you are in EXEC mode on a device with the hostname DTW_Switch, the prompt would be DTW_Switch>.
- **Privileged EXEC:** Allows you to access all device commands, such as those that you would use for configuration and management. It can be password-protected to allow only authorized users to access the device. When in this mode, the prompt ends with the octothorpe or pound (#) symbol. For example, when you are in privileged EXEC mode on a device with the hostname DTW_Switch, the prompt would look like DTW_Switch#. To change from user EXEC mode to privileged EXEC mode, enter the **enable** command at the hostname> prompt. To return to the user EXEC level, enter the **disable** command at the hostname# prompt.

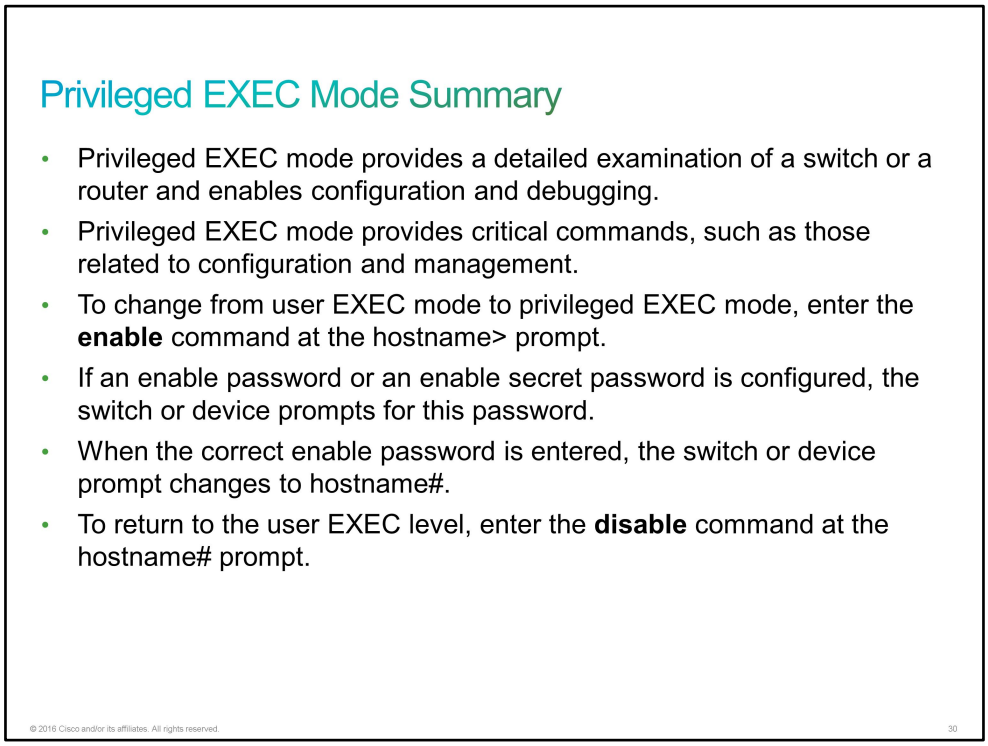
By default, no authentication is required to access user EXEC mode from the console. You can enter the EXEC mode by simply pressing the Enter key. However, if login is configured, you must enter a username and password to enter the EXEC mode. It is a good practice to ensure that authentication is configured during the initial configuration.

Entering the question mark (**?**) in privileged EXEC mode reveals many more command options than entering the command at the user EXEC level. This feature is referred to as *context-sensitive help*.

User EXEC Mode Summary



Privileged EXEC Mode Summary



Step 4 Use the **enable** command to access the privileged EXEC.

The last character in the prompt has changed to the octothorpe (#) symbol. This symbol indicates to you that you are in privileged EXEC.

```
SW2> enable
SW2#
```

Step 5 Use the **?** command again to display the commands that you can use under privileged EXEC. Use the space bar to scroll through the entire list of the output.

```
SW2# ?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  access-template  Create a temporary Access-List entry
  archive          manage archive files
  beep             Blocks Extensible Exchange Protocol commands
  calendar         Manage the hardware calendar
  cd               Change current directory
  clear            Reset functions
  clock            Manage the system clock
  cns              CNS agents
  configure        Enter configuration mode
  connect          Open a terminal connection
  <... output omitted ...>
  enable           Turn on privileged commands
  eou              EAPoUDP
--More--  <Enter>
  erase            Erase a filesystem
  <... output omitted ...>
```

Note	Under privileged EXEC, you needed to press the space bar four times to get through the entire list of commands. Under user EXEC, you only needed to hit the space bar twice.

Note	Under privileged EXEC, you can use the configuration command. You cannot proceed to the configuration mode from the user EXEC—you must traverse through the privileged EXEC first.

Step 6 Use the **disable** command to return to user EXEC.

```
SW2# disable
SW2>
```

Note	The last character in the system prompt has returned to the greater than sign (>).

Task 2: Explore CLI Help

This session will guide you through using the question mark (**?**) command for help on the IOS CLI. It will also demonstrate how you can take advantage of the tab completion feature of the IOS CLI. The lab is prepared with the devices that are represented in the topology, but for this session you will only be using SW2. You will also take a look at CLI error messages.

Activity

Step 1 On SW2, use the **enable** command to access privileged EXEC.

```
SW2> enable
SW2#
```

Step 2 Use the question mark (**?**) to display all the commands that you can use under privileged EXEC.

Use the space bar to scroll through the entire list of the output.

```
SW2# ?
Exec commands:
  access-enable    Create a temporary Access-List entry
  access-profile   Apply user-profile to interface
  access-template  Create a temporary Access-List entry
  archive          manage archive files
  beep             Blocks Extensible Exchange Protocol commands
  calendar         Manage the hardware calendar
  cd               Change current directory
  clear            Reset functions
  clock            Manage the system clock
  cns              CNS agents
  configure        Enter configuration mode
  connect          Open a terminal connection
  <... output omitted ...>
  enable           Turn on privileged commands
  eou              EAPoUDP
--More--  <space bar>
  <... output omitted ...>
```

Note	The list is quite long. You have to use the space bar four times to get through the entire list.

CLI Help

When you are learning a new program or interface, you usually depend on the Help features the program offers. Cisco IOS Software includes extensive command-line help functions, including context-sensitive help. There are two basic types of CLI keyboard help that the Cisco IOS devices enable. The first is context-sensitive help, which offers assistance when you are trying to determine the proper command and syntax. To use it, press the question mark (**?**) key. For example, you know that the command that you want to use starts with **sh**, but you are not sure what the rest of the command is. Enter **sh?** and you are presented with every command that starts with sh and that you can use in the current privilege mode. You can also use context-sensitive help to figure out the syntax for a command.

Another use of the context-sensitive help is to get a list of available commands for the current CLI mode. This list can be used when you are unsure of the name of a command or you want to see if Cisco IOS Software supports a particular command in a particular mode. To use context-sensitive help in this way, enter the question mark (**?**) at any prompt.

The other type of the CLI keyboard help is the error messages. When you enter a command in the CLI, the syntax is checked. If it is not correct, you receive an error that states "Invalid input detected at '^' marker." In addition to the message, the caret symbol (^) is added below the place in the command at which the error was detected. Basically, Cisco IOS Software is saying "I understood what you typed up to this point."

You may also receive an error message for an ambiguous command. This type of error occurs when you use an abbreviation for a command and the abbreviation results in multiple matches. In Cisco IOS Software, when you type enough letters that match only one command, you may press the Enter key. Because there is no other command that starts with those letters, Cisco IOS Software executes the command. For example, assume that there are several commands that start with the letter "c" but only one command that begins with "clo." If you press Enter after entering only **c**, you receive an "ambiguous command" error message. However, if you enter **clo** and press Enter, you do not receive a message that the command is ambiguous because the **clock** command is the only command that starts with those three letters. However, you would receive an "incomplete command" error message because clock is not a complete command. This message means that you did not enter enough information for Cisco IOS Software to understand what you were requesting.

Note	This functionality may vary across Cisco IOS platforms.

Note	The question mark is your friend when you are using the CLI. It is impossible to remember the syntax for every Cisco IOS command, so be sure to use this tool.

CLI Help

Type of CLI Help	Description
Context-sensitive help	Provides a list of commands and the arguments that are associated with a specific command.
Console error message	Identifies problems with commands that you have incorrectly entered so that you can alter or correct them.

CLI Help (Cont.)

How to utilize context-sensitive help?

- Word Help
  - To get word help, enter a character sequence followed immediately by a question mark. Do not include a space before the question mark. The device then displays a list of commands that start with the characters that you entered.
- Command Syntax Help
  - To get command syntax help, enter a question mark after a command name in place of a keyword or argument. Include a space before the question mark. For example, enter **show ?** to get a list of the command options that the **show** command supports. The network device then displays a list of available command options, with <cr> standing for carriage return. You can access command syntax help after any command or command option to help you determine what you can or should enter next.

Step 3 List only the commands that start with the letter "s" by entering **s?** on the command line.

```
SW2# s?
*s=show
sdlc        send    set        setup
show        slip    spec-file  ssh
start-chat  systat
```

The output shows that there is an exception to normal command parsing rules. The CLI will interpret the letter "s" all by itself as "show". This feature is specific to the device and operating system version. While it will work on SW2, it may not work on all devices. Abbreviating "show" with the characters "sh" is going to be effective more consistently across IOS devices.

Step 4 Try out the tab completion feature.

Like command abbreviation, tab completion works as long as you have entered enough characters to remove ambiguity. Type **sh** and then press the **tab** key.

The CLI parser expands the unambiguous abbreviation into the full command.

```
SW2# sh<tab>
SW2# show
```

Step 5 You might find tab completion helpful because it prevents you from attempting to use command abbreviation and accidentally abbreviate too much.

If there are multiple matches for the abbreviation, tab completion will not work. If you are not sure why, you can always use the question mark (**?**) at that point. Demonstrate this example by attempting to abbreviate the configure command with "con".

When you tried to use the tab to complete the abbreviation "con," it did not work. The command parser simply redisplayed "con". Using the question mark (**?**) at that point shows that there are two commands that begin with "con". To be unambiguous, you must use at least "conf" as your abbreviation for *configure*.

```
SW2# con<tab>
SW2# con?
configure  connect
SW2# con
```

Step 6 You will not go into the configuration mode during this session. Use the **Backspace** key to delete the "con" that is currently on the CLI input line.

Step 7 You have just demonstrated that the question mark (**?**) and tab completion work for commands.

They are also helpful for arguments to commands. For example, if you want to display all the arguments that you can use with the **show** command, use the question mark (**?**) and separate it from the **show** command by a space.

```
SW2# show ?
  aaa                Show AAA values
  access-expression  List access expression
  <... output omitted ...>
 --More—  <space bar>
  <... output omitted ...>
```


Note	There are a lot of **show** commands. To scroll through the entire list you have to press the space bar nine times.

Step 8 Just like with commands, you can combine some explicit characters followed by the question mark to display a subset of the argument options.

For example, use **show r?** to display all the show command options that start with the letter "r".

```
SW2# show r?
radius     region     registry        reload
resource   rhosts     rib             rif
route-map  route-tag  running-config
```

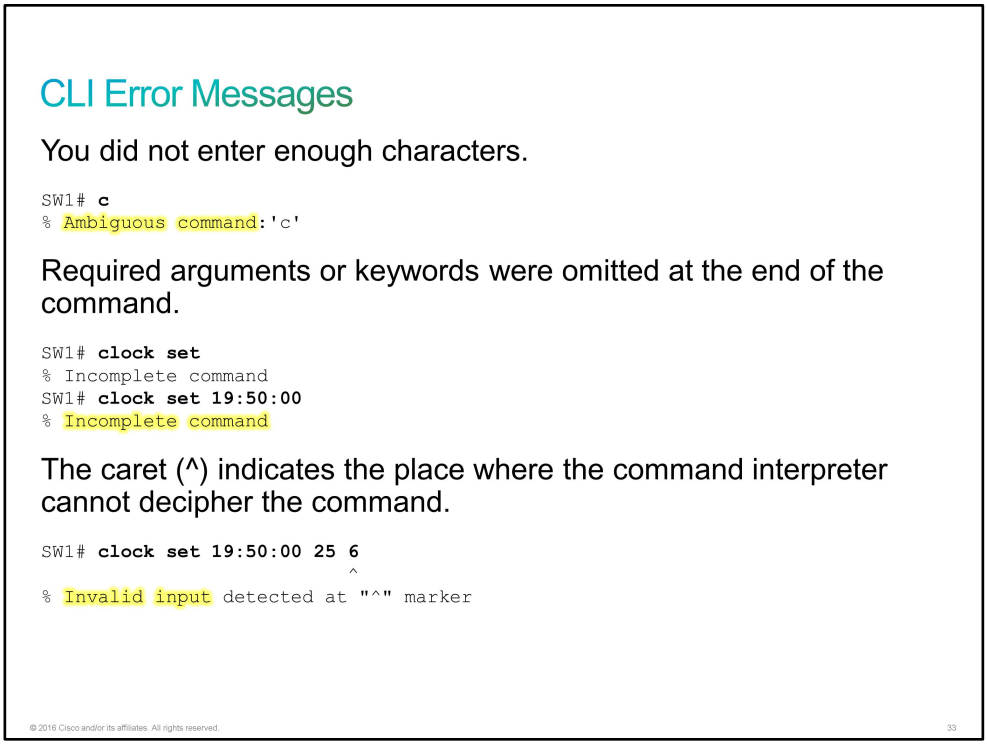
Step 9 Experiment with command abbreviation and tab completion in creative ways, until you feel you are comfortable using them.

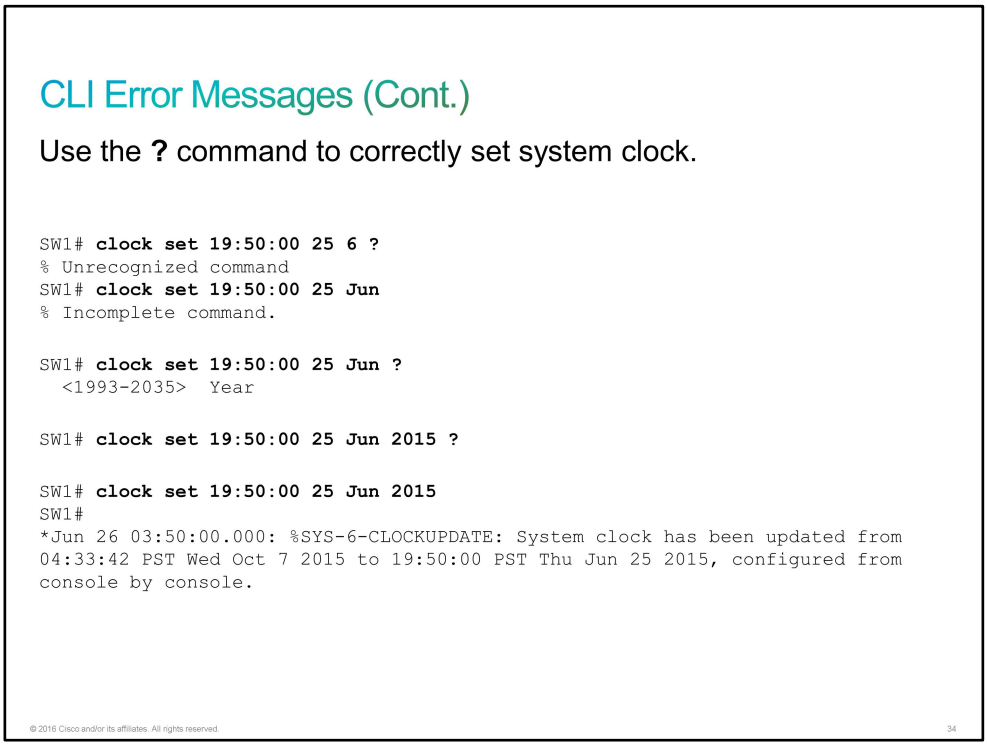
You can see one example for **show running-config**, but still, feel free to experiment independently.

```
SW2# sh<tab>
SW2# show run<tab>
SW2# show running-config
Building configuration...

Current configuration : 865 bytes
!
version 15.1
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service compress-config
<... output omitted ...>
```

Note	You may also find tab completion useful if you are working with someone else. If you are typing at the CLI, you may understand some command abbreviations that your partner does not. Using command completion allows your partner to see the entire command verbiage.

CLI Error Messages





There are three types of console error messages:

- Ambiguous command
- Incomplete command
- Incorrect command

CLI Error Messages (Cont.)

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for your device to recognize the command.	Re-enter the command, followed by **?** without a space before it. The CLI displays possible keywords that you can enter with the command.
% Incomplete command	You did not enter all the keywords or values that are required by this command.	Re-enter the command, followed by **?** with a space before it.
% Invalid input detected at '^' marker	You entered the command incorrectly. The ^ marks the point of the error.	Enter **?** to display all the commands or parameters that you can use.

The command history buffer stores the commands that have been most recently entered. To see these commands, enter the Cisco IOS **show history** command.

You can use context-sensitive help to determine the syntax of a particular command. For example, if the device clock needs to be set but you are not sure of the **clock** command syntax, the context-sensitive help provides a means to check the syntax.

Context-sensitive help supplies the whole command even if you enter just the first part of the command, such as **cl?**.

If you enter the command **clock** but an error message is displayed, indicating that the command is incomplete, enter the **?** command (preceded by a space) to determine which arguments are required for the command. In the **clock ?** example, the help output shows that the keyword **set** is required after **clock**.

If you now enter the command **clock set** but another error message appears, indicating that the command is still incomplete, press the **Up Arrow** key to repeat the command entry. Then, add a space and enter the question mark (**?**) to display a list of arguments that you can use for the command.

The example shows that after the last command recall, the administrator used the **?** to reveal additional arguments, which involve entering the current time using correct form of month and year..

The figure continues to illustrate how to set the device clock.

If after entering the current time you still see the Cisco IOS Software error message indicating that the command that you have entered is incomplete, recall the command, add a space, and enter the **?** command to display a list of arguments that are available for the command. In this example, enter the day, month, and year using the correct syntax. Then press **Enter** to execute the command.

Syntax checking uses the caret symbol (^) as an error-location indicator. It appears at the point in the command string where the user has entered an incorrect command, keyword, or argument. The error-location indicator and interactive help system provide a way to easily find and correct syntax errors. In the clock example, the caret symbol indicates that the month was entered incorrectly as a number. The parser is expecting the month to be spelled out.

Task 3: Manage Cisco IOS Configuration

Now you will go through the startup and running configurations on a Cisco IOS device. The lab is prepared with the devices that are represented in the topology, but for this session you will only be using SW2. In the end, you will erase the configuration on SW2. Do not worry, though. The lab system will return the configurations the next time the lab is initialized.

The prompt displays the hostname that is configured on the device. You will modify this component of the switch configuration as you experiment with the startup and running configurations.

Activity

Step 1 On SW2, enter the global configuration mode and change the hostname of the switch to "Temp" and return to privileged EXEC.

Immediately after you change the hostname setting on the switch, the system prompt reflects the new name.

```
SW2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW2(config)# hostname Temp
Temp(config)# end
Temp#
```

You just modified the running configuration on the switch. The startup configuration has not changed.

Step 2 Display the running configuration that is parsed through the include filter, showing only the lines that include the string "hostname".

The use of tab completion and the question mark is intended to remind you that these options are always available to you. They will not be demonstrated any further in this session, but feel free to take advantage of them at any time.

```
Temp# sh<tab>
Temp# show r?
radius     region     registry        reload
resource   rhosts     rib             rif
route-map  route-tag  running-config

Temp# show run<tab>
Temp# show running-config | inc<tab>
Temp# show running-config | include hostname
hostname Temp
```

Tab completion was available for **show** and **running-config** and **include**, but not for **hostname**, because hostname is a freeform variable. It can be any string. There is no way for the Cisco IOS parser to guess what you want that string to be.

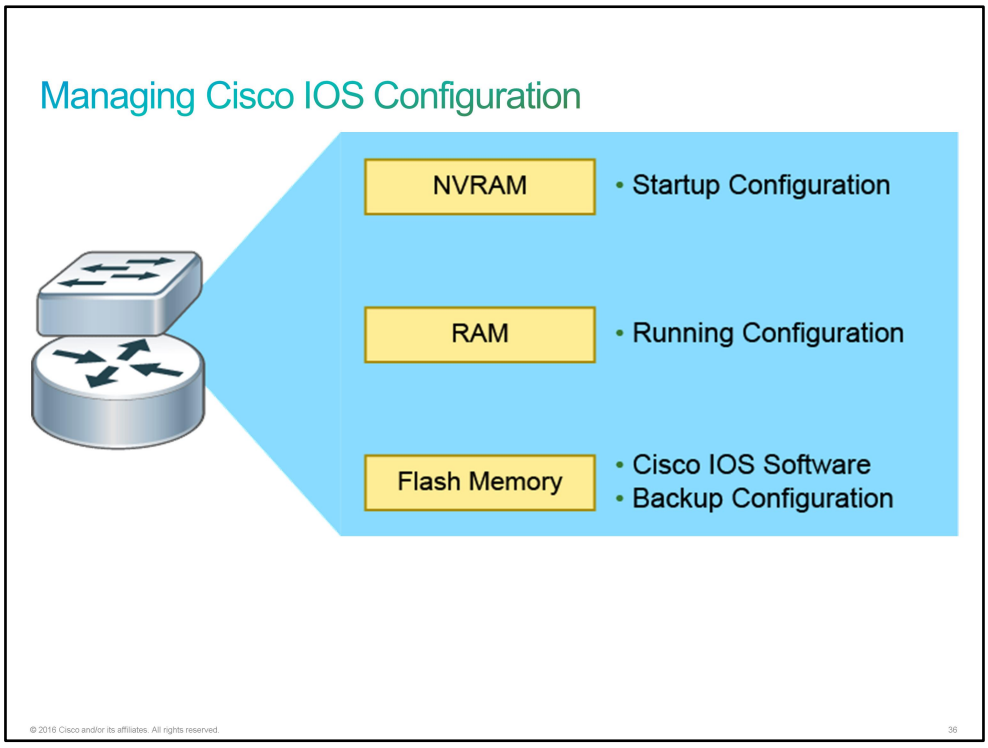
The one line in the configuration that includes the string hostname is the **hostname** command setting the hostname to "Temp."

Managing Cisco IOS Configuration

When a switch or a router starts, it looks for a configuration file in the NVRAM of the device. NVRAM is the memory in the device that retains information even when the device is powered down. The configuration file that is stored in NVRAM is called the startup-config file. If there is no startup-config file in NVRAM, the router or switch enters the setup utility and loads a blank configuration. The setup utility prompts you at the console for specific configuration information to create a basic initial configuration on the router or switch. You can also interrupt the setup utility and start configuring the device manually.

Once the device has started, the system copies the startup configuration to RAM. The configuration file in RAM is called the running-config file. As you make additional configurations, the system stores them in the running configuration. It is important to understand that RAM does not retain its information when the device is powered off or rebooted. If a change is made to the running configuration, it must be copied to the startup configuration, which is stored in the NVRAM, for it to be retained after a reboot.

In addition to NVRAM and RAM, Cisco devices have a third type of memory, called *flash memory*. Flash memory is similar to a hard drive in that the information that the system stores there is retained even when the device is powered off. Cisco IOS Software is stored in flash memory. Flash memory may also store backup configuration files and additional device-supported files.

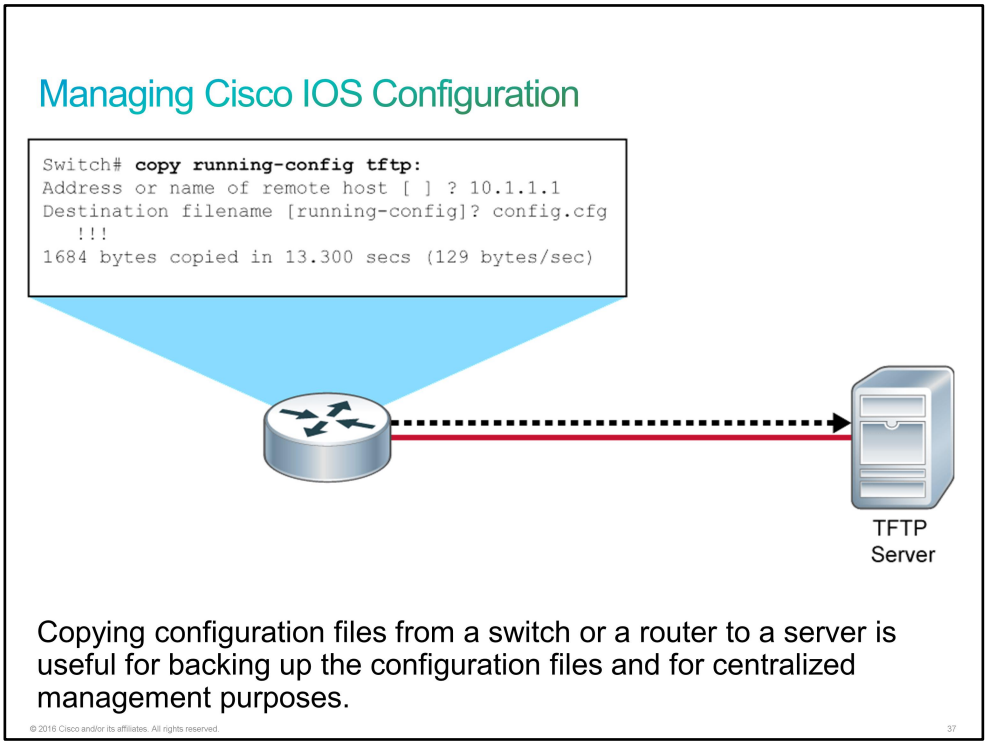


To view the configuration files, use the **show** command followed by the name of the file. For example, if you want to view the configuration that is stored in RAM, type **show running-config**. To save the running configuration, copy it to NVRAM. To do so, use the **copy** command followed by the names of the source and destination files. The complete command is **copy running-config startup-config**. Review the table for additional commonly used Cisco IOS commands.

Common IOS Management

Command	Function
show running-config	Displays the current running configuration. You can also use filters. For example, you can use the **show running-config interface GigabitEthernet0/1** command to display only the interface GigabitEthernet0/1 running configuration.
show startup-config	Displays the saved configuration in NVRAM.
configure terminal	Enters the configuration mode, where you can interactively create configurations in RAM from the console or remote terminal.
copy running-config startup-config	Saves the running configuration to NVRAM.
copy startup-config running-config	Startup configuration in NVRAM is merged into running configuration.
erase startup-config	Deletes the saved startup-config file in NVRAM.

You can also use the **copy** command to copy configuration files and Cisco IOS Software files from a switch or a router to a server (or vice versa) using FTP, SCP, HTTP, TFTP, and other protocols. For example, in the **copy running-config tftp:** command, the system copies the running configuration in RAM to a TFTP server. You must supply the IP address or name of the TFTP server and a destination filename. During the copying process, a series of exclamation marks show the progress of the upload.



Note	Regardless of the size of the network, there should always be a copy of the current running configuration online as a backup.

Copying configuration files from an external server to the running configuration in RAM or to the startup configuration file in NVRAM of the router or switch is useful for restoring backups. You should copy the files to a device other than the one that they were created on.

When you copy a configuration into RAM from any source, the configuration merges with the existing configuration in RAM. New configuration parameters are added, and changes to existing parameters overwrite the old parameters. Configuration commands in RAM for which there is no corresponding command in NVRAM remain unaffected.

Step 3 Now display the startup configuration that is parsed through the include filter, showing only the lines that include the string "hostname".

When you make changes to the running configuration, it does not affect the startup configuration. The startup configuration still has SW2 configured as the hostname.

```
Temp# show startup-config | include hostname
hostname SW2
```

Step 4 Use the **reload** command which will reboot the switch. This action will cause the switch to throw away the running configuration and read the startup configuration from scratch.

Answer **no** to the query about saving the modified configuration. The goal is to demonstrate how to return to the old configuration. If you save the modified configuration, the system will overwrite the old configuration.

After the reload, as indicated by the system prompt, the hostname has returned to SW2.

```
Temp# reload

System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm] <Enter>
<... output omitted ...>
Press RETURN to get started! <Enter>
<... output omitted ...>
SW2>
```

To make changes to the running configuration permanent, you have to save the running configuration over the startup configuration.

Step 5 Change the hostname one more time.

This time, set the hostname to "ThisWillStick."

```
SW2> enable
SW2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SW2(config)# hostname ThisWillStick
ThisWillStick(config)# end
ThisWillStick#
```

Step 6 Copy the running configuration over the startup configuration.

```
ThisWillStick# copy running-config startup-config
Destination filename [startup-config]? <Enter>
Building configuration...
Compressed configuration from 936 bytes to 641 bytes[OK]
ThisWillStick#
```

After this copy operation, the change to the hostname is reflected in the startup configuration and will now be able to survive a reload event.

Note	Optionally, you can use the **show startup-configuration** command to verify that the change is reflected there.

Step 7 Use the **reload** command again, and verify that the new hostname setting is still in place after the reboot event.

```
ThisWillStick# reload
Proceed with reload? [confirm] <Enter>
<... output omitted ...>
Press RETURN to get started! <Enter>
<... output omitted ...>
ThisWillStick>
```

The hostname does, indeed, remain as ThisWillStick.

Step 8 Now erase the startup configuration with the **erase startup-config** command.

```
ThisWillStick> enable
ThisWillStick# erase startup-config
Erasing the nvram filesystem will remove all configuration files! Continue?
[confirm]  <Enter>
[OK]
Erase of nvram: complete
*Jul  6 08:40:12.990: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
ThisWillStick#
```

Note	Even though the system erased the startup configuration, this action does not have an effect on the running configuration. In fact, (do not do this now) you could use the **copy running-config startup-config** command now to put the startup configuration back to the way it was.

Step 9 Verify that the system actually erased the startup configuration using **show startup-config**.

```
ThisWillStick# show startup-config
startup-config is not present
```

Step 10 Reload the switch.

Understand that the switch will attempt to read the startup configuration and find it missing. This situation will essentially set the switch back to the factory default state. Do not worry, when the lab is reinitialized, the lab system will set all device configurations appropriately.

```
ThisWillStick# reload
Proceed with reload? [confirm] <Enter>
<... output omitted ...>
Press RETURN to get started!  <Enter>
Switch>
```

Step 11 Verify that the hostname setting in the running configuration is the default value (Switch).

```
Switch> enable
Switch# show running-config | include hostname
hostname Switch
Switch#
```

Task 4: Improve User Experience in CLI

In this session, you will be able to practice using terminal history. Recalling previous commands is useful simply to reduce typing. When you recall a command, you can simply press Enter to use the exact same command, or you can edit it to suit your new purpose. The lab is prepared with the devices that are represented in the topology, but for this session you will only be using R1.

The prompt displays the hostname that is configured on the device. You will modify this component of the router configuration as you experiment with the startup and running configurations.

Activity

Step 1 On R1, use the **enable** command to access the privileged EXEC.

```
R1> enable
R1#
```

Step 2 Enter the sequence of commands that are shown below.

The sequence is rather arbitrary. The selection criteria were to include three EXEC commands and two configuration mode commands. Do not be concerned if the commands are new to you. This part will simply give you a little bit of data in the terminal history.

- **show ip route** (in privileged EXEC mode)
- **show clock** (in privileged EXEC mode)
- **show ip interface brief** (in privileged EXEC mode)
- **configure terminal** (to go in global configuration mode)
- **clock timezone EST 0** (in global configuration mode)
- **no ip domain-lookup** (in global configuration mode)

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
C        10.10.3.0/24 is directly connected, Loopback0
L        10.10.3.1/32 is directly connected, Loopback0


R1# show clock
*00:47:02.857 PST Mon Jul 6 2015


R1# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES NVRAM  up                    up
Ethernet0/1                unassigned      YES NVRAM  administratively down
down
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                unassigned      YES NVRAM  administratively down
down
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  unassigned      YES NVRAM  administratively down
down
Serial1/2                  unassigned      YES NVRAM  administratively down
down
Serial1/3                  unassigned      YES NVRAM  administratively down
down
Loopback0                  10.10.3.1       YES NVRAM  up                    up


R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# clock timezone EST 0
*Jul  6 08:48:41.931: %SYS-6-CLOCKUPDATE: System clock has been updated from
00:48:41 PST Mon Jul 6 2015 to 08:48:41 EST Mon Jul 6 2015, configured from
console by console.
R1(config)# no ip domain-lookup
```

Improving User Experience in CLI

The Cisco IOS CLI includes many features that make the configuration process easier and faster. These features include command-line editing keys, command history, and filtering parameters.

Command-Line Editing Keys

Command-line editing keys are shortcuts and hot keys that the CLI provides. Use these shortcuts and hot keys to move the cursor around on the command line for corrections or changes. Use them also to make configuring, monitoring, and troubleshooting easier. The table describes each of the shortcuts for command-line editing and controlling command entry.

Command-Line Editing Key Sequence	Description
Ctrl-A	Moves the cursor to the beginning of the command line
Ctrl-C	Aborts the current command and exits the configuration mode
Ctrl-E	Moves the cursor to the end of the command line
Esc-B	Moves the cursor back one word
Esc-F	Moves the cursor forward one word
Ctrl-B	Moves the cursor back one character
Ctrl-F	Moves the cursor forward one character
Ctrl-D	Deletes a single character at the cursor
Backspace	Removes one character to the left of the cursor
Ctrl-P	Redisplays the current command line
Ctrl-U	Erases a line
Ctrl-W	Erases a word to the left of the cursor
Ctrl-Z	Ends configuration mode and returns to the EXEC prompt
Tab	Completes a partially entered command if enough characters have been entered to make it unambiguous
Ctrl-Shift-6	Allows the user to interrupt a Cisco IOS process such as ping or traceroute
Ctrl-P or Up Arrow	Recalls last (previous) commands
Ctrl-N or Down Arrow	Recalls more recent commands

Note The Esc key is not functional on all terminals.

Command History

The Cisco CLI provides a history or record of commands that users have entered. You will find this feature, which is called the command history, particularly useful in helping you to recall long or complex commands or entries.

With the command history feature, you can complete the following tasks:

- **Display the contents of the command buffer:** By default, command history is enabled, and the system records the last 10 command lines in its history buffer.
- **Set the command history buffer size:** To change the number of command lines that the system will record during the current terminal session only, use the **terminal history** command in user EXEC mode.
- **Recall previously entered commands that are stored in the history buffer:** There is a buffer for EXEC mode and another buffer for the configuration mode. 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 (after recalling older commands with Ctrl-P or the Up Arrow key), press **Ctrl-N**, or the **Down Arrow** key. Repeat the key sequence to recall successively more recent commands.

  On most computers, there are additional select and copy functions available. Copy a previous command string, then paste or insert it as the current command entry, and press **Enter**.

  When you use **show** commands such as **show running-config**, Cisco IOS Software automatically pauses when displaying the output after a specified number of lines. The process of displaying the output pauses, and Cisco IOS Software displays "--More--." It then waits for user input to continue with the display process. You can press the Spacebar key to display another set of subsequent lines or press Enter to display a single line.
- **Set the number of lines on the current terminal screen:** You can use the **terminal length** command, followed by a number, to control the number of lines that the CLI displays without pausing during the output. A value of zero prevents the router from pausing between screens of output. By default, the value is set to 24.

Step 3 Now, while remaining in the configuration mode, use the **Up Arrow** and **Down Arrow** keys to scroll through the terminal history buffer.

Note that you do not see the EXEC commands. There is a separate terminal history buffer for configuration and EXEC modes.

Step 4 Leave the configuration mode (use **end, exit,** or press **Ctrl-Z**) to return to privileged EXEC.

Step 5 Again use the **Up Arrow** and **Down Arrow** keys to show that you can recall previous commands.

Step 6 Recall the **show ip route** command and then press the **Enter** key to resubmit it without any edits.

It is a common exercise to revisit show commands that display operational status as you make changes to the configurations on IOS devices and their neighbors.

Step 7 Now, type the following command, purposely mistyping "show" as "snow."


```
R1# snow ip interface brief
      ^
% Invalid input detected at '^' marker.
```

Everyone makes typographical errors. Dealing with them is one of the best uses of the terminal history and the command line editing tools.

Step 8 Follow this sequence to quickly and easily correct the typographical error and resubmit the corrected command:

a. Press the **Up Arrow** key once to retrieve the previous command.

b. Press **Ctrl-A** to move the cursor to the beginning of the line.

c. Press the **Right Arrow** twice to move the cursor to the right of the incorrect letter "n."

d. Press **Backspace** to erase the letter "n."

e. Press **h** to insert the correct letter "h."

f. Press **Enter** to resubmit the corrected command.

Step 9 Return to the global configuration mode.

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#
```

Note	As before, you will be using commands that you are not familiar with to facilitate the demonstration of the power of the terminal history buffer. Do not concern yourself with commands themselves. Instead, focus on how beneficial the terminal history buffer can be.

Step 10 Configure the description of interface Serial 1/0 and enable the interface by overriding the default **shutdown** command.

```
R1(config)# interface Serial 1/0
R1(config-if)# description Link to SP1
R1(config-if)# no shutdown
*Jul  6 08:51:13.776: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
*Jul  6 08:51:14.780: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/0, changed state to up
R1(config-if)#
```

Step 11 Repeat a very similar configuration for interface Serial 1/1.

The following process can make this task relatively easy:

a. Press the **Up Arrow** key three times to recall the interface command. Edit the 1/0 to be 1/1 and press the **Enter** key to resubmit the edited command.

b. Press the **Up Arrow** key three times, to recall the description command, edit the SP1 to be SP2 and press the **Enter** key to resubmit the edited command.

c. Press the **Up Arrow** key three times, to recall the **no shutdown** command, and press the **Enter** key to resubmit the command without any editing.

The resulting sequence should look like the following example:

```
R1(config)# interface Serial 1/1
R1(config-if)# description Link to SP2
R1(config-if)# no shutdown
*Jul  6 09:02:22.638: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Jul  6 09:02:23.642: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
R1(config-if)#
```

Step 12 Leave the configuration mode by using **end**, **exit** (2 times), or pressing **Ctrl-Z** to return to privileged EXEC.

Optionally, you can save the running configuration to the startup configuration, but it is not necessary in the automated lab environment.

Filtering Parameters

Another useful feature that improves the user experience in the CLI is the filtering of **show** outputs. Using filtering, you can display only the parts of **show** outputs that you are interested in. You can filter outputs by typing the pipe (|) character after a **show** command, followed by a filtering parameter and a filtering expression. The table describes filtering parameters that are available for output filtering.

Parameter	Description
begin	Shows all output lines, starting with the line that matches the filtering expression
exclude	Excludes all output lines that match the filtering expression
include	Includes all output lines that match the filtering expression
section	Shows the entire section that starts with the filtering expression

Step 13 On the R1 router use **begin** and **include** options with **show running-config** command and filtering expression *interface*.

You should see following output when using **begin** option:


```
R1# show running-config | begin interface
interface Loopback0
 ip address 10.10.3.1 255.255.255.0
!
interface Ethernet0/0
 description Link to SW2
 ip address 10.10.1.1 255.255.255.0
!
interface Ethernet0/1
 no ip address
 shutdown
!
interface Ethernet0/2
 no ip address
 shutdown
!
interface Ethernet0/3
 no ip address
 shutdown
!
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
!
!
!
control-plane
!
!
!
!
!
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
 transport input all
!
!
end
```

You should see following output when using **include** option:

```
R1# show running-config | include interface
interface Loopback0
interface Ethernet0/0
interface Ethernet0/1
interface Ethernet0/2
interface Ethernet0/3
interface Serial1/0
interface Serial1/1
interface Serial1/2
interface Serial1/3
```

Step 14 On the R1 router use **section** option with **show running-config** command and filtering expression *interface*.

You should see following output when using **section** option:

```
R1# show running-config | section interface
interface Loopback0
 ip address 10.10.3.1 255.255.255.0
interface Ethernet0/0
 description Link to SW2
 ip address 10.10.1.1 255.255.255.0
interface Ethernet0/1
 no ip address
 shutdown
interface Ethernet0/2
 no ip address
 shutdown
interface Ethernet0/3
 no ip address
 shutdown
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
```

Step 15 On the R1 router use **exclude** option with **show running-config** command and filtering expression *!*.

You should see following output when using **exclude** option:


```
R1# show running-config | exclude !
Building configuration...

Current configuration : 1223 bytes
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname R1
boot-start-marker
boot-end-marker
no aaa new-model
clock timezone PST -8 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180


no ip domain lookup
ip cef
no ipv6 cef
multilink bundle-name authenticated
redundancy
interface Loopback0
 ip address 10.10.3.1 255.255.255.0
interface Ethernet0/0
 description Link to SW2
 ip address 10.10.1.1 255.255.255.0
interface Ethernet0/1
 no ip address
 shutdown
interface Ethernet0/2
 no ip address
 shutdown
interface Ethernet0/3
 no ip address
 shutdown
interface Serial1/0
 no ip address
 shutdown
 serial restart-delay 0
interface Serial1/1
 no ip address
 shutdown
 serial restart-delay 0
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
ip forward-protocol nd
no ip http server
no ip http secure-server
control-plane
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
 transport input all
end
```

Challenge

1. Which network service is not delivered by Cisco IOS Software?

   A. features to carry the chosen network protocols and functions
   B. connectivity for high-speed traffic between devices
   C. security to control access and prohibit unauthorized network use
   D. scalability to add interfaces and capability as needed for network growth
   E. Microsoft operating system that is called MS-DOS

2. How can you generally access Cisco IOS Software services?

   A. using a CLI
   B. using an MS-DOS
   C. using an EXEC session
   D. using a GNOME

3. In the Cisco IOS Software what are the two EXEC access levels? (Choose two.)

   A. admin EXEC
   B. user EXEC
   C. privileged EXEC
   D. basic EXEC
   E. advanced EXEC

4. Which Cisco IOS Software EXEC levels allow a person to access only a limited number of basic monitoring commands?

   A. admin EXEC
   B. user EXEC
   C. privileged EXEC
   D. basic EXEC
   E. advanced EXEC

5. Which Cisco IOS command do you use to change from user EXEC level into privileged EXEC level?

   A. **enable**
   B. **disable**
   C. **admin**
   D. **configure**
   E. **configure terminal**

6. Which Cisco IOS command do you use to change from privileged EXEC level into user EXEC level?

   A. **enable**
   B. **disable**
   C. question mark (**?**)
   D. **exit**

7. Which Cisco IOS command do you use to display configuration in RAM?

   A. **show startup-config**
   B. **show config**
   C. **show ram-config**
   D. **show running-config**

Answer Key

Challenge

1. E
2. A
3. B, C
4. B
5. A
6. B
7. D

Lesson 5: Starting a Switch

Introduction

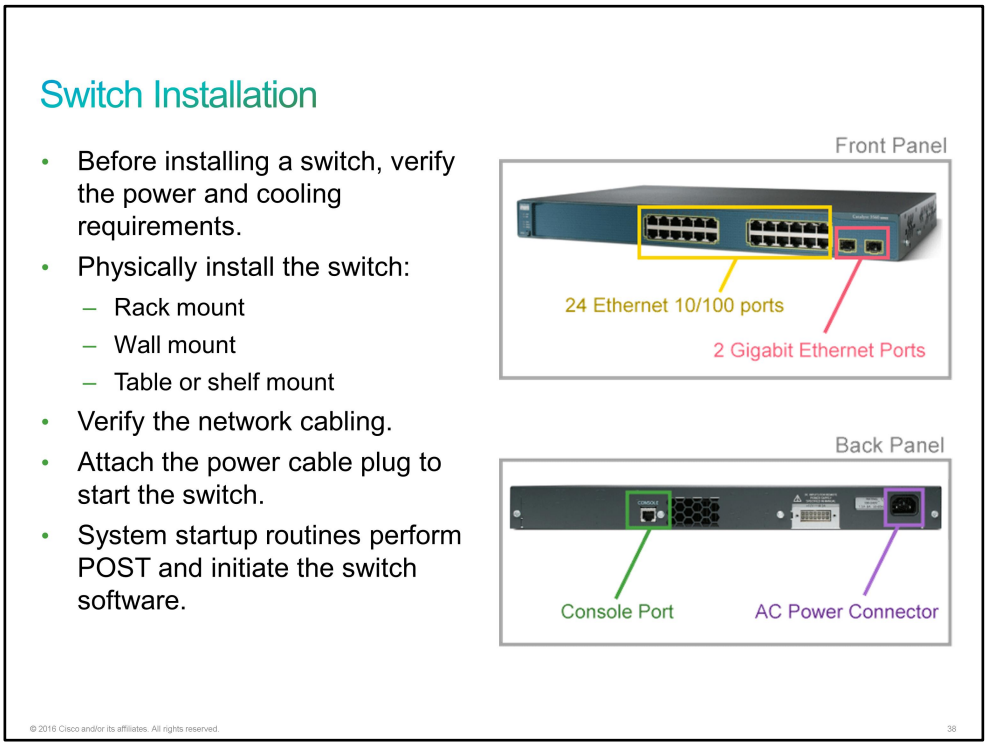
Congratulations, you got the job. You now have the opportunity to perform your first installation. A law firm contracted CCS to install a small network to share printers and other resources, using a switch. Bob tells you that, in addition to performing the physical installation, you will need to be able to specify the hostname, enable the interface, assign a host IP address, and configure the default gateway and interface descriptions.

Bob asks if you would like to go on site to set up the switch or review one or more topics before performing the tasks.

You might want to review general requirements for physical switch installation. You should also know how to read switch LED indicators to recognize the status of a switch and how to access a switch CLI. Get familiar with accessing the switch CLI and configuration commands. You should also not forget to review the show commands, which enable you to verify the status of the switch.

Switch Installation

Before you physically install a Catalyst switch, you must have the correct power and operating environment. When you have correctly connected cable, you can power up the switch.



Physical installation and startup of a Catalyst switch requires completion of these steps:

1. Before performing physical installation, verify the following:
   - Switch power requirements
   - Switch operating environment requirements (operational temperature and humidity)
2. Use the appropriate installation procedures for rack mounting, wall mounting, or table or shelf mounting.
3. Before starting the switch, verify that network cable connections are secure.
4. Attach the power cable plug to the power supply socket of the switch. The switch will start. Some Catalyst switches do not have power buttons.
5. Observe the boot sequence:
   - When the switch is on, POST begins. During POST, the switch LED indicators blink while a series of tests determine that the switch is functioning properly.
   - The Cisco IOS Software output text is displayed on the console.

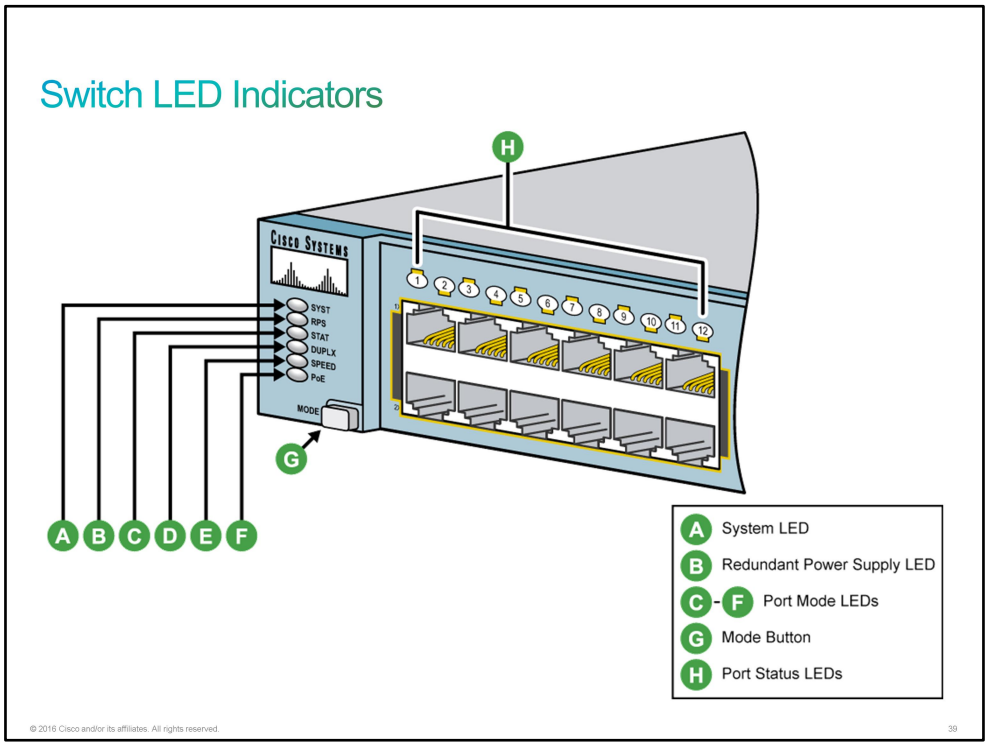
When all startup procedures are finished, the switch is ready to configure.

Switch LED Indicators

Typically, before turning on a device, you need to plug it in. However, some Cisco switches do not have power switches, so when you plug them in, they power up automatically. Because of this fact, you should make sure that the console cable is connected and the terminal program running before you plug in the switch the first time. This preparation will allow you to monitor the boot process of the switch. As the switch powers on, it begins POST, a series of tests that run automatically to ensure that the switch functions properly. Ensuring the switch passes POST is the first step of deploying a switch.

When you need to examine how a switch is working, or to verify its status, and to troubleshoot any problems, you usually mostly use commands from the Cisco IOS CLI. However, the switch hardware does include several LEDs that provide some status and troubleshooting information. Generally, when the Cisco switch is functioning normally, the LEDs are lit in green, and if there is a malfunction, the LEDs are lit in amber.

The figure here shows the front of a Cisco Catalyst 3750 switch, with six LEDs on the left, one LED over each port, and a mode button.



LED Status

Letter in Figure	Name	Description
A	SYST	Implies the overall system status.
B	RPS	Suggests the status of the extra (redundant) power supply.
C	STAT	If on (green), each port LED implies the status of that port.

Letter in Figure	Name	Description
D	DUPLX	If on (green), each port LED implies the duplex of that port (on is full duplex; off means half duplex).
E	SPEED	If on (green), each port LED implies the speed of that port, as follows: off means 10 Mbps, solid green means 100 Mbps, and flashing green means 1 Gbps.
F	PoE	Some switches have a PoE LED in the system status group of LEDs. This LED indicates the per-port and system PoE status.
G	MODE	A button that cycles the meaning of the LEDs through three states (STAT, DUPLX, SPEED).
H	Port	Has different meanings, depending on the port mode as toggled using the mode button.

To help make sense of the LEDs, consider the example of the SYST LED for a moment. This LED provides a quick overall status of the switch, with three simple states on most Cisco Catalyst 2960 switch models:

- Off: The switch is not powered on.
- On (green): The switch is powered on and operational. Cisco IOS Software has been loaded.
- On (amber): The switch POST process failed and the Cisco IOS Software did not load.

So, just looking at the SYST LED on the switch tells you whether the switch is working and, if it is not, whether this issue is due to the loss of power (the SYST LED is off) or some kind of POST problem (the LED is amber).

Click the **Play** Button to watch a short video about Cisco catalyst switch LED indicators.

Connecting to a Console Port

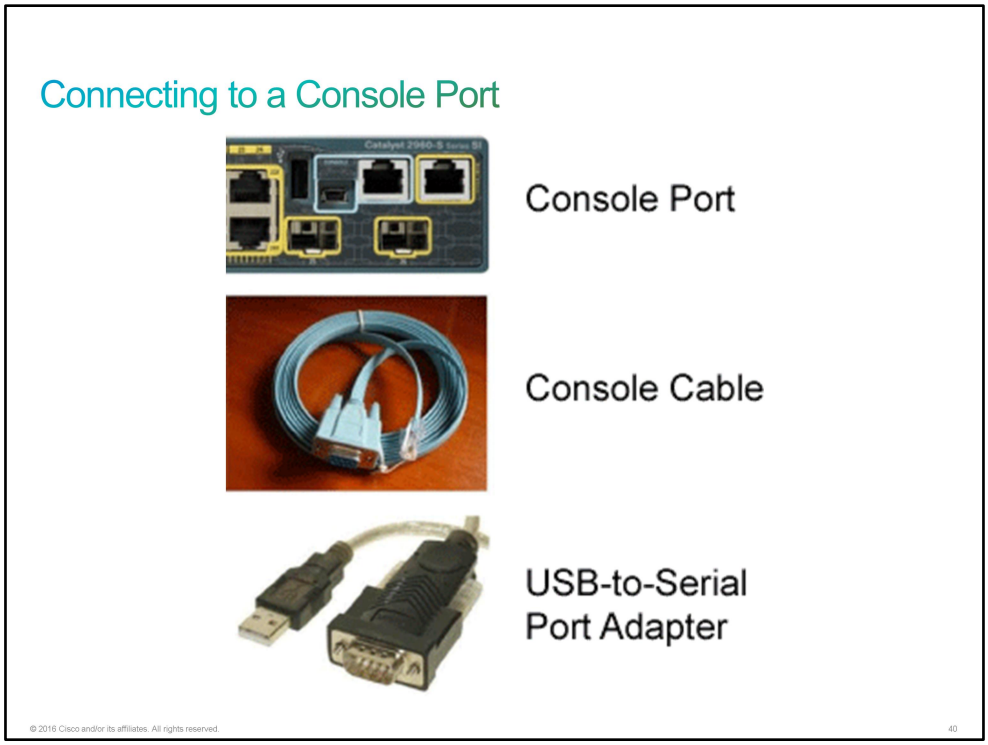
Unlike a computer host, Cisco switches do not have a keyboard, monitor, or mouse device to allow direct user interaction. Upon initial installation, you can configure the switch from a PC that is connected directly through the console port on the switch.

You need the following equipment to access a Cisco device through the console port:

- RJ-45-to-DB-9 console cable
- PC or equivalent with serial port and communications software, such as HyperTerminal, configured with these settings:
  - Speed: 9600 bps
  - Data bits: 8
  - Parity: None
  - Stop bit: 1
  - Flow control: None

Modern computers and notebooks rarely include built-in serial ports. You often use a USB-to-RS-232-compatible serial port adapter instead.

On newer Cisco network devices, a USB serial console connection is also supported. You need a suitable USB cable (USB Type A-to-5-pin mini Type B) and operating system device driver to establish connectivity.



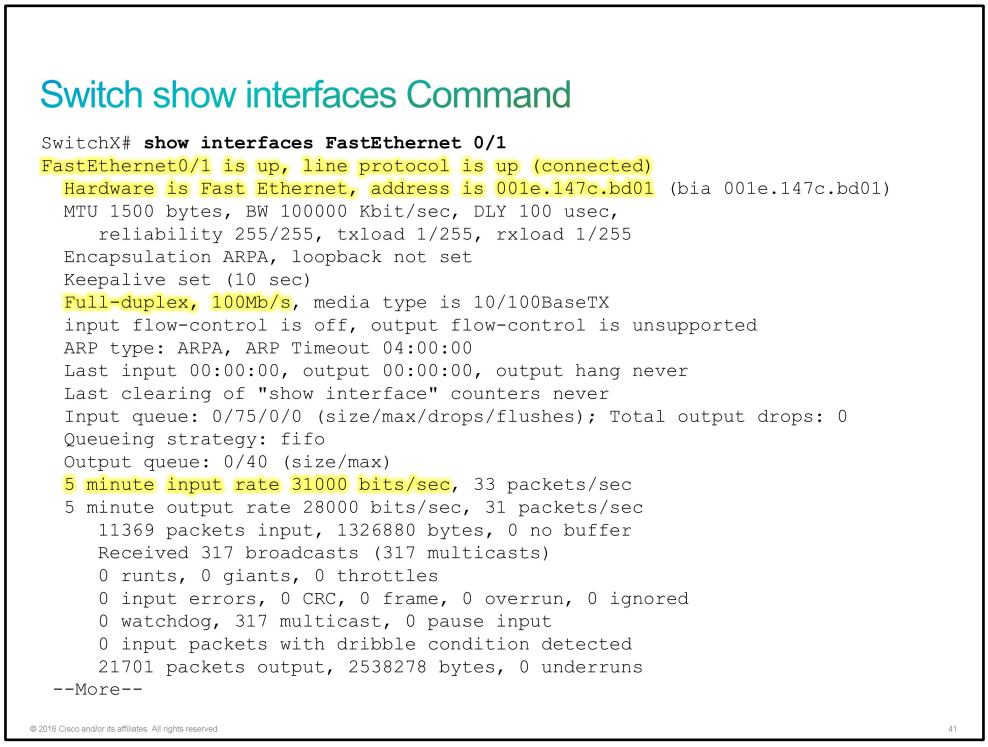
Note	Only one console port can be active at a time. When a cable is plugged into the USB console port, the RJ-45 port becomes inactive. When the USB cable is removed from the USB port, the RJ-45 port becomes active.

When a console connection is established, you gain access to user EXEC mode, by default. To start configuration, you must enter privileged EXEC mode by using the **enable** command.

Basic Show Commands and Information

Switch show version Command

- **show interfaces**: The **show interfaces** command displays the status and statistical information for the network interfaces of the switch. The resulting output varies, depending on the network for which a particular interface has been configured. You usually enter this command with the options *type* and *slot/number*. The *type* option allows values such as **FastEthernet** and **GigabitEthernet.** The *slot/number* option indicates slot 0 and the port number on the selected interface (for example, **fa0/1**).



The table shows some of the fields in the display that you will find useful for verifying fundamental switch details:

Fundamental Switch Details

Output	Description
FastEthernet0/1 is up, line protocol is up (connected)	Indicates the status of the interface hardware. In this example, it is functioning correctly. The hardware status is followed by the status of the line protocol, which in this example is also operational and active.
Hardware is Fast Ethernet, address is 001e.147c.bd01	Indicates the MAC address of the interface
Full-duplex, 100 Mb/s	Shows the type and mode of connection. Other possibilities include half-duplex, 10 Mb/s.

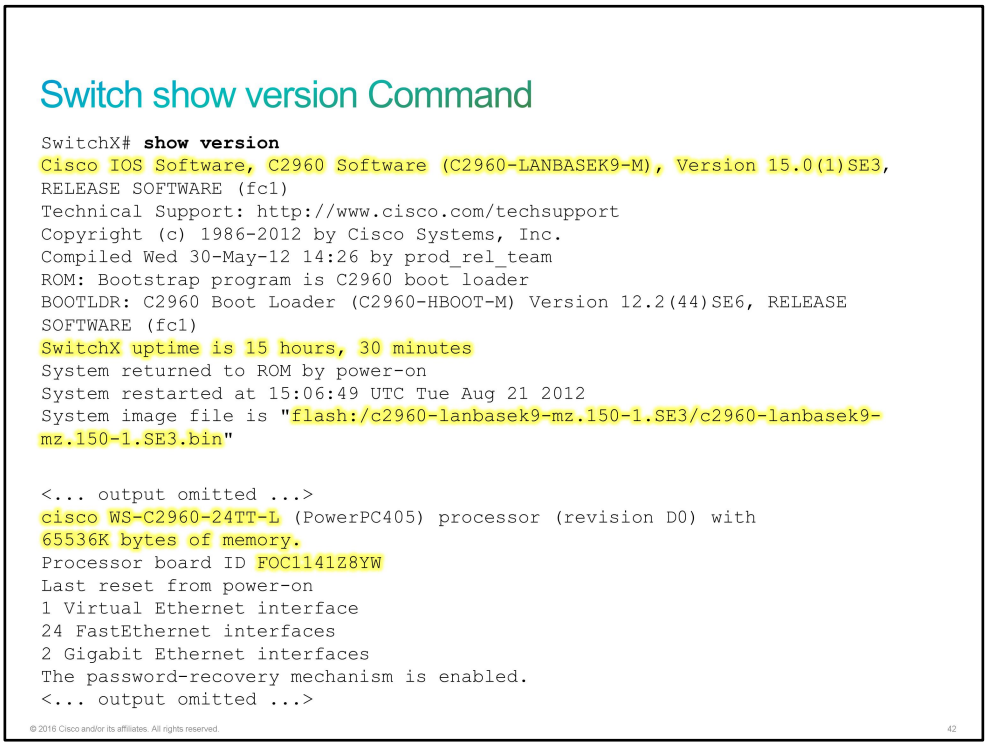
Output	Description
5 minute input rate 31000 bits/sec	Reports interface traffic statistics for average input rate

Note	The **show interfaces** command is used frequently when you are configuring and monitoring network devices.

Switch show interfaces Command

After you log into a Cisco switch, you can verify the switch software and hardware status by using several commands that you execute from privileged EXEC mode. These commands include the **show version**, **show interfaces**, and **show running-config** commands. Here is a look at each of these commands in more detail.

- **show version**: You can use the **show version** IOS command in privileged EXEC mode to verify the IOS version and release numbers of the IOS software that is running on a Cisco switch.

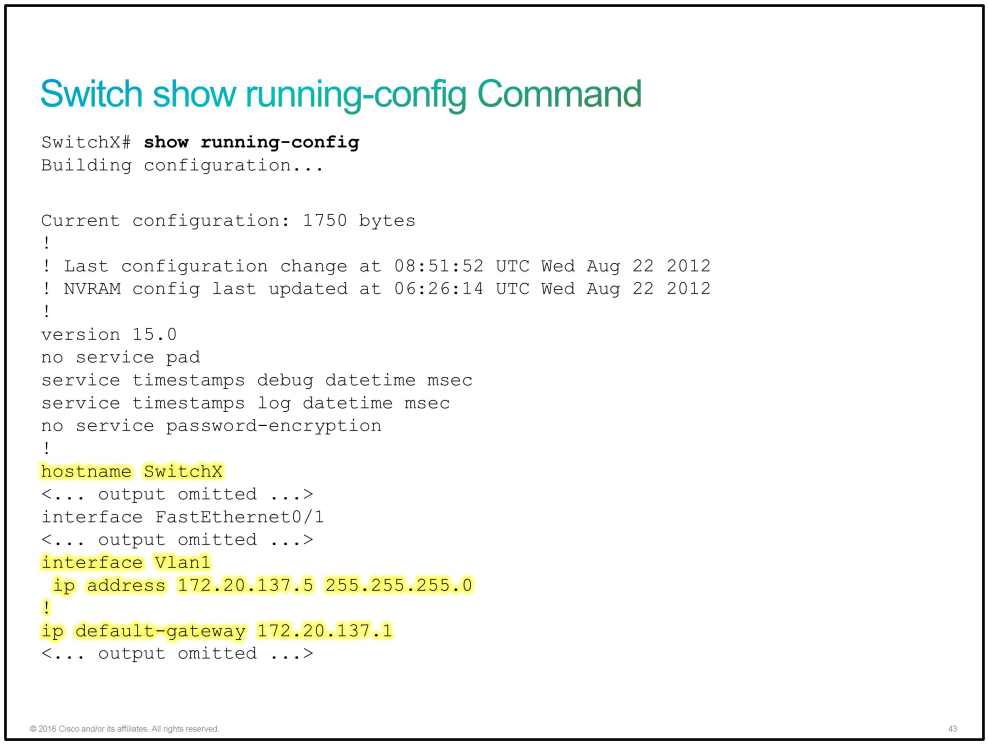


The following table describes some of the output fields of the **show version** command:

Output Fields from the show version Command

Output	Description
Cisco IOS Software version	Identification of the software by name and version number Always specify the complete version number when reporting a possible software problem. In this example, the switch is running Cisco IOS Release 15.0(1)SE3.
Switch uptime	Current days and time since the system was last booted In this example, the switch uptime is 15 hours and 30 minutes.
Switch platform	Hardware platform information, including revision and amount of RAM
Processor board ID	Device serial number

Switch show running-config Command



- **show running-config**: The **show running-config** command displays the current running (active) configuration file of the switch. This command requires privileged EXEC mode access. This command displays the IP address, subnet mask, and default gateway settings.

Discovery 2: Perform Basic Switch Configuration

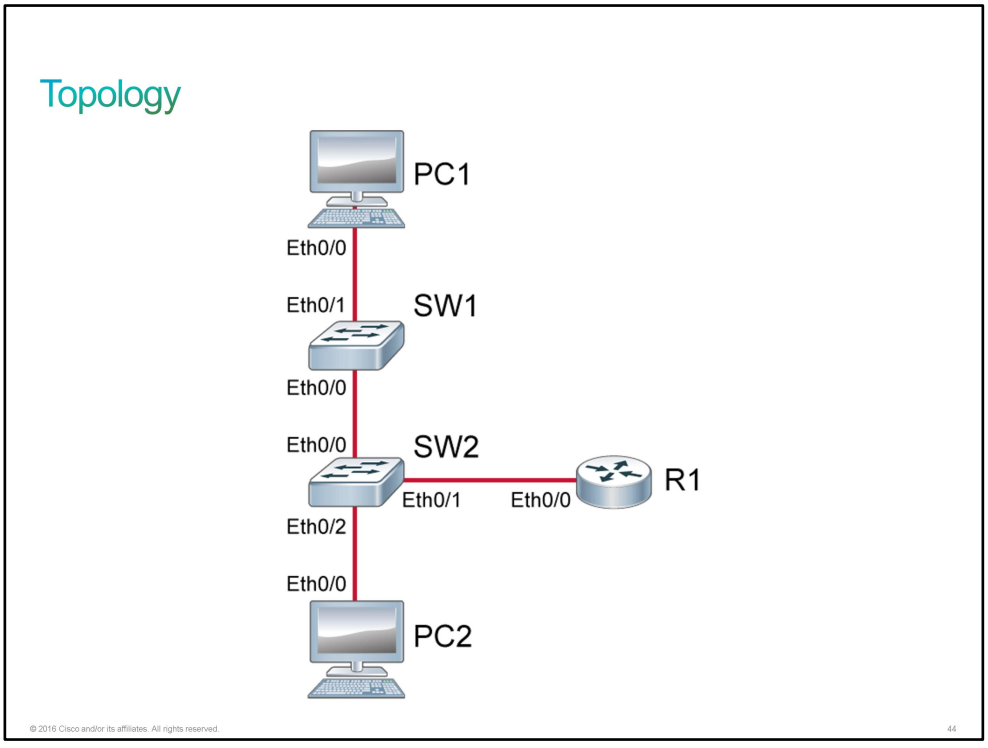
Introduction

This discovery session guides you through the initial configuration of a switch with Cisco IOS Software. The lab is prepared with the devices that are represented in the topology diagram, with the IP addresses as depicted in the table. Note that PC1, PC2, SW2, and R1 are fully configured. In this discovery session, your task will be to provide an initial configuration for SW1. During the session, you will configure and verify each of the following settings on SW1:

- Hostname
- IP address
- Default gateway
- Interface descriptions on the interfaces connecting to PC1 and SW2

You will also verify switch settings by using different **show** commands.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Configure a Switch from the Command Line

Activity

Step 1 Access the console of SW1 and use the **enable** command to access the privileged EXEC.

On SW1, enter the following command:

```
Switch> enable
Switch#
```

You can, of course, use unambiguous abbreviations for commands, such as **en**. You can also take advantage of tab completion using something like **en<tab>**.

The change of the last character in the prompt from > to # is an indication that you have successfully accessed privileged mode.

Step 2 Enter the global configuration mode using the configure terminal command.

On SW1, enter the following command:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#
```

The change in prompt to include (config) indicates that you are now in the global configuration mode

Step 3 Set the hostname of the switch to SW1 by using the **hostname** command.

On SW1, enter the following command:

```
Switch(config)# hostname SW1
SW1(config)#
```

The prompt reflects the hostname. It was the default (Switch) and is now set to SW1.

Step 4 In this topology, only VLAN 1 is in use. Set the IP address that SW1 uses on VLAN 1 to 10.10.1.2 with a 24-bit subnet mask. To do so, you will have to enter the interface configuration mode and use the **ip address** command. You will also need to enable the interface with the **no shutdown** command.

On SW1, enter the following command:

```
SW1(config)# interface vlan 1
SW1(config-if)# ip address 10.10.1.2 255.255.255.0
SW1(config-if)# no shutdown
```

Again, the prompt changes as you move through the hierarchy of CLI modes. The prompt includes *config*, indicating that you are in the global configuration mode.

Step 5 Next, set the SW1 default gateway to 10.10.1.1. You do this action from the global configuration mode. Use the **exit** command to return to the global configuration mode, then use the **ip default-gateway** command appropriately.

On SW1, enter the following command:

```
SW1(config-if)# exit
SW1(config)# ip default-gateway 10.10.1.1
```

Again, the prompt changes as you move through the hierarchy of CLI modes. The prompt includes *config*, indicating that you are in the global configuration mode.

Step 6 Finish the configuration requirements by setting the descriptions on interfaces Ethernet 0/0 and Ethernet 0/1, which are links to SW2 and PC1 respectively. The **description** command is available in the interface configuration mode.

On SW1, enter the following command:

```
SW1(config)# interface ethernet 0/0
SW1(config-if)# description Link to SW2
SW1(config-if)# interface ethernet 0/1
SW1(config-if)# description Link to PC1
```

Step 7 SW1 is now properly configured. Leave configuration mode and return to privileged EXEC mode.

On SW1, enter the following command:

```
SW1(config-if)# end
SW1#
```

You can accomplish the same thing in several ways within Cisco IOS. Instead of using the **end** command to go from interface configuration mode all the way back to privileged EXEC, you could also have used the **exit** command twice, or simply pressed **Ctrl-Z**.

Task 2: Verify the Switch Initial Startup Status

Activity

This discovery session assumes that you have just finished configuration of the following settings on SW1:

- Hostname
- IP address
- Default gateway
- Interface descriptions on the interfaces connecting to PC1 and SW2

You will now verify these settings on SW1. Consult the topology diagram and configuration specifications table for the complete connectivity and configuration details. Note that PC1, PC2, SW2, and R1 were already fully configured. In this discovery session, you will focus solely on SW1.

Step 1 On SW1, verify the correct IP address configuration on interface VLAN1. To verify proper IP configuration, you have several options. Normally, you have several ways to verify configuration elements. Often, these include directly viewing the configuration, showing operational status, and verifying behavior. You will utilize all three methodologies here.

On SW1, enter the following command:

One way that you can verify the configuration is by simply viewing it with the **show running-config** command. You can pare down the output of this command by piping it to the include or the section filter. But, since viewing the configuration of a particular interface is a common exercise, you can specify an interface directly to the **show running-config** command. Give the following a try:

```
SW1# show running-config interface vlan 1
Building configuration...

Current configuration : 59 bytes
!
interface Vlan1
 ip address 10.10.1.2 255.255.255.0
end
```

Another option that you have for verifying the IP address configuration is by viewing the status of interfaces. Use the **show ip interface brief** command to see the status of interface VLAN1. Verify that it is up and that the IP address is correct.

```
SW1# show ip interface brief
Interface              IP-Address      OK? Method Status
Protocol
<... output omitted ...>
Vlan1                  10.10.1.2       YES manual up                    up
```

You might find looking at the configuration or at system status useful, but often the most satisfying method is verifying system behavior. If the interface has been properly configured, you should be able to ping other IP addresses on the local subnet. Try to ping PC1, PC2, and R1.

```
SW1# ping 10.10.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.10, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/5 ms
SW1# ping 10.10.1.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.20, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
SW1# ping 10.10.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1009 ms
```

The **ping** command will send five ICMP echo requests and wait for a reply after each request. A period (**.**) indicates a timeout on the reply. An exclamation point (**!**) indicates that the reply was received.

It is common for a timeout to occur on the first echo request, but you may not see it. It happens when the local system does not have an entry in its ARP table for the remote system.

A ping can provide rudimentary performance indications. Timeouts are obviously bad, but the command also displays response time statistics for the replies that were received.

Step 2 Now, verify that the default gateway is configured appropriately. Again, you have multiple options.

On SW1, enter the following command:

View the running configuration, including only lines which include the string "default".

```
SW1# show running-config | include default
ip default-gateway 10.10.1.1
```

Besides looking at the configuration, you verify the status. Use the **show ip route** command to view the IP routing table of SW1.

```
SW1# show ip route
Default gateway is 10.10.1.1
Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

Again, you can also verify system behavior. If your default gateway is properly set, you should be able to ping IP addresses on remote subnets. Try to ping the address 10.10.3.1 which is on the other side of R1.

```
SW1# ping 10.10.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1012 ms
```

Step 3 The last thing to verify is the description on the appropriate interfaces. You should have configured descriptions on both Ethernet 0/0 and Ethernet 0/1. As usual, there are multiple strategies that you can use for verification.

On SW1, enter the following command:

View the running configuration, but use the section filter to view sections that include the "0/0" string. Repeat this action for the "0/1" string.

```
SW1# show running-config | section 0/0
interface Ethernet0/0
 description Link to SW2
 duplex auto
SW1# show running-config | section 0/1
interface Ethernet0/1
 description Link to PC1
 duplex auto
```

Verify the system status using the **show interface status** command.

```
SW1# show interface status
Port        Name          Status       Vlan       Duplex  Speed Type
Et0/0       Link to SW2   connected    trunk        auto   auto unknown
Et0/1       Link to PC1   connected    1            auto   auto unknown
<... output omitted ...>
```

Challenge

1. Which of the following happens first when a switch is powered on?

   A. IOS is booted up
   B. Linux is booted up
   C. POST begins
   D. Command Line appears

2. Some Catalyst switches do not have power ON buttons. True or False?

   A. True
   B. False

3. On some simple Cisco Catalyst Switches, the SYST LED will blink which color when the switch POST process fails and the Cisco IOS Software does not load?

   A. RED
   B. AMBER
   C. GREEN
   D. WHITE

4. Which of the following will help connect you to a Cisco device through the console port?

   A. Console Cable
   B. Serial Cable
   C. USB Cable
   D. RJ45 Cable

5. Which of the following would you use on the PC to connect through the console cable to the Cisco Device?

   A. Command Line
   B. Terminal
   C. HyperTerminal
   D. Web based Application

6. Each port on a Cisco Catalyst Switch has two LEDs to show its status details. True or False?

   A. True
   B. False

7. The console port on a Cisco Device looks like which of the following:

   A. RJ45 port
   B. Serial DB-9 Port
   C. USB port

Answer Key

Challenge

1. C
2. A
3. B
4. A
5. C
6. B
7. A

Lesson 6: Understanding Ethernet and Switch Operation

Introduction

CCS has received an email from a lawyer at the XYZ law firm, for which you recently implemented a small network. The lawyer, Li, explains that XYZ has decided to contract with an IT services company for all its networking needs. As the most computer-savvy employee of the firm, Li has taken on the responsibility of selecting the IT services company and then serving as a liaison between it and the law firm. Because he enjoyed working with you during the network implementation, he wants to give CCS the opportunity to win the contract.

Li is a highly curious individual who takes every opportunity to learn about networking, so in addition to requesting that CCS contact him about providing IT services, he lists several topics that he would like to discuss:

- Ethernet LAN connection media
- Ethernet frame structure
- Ethernet addresses
- Switch operation
- Duplex communication

Hoping to get the contract, Bob asks you to go to the law firm and speak to Li in person.

Before talking to Li, you should feel confident discussing different Ethernet media options, including the most common connectors and cable types. You might want to refresh your knowledge of the Ethernet frame structure and also MAC addresses and their function.

Make sure that you are familiar with switch operation, duplex options, and collision domains.

Ethernet LAN Connection Media

To connect a switch to a LAN, you must use some sort of media. The most common LAN media is Ethernet. Ethernet is not just a type of cable or protocol. It is a network standard that the IEEE published. So you can hear various Ethernet terms, such as Ethernet protocols, Ethernet cables, Ethernet ports, and Ethernet switches. Ethernet is basically a set of guidelines that enable various network components to work together. These guidelines specify cabling and signaling at the physical and data link layers of the OSI model. For example, Ethernet standards recommend different types of cable and specify maximum segment lengths for each type.

The names of the standards (shown in the top row of the table) specify the transmission speed, the type of signaling, and the type of cabling. For example, in the standard name 10BASE-T, the "10" specifies a transmission speed of 10 Mbps, the word "base" refers to baseband signaling (which means that only Ethernet signals are carried on the medium), and the letter "T" represents twisted-pair cabling. Twisted-pair cabling is a type of wiring in which two conductors are twisted together for the purposes of canceling EMI from external sources.

Ethernet Media Standards

Requirement	100BASE-TX	100BASE-FX	1000BASE-T	1000BASE-SX	1000BASE-LX
Media	TIA Category 5 UTP two-pair	62.5/125 micron multimode fiber	TIA Category 5, 5e UTP four-pair	62.5/50 micron multimode fiber	9 micron single-mode fiber
Maximum Segment Length	100 m (328 ft)	400 m (1312.3 ft)	100 m (328 ft)	275 m (62.5 micron) 550 m (50 micron)	5–10 km (1.86–6.2 miles)

Requirement	100BASE-TX	100BASE-FX	1000BASE-T	1000BASE-SX	1000BASE-LX
Connector	ISO 8877 (RJ-45)	Duplex MIC ST	ISO 8877 (RJ-45)	—	—

Ethernet LAN Connection Media

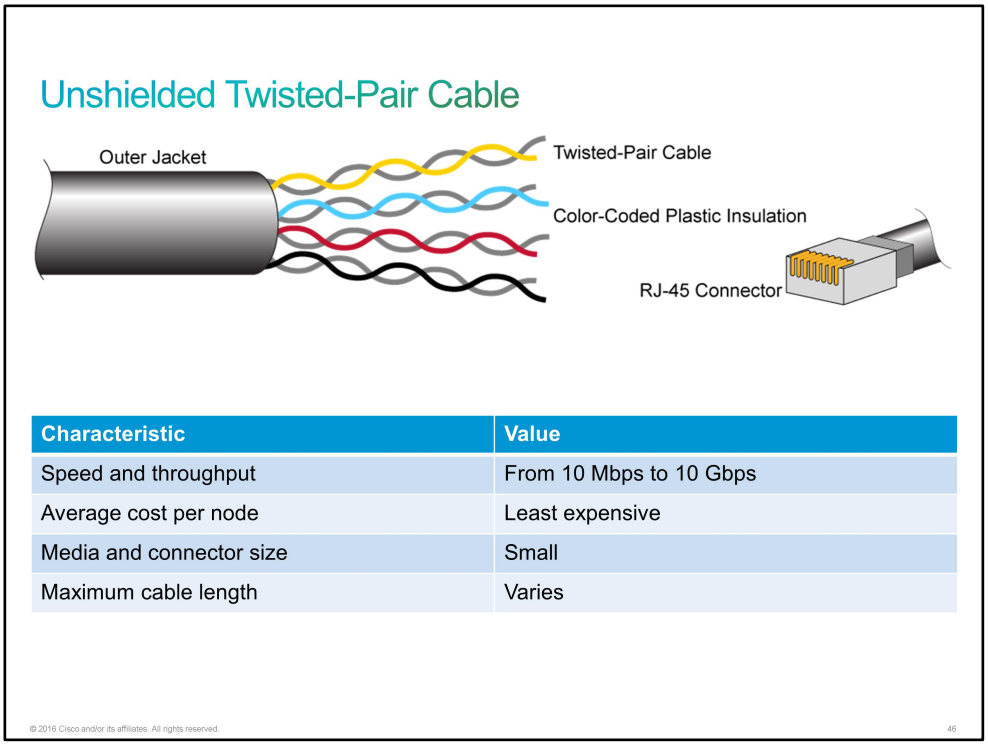
- The mechanical properties of Ethernet depend on the type of physical medium:
  - Coaxial (not used anymore)
  - Twisted copper pair
  - Fiber-optic
  - Wireless
- Although wireless is increasing in popularity for desktop connectivity, copper and fiber are the most popular physical layer media for network deployments.
- Ethernet was originally based on the concept of computers communicating over a shared coaxial cable.

Coaxial Cable

Copper Media

Take a look at copper media. Most Ethernet networks use UTP copper cabling for short and medium-length distances because of its low cost, when compared to fiber-optic or coaxial cable.

Unshielded Twisted-Pair Cable



Ethernet over twisted-pair technologies use twisted-pair cables for the physical layer of an Ethernet computer network. Twisted-pair cabling is a type of wiring in which two conductors (the forward and return conductors of a single circuit) are twisted together for the purposes of canceling EMI from external sources (for example, electromagnetic radiation from UTP cables and crosstalk between neighboring pairs).

A UTP cable is a four-pair wire. Each of the eight individual copper wires in a UTP cable is covered by an insulating material. In addition, the wires in each pair are twisted around each other. The advantage of a UTP cable is its ability to cancel interference, because the twisted-wire pairs limit signal degradation from EMI and RFI. To further reduce crosstalk between the pairs in a UTP cable, the number of twists in the wire pairs varies. Cables must follow precise specifications regarding how many twists or braids are permitted per meter.

A UTP cable is used in various types of networks. When used as a networking medium, a UTP cable has four pairs of either 22- or 24-gauge copper wire. A UTP that is used as a networking medium has an impedance of 100 ohms, differentiating it from other types of twisted-pair wiring such as that used for telephone wiring. A UTP cable has an external diameter of approximately 0.43 cm (0.17 inches), and its small size can be advantageous during installation.

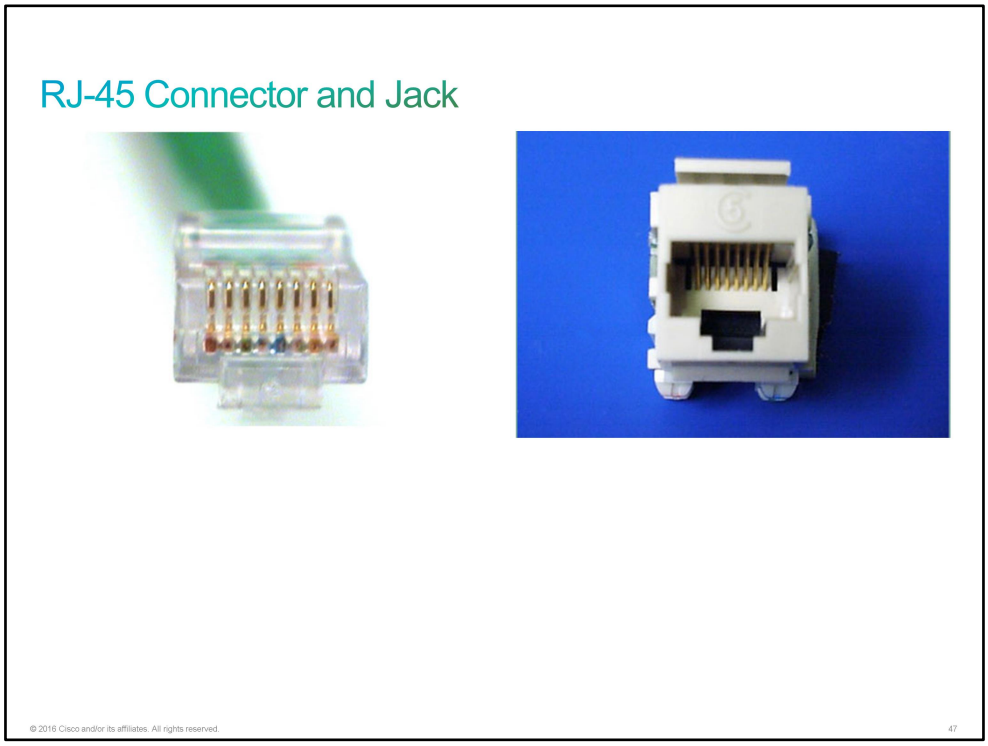
Several categories of UTP cable exist:

- **Category 1:** Used for telephone communications, not suitable for transmitting data
- **Category 2:** Capable of transmitting data at speeds of up to 4 Mbps
- **Category 3:** Used in 10BASE-T networks—can transmit data at speeds of up to 10 Mbps
- **Category 4:** Used in Token Ring networks—can transmit data at speeds of up to 16 Mbps

- **Category 5:** Capable of transmitting data at speeds of up to 100 Mbps
- **Category 5e:** Used in networks running at speeds of up to 1000 Mbps (1 Gbps)
- **Category 6:** Consists of four pairs of 24-gauge copper wires, which can transmit data at speeds of up to 10 Gbps
- **Category 6a:** Used in networks running at speeds of up to 10 Gbps
- **Category 7:** Used in networks running at speeds of up to 10 Gbps

RJ-45 Connector and Jack

UTP cables are used with RJ-45 connectors. The figure shows a UTP cable with an RJ-45 connector and a jack.



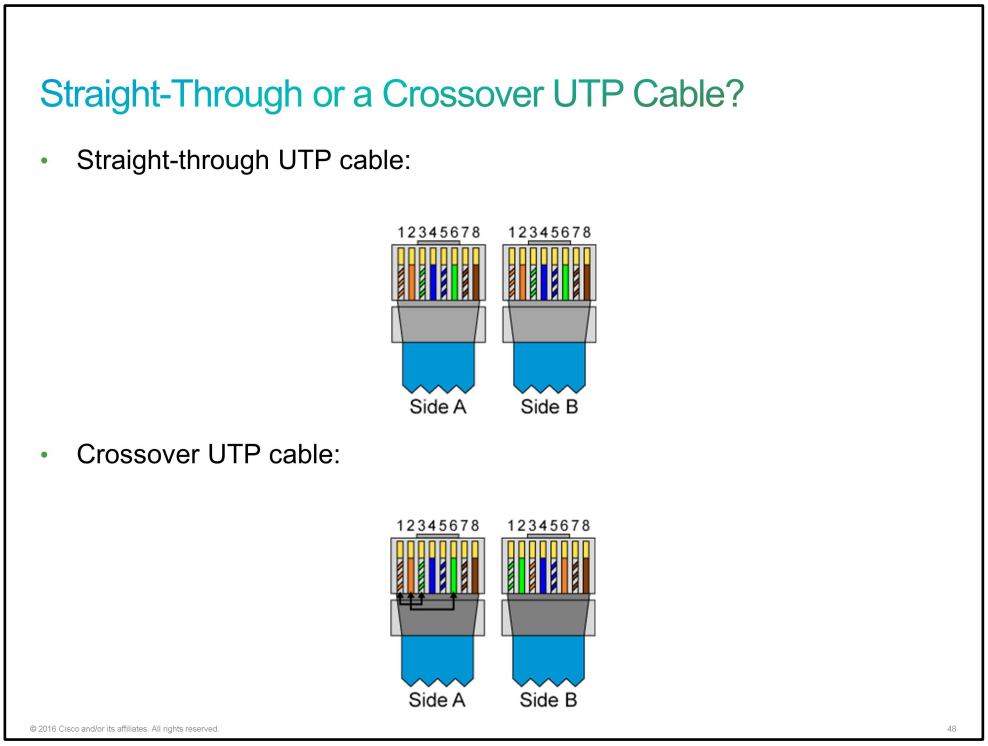
The RJ-45 plug is the male component, which is crimped at the end of the cable. As you look at the male connector from the front, as shown in the figure, the pin locations are numbered from 8 on the left to 1 on the right.

The jack is the female component in a network device, wall, cubicle partition outlet, or patch panel. As you look at the female connector from the front, as shown in the figure, the pin locations are numbered from 1 on the left to 8 on the right.

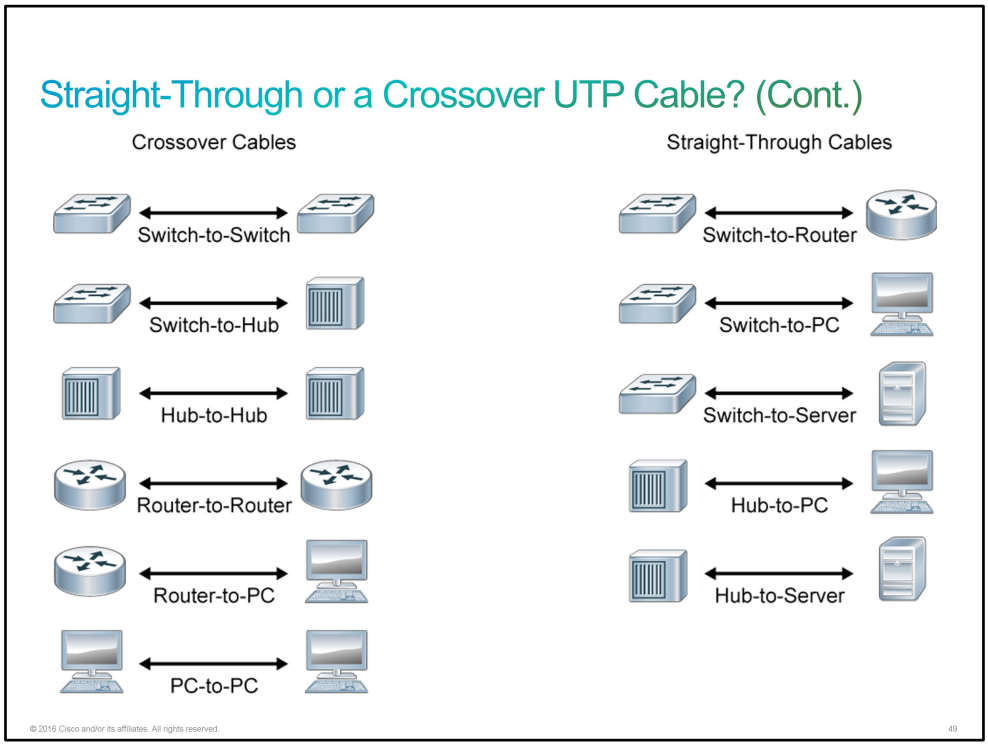
Straight-Through or a Crossover UTP Cable?

When choosing a UTP cable, you must also determine whether you need a straight-through UTP cable or a crossover UTP cable. Straight-through cables are primarily used for connecting unlike devices, while crossover cables are used for connecting like devices. To tell the difference in the two types of cabling, hold the ends of the cable next to each other with the connector side of each end facing you.

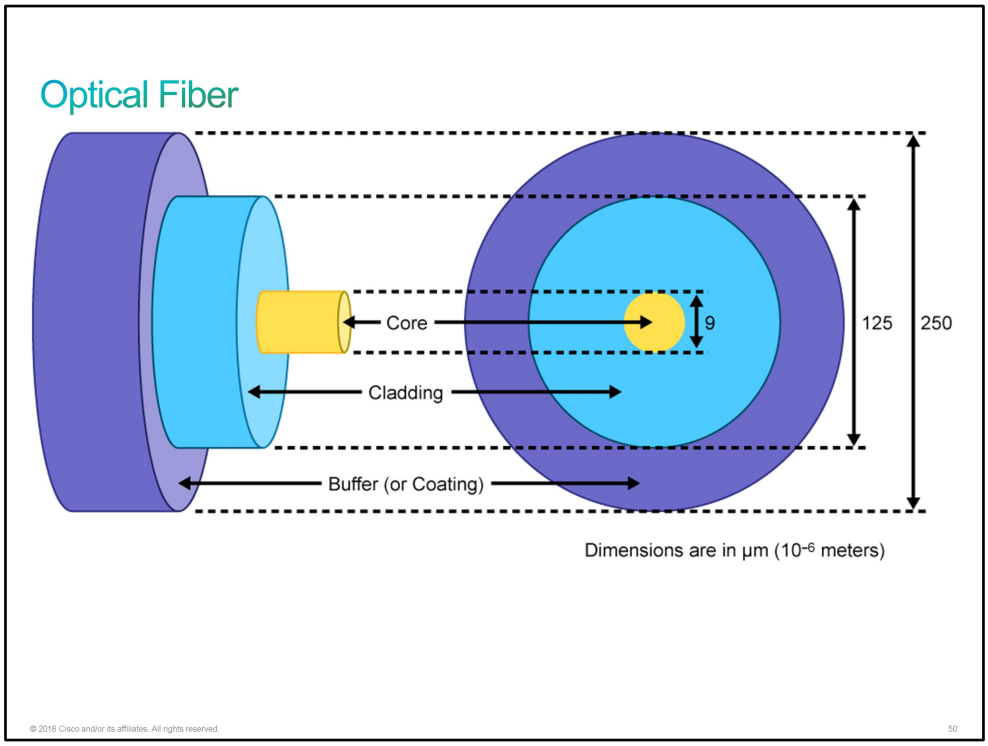
The cable is a straight-though cable if each of the eight pins corresponds to the same pin on the opposite side, as shown in the figure. The cable is a crossover cable if some of the wires on one end of the cable are crossed to a different pin on the other side of the cable, as shown in the figure.



The following figure shows when to use straight-through and crossover cables.



Optical Fiber

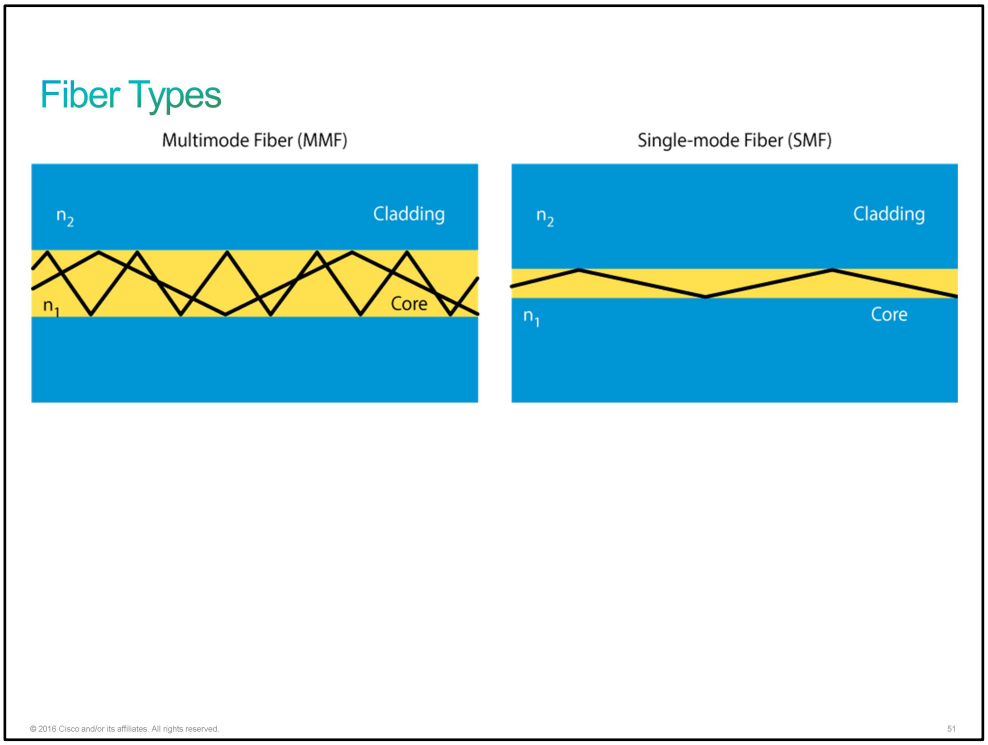


An optical fiber is a flexible, transparent fiber that is made of very pure glass (silica) and is not much larger than a human hair. It acts as a waveguide, or "light pipe," to transmit light between the two ends of the fiber. Optical fibers are widely used in fiber-optic communication, which permits transmission over longer distances and at higher bandwidths (data rates) than other forms of communication. Fibers are used instead of metal wires because signals travel along them with less loss and with immunity to electromagnetic interference.

The two fundamental components that allow a fiber to confine light are the core and the cladding. Most of the light travels from the beginning to the end inside the core. The cladding around the core provides confinement. The diameters of the core and cladding are shown in the figure, but the core diameter may vary for various fiber types. In this case, the core diameter of 9 micrometers is very small—the diameter of a human hair is about 50 micrometers. The outer diameter of the cladding is a standard size of 125 micrometers. Standardizing the size means that component manufacturers can make connectors for all fiber-optic cables.

The third element in this picture is the buffer (coating), which has nothing to do with the confinement of the light in the fiber. Its purpose is to protect the glass from scratches and moisture. The fiber-optic cable can be easily scratched and broken, like a glass pane. If the fiber is scratched, the scratch could propagate and break the fiber. Another important aspect is the need to keep the fiber dry.

Fiber Types



The most significant difference between SMF and MMF is in the ability of the fiber to send light over a long distance at high bit rates. In general, MMF is used for shorter distances at a lower bit rate than SMF. For long-distance communications, SMF is preferred. There are many variations of fiber for both MMF and SMF.

The most significant physical difference is in the size of the core. The glass in the two fibers is the same, and the index of refraction change is similar. The core diameter can make a major difference. The diameter of the fiber cladding is universal for matching fiber ends.

The effect of having different-size cores in the fiber is that the two fiber types will support various ways for the light to get through the fiber. MMF supports multiple ways for the light from one source to travel through the fiber (the source of the designation "multimode"). Each path can be thought of as a mode.

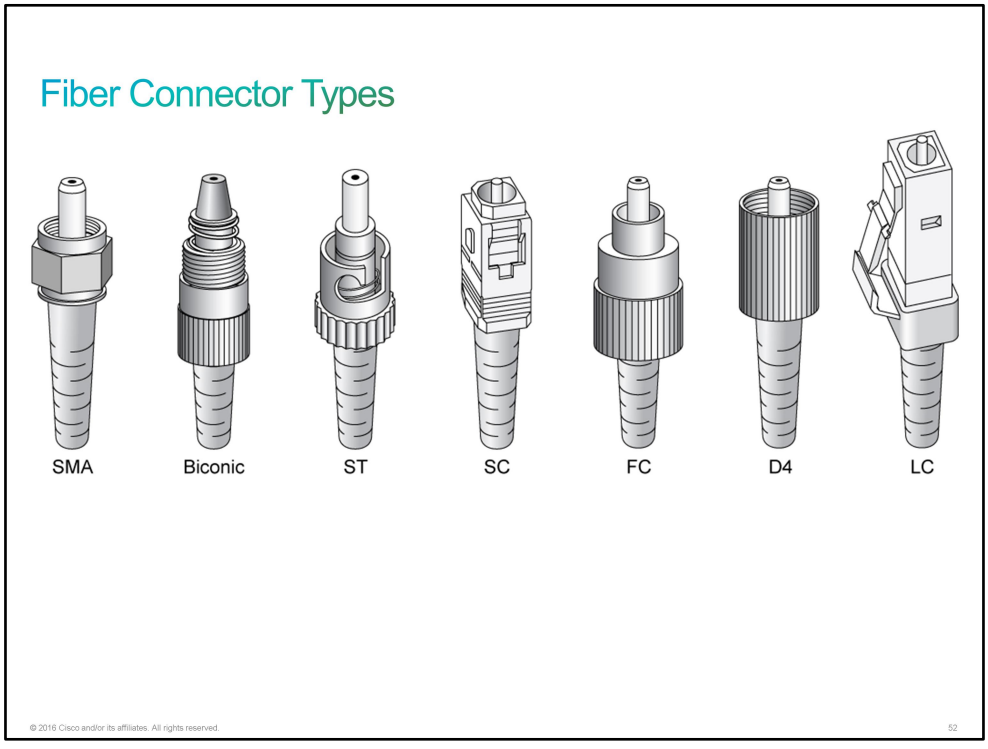
For SMF, the possible ways for light to get through the fiber have been reduced to one, a "single mode." It is not exactly one, but it is a useful approximation.

The table summarizes MMF and SMF characteristics.

MMF and SMF Characteristics

MMF Characteristics	SMF Characteristics
LED transmitter is usually used	Laser transmitter is usually used
Lower bandwidth and speed	Higher bandwidth and speed
Shorter distances	Longer distances
Less expensive	More expensive

Fiber Connector Types



An optical fiber connector terminates the end of an optical fiber. Various optical fiber connectors are available. The main differences among the types of connectors are dimensions and methods of mechanical coupling. Generally, organizations standardize on one type of connector, depending on the equipment that they commonly use, or they standardize per type of fiber (one for MMF, one for SMF). There are about 70 connector types in use today.

There are three types of connectors:

- Threaded
- Bayonet
- Push-pull

These materials are used for connectors:

- Metal
- Plastic sleeve

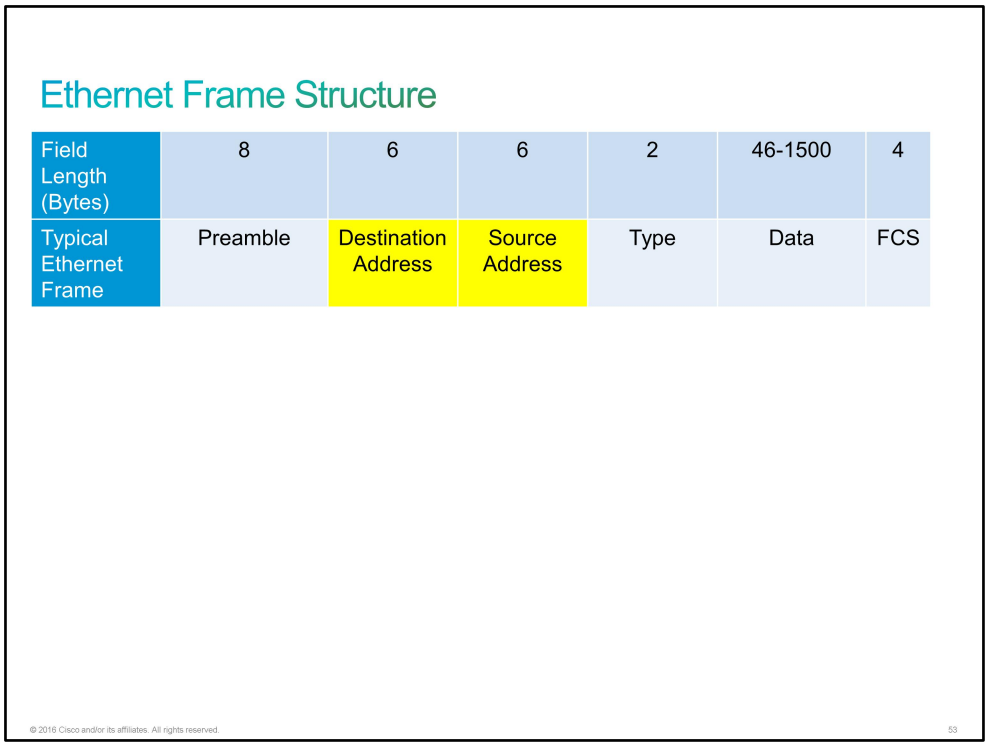
Here you can see the most common types of connectors and their typical uses:

- **ST:** For patch panels (for their durability)
- **FC:** For patch panels; used by service providers
- **SC:** For enterprise equipment
- **LC**: For enterprise equipment, commonly used on SFP modules

In data communications and telecommunications applications today, small-form-factor connectors (for example, LCs) are replacing the traditional connectors (for example, SCs), mainly to pack more connectors on the faceplate and as a result reduce system footprints.

Ethernet Frame Structure

Bits that are transmitted over an Ethernet LAN are organized into frames.



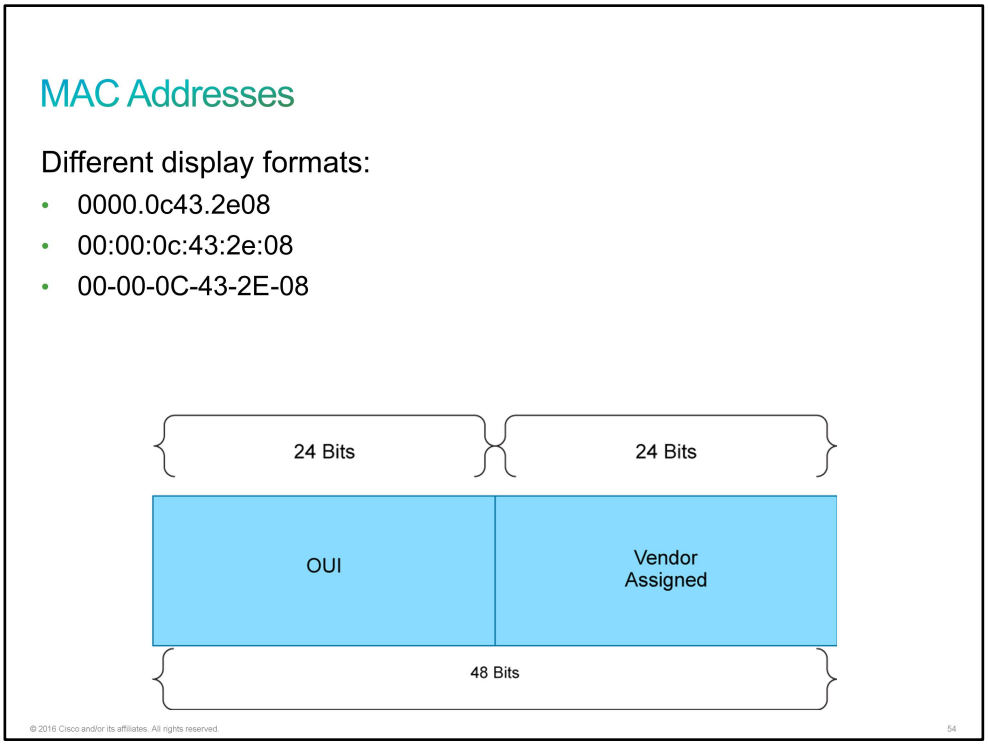
In Ethernet terminology, the container into which data is placed for transmission is called a *frame*. The frame contains header information, trailer information, and the actual data that is being transmitted.

The table shows the most important fields of a MAC layer of the Ethernet frame:

- **Preamble:** This field consists of 8 bytes of alternating 1s and 0s that are used to synchronize the signals of the communicating computers.
- **Destination address:** This field contains the address of the NIC on the local network to which the packet is being sent.
- **Source address:** This field contains the address of the NIC of the sending computer.
- **Type:** This field contains a code that identifies the network layer protocol.
- **Data and pad:** This field contains the data that is received from the network layer on the transmitting computer. This data is then sent to the same protocol on the destination computer. If the data is shorter than the minimum length of 46 bytes, a string of extraneous bits is used to pad the field.
- **FCS:** The FCS field includes a checking mechanism to ensure that the packet of data has been transmitted without corruption.

MAC Addresses

All network devices on the same network must have a unique MAC address. The MAC address is the means by which data is directed to the proper destination device. The MAC address of a device is an address that is burned into the NIC. Therefore, it is also referred to as the physical address or BIA. The MAC address is expressed as groups of hexadecimal digits that are organized in pairs or quads.



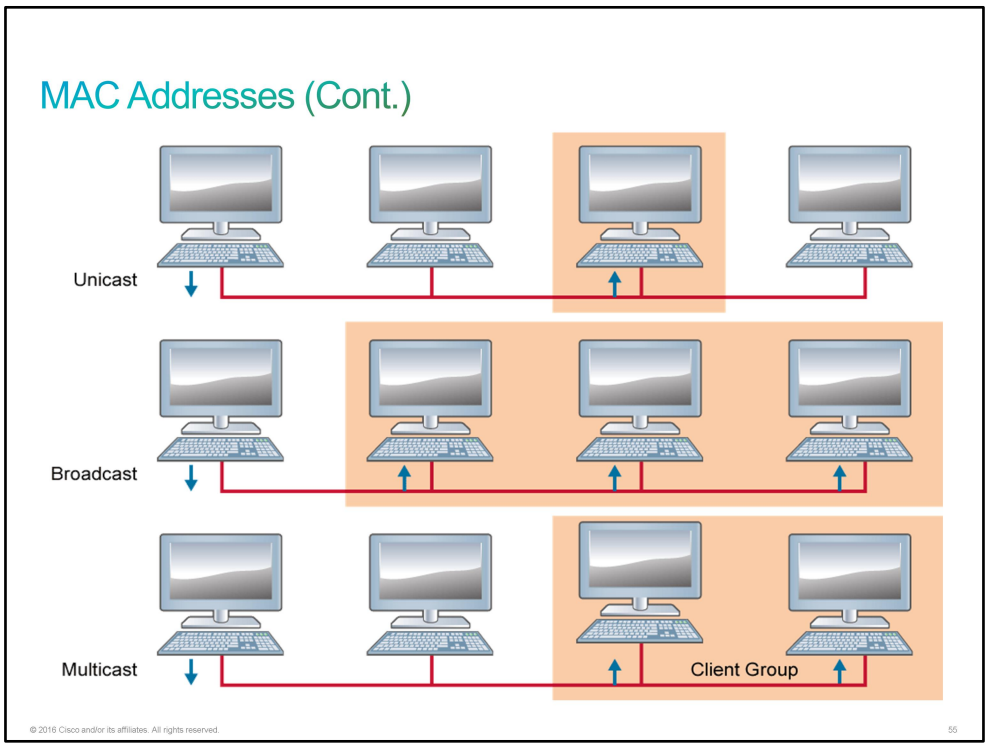
Note	What is hexadecimal?
Note	Hexadecimal (often referred to as simply *hex*) is a numbering system with a base of 16. This means that it uses 16 unique symbols as digits. The decimal system that you use on a daily basis has a base of 10, which means that it is made up of 10 unique symbols, 0 through 9. The valid symbols in hexadecimal are 0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. In decimal, A, B, C, D, E, and F equal 10, 11, 12, 13, 14, and 15. Each hexadecimal digit is 4 bits long, because it requires 4 bits in binary to count to 15. Because a MAC address is made up of 12 hexadecimal digits, it is 48 bits long.

A MAC address is made up of 12 hexadecimal numbers, which means it has 48 bits. There are two main components of a MAC. The first 24 bits constitute the OUI. The last 24 bits constitute the vendor-assigned end-station address.

- **24-bit OUI:** The OUI identifies the manufacturer of the NIC. The IEEE regulates the assignment of OUI numbers. Within the OUI, there are two bits that have meaning only when used in the destination address:
  - **Broadcast or multicast bit:** This bit indicates to the receiving interface that the frame is destined for all or a group of end stations on the LAN segment.

  - **Locally administered address bit:** Normally, the combination of the OUI and a 24-bit station address is universally unique. However, if the address is modified locally, this bit should be set.
- **24-bit vendor-assigned end-station address:** This portion uniquely identifies the Ethernet hardware.

The MAC address identifies the location of a specific computer on a LAN. Unlike other kinds of addresses that are used in networks, the MAC address should not be changed unless there is some specific need to do so.

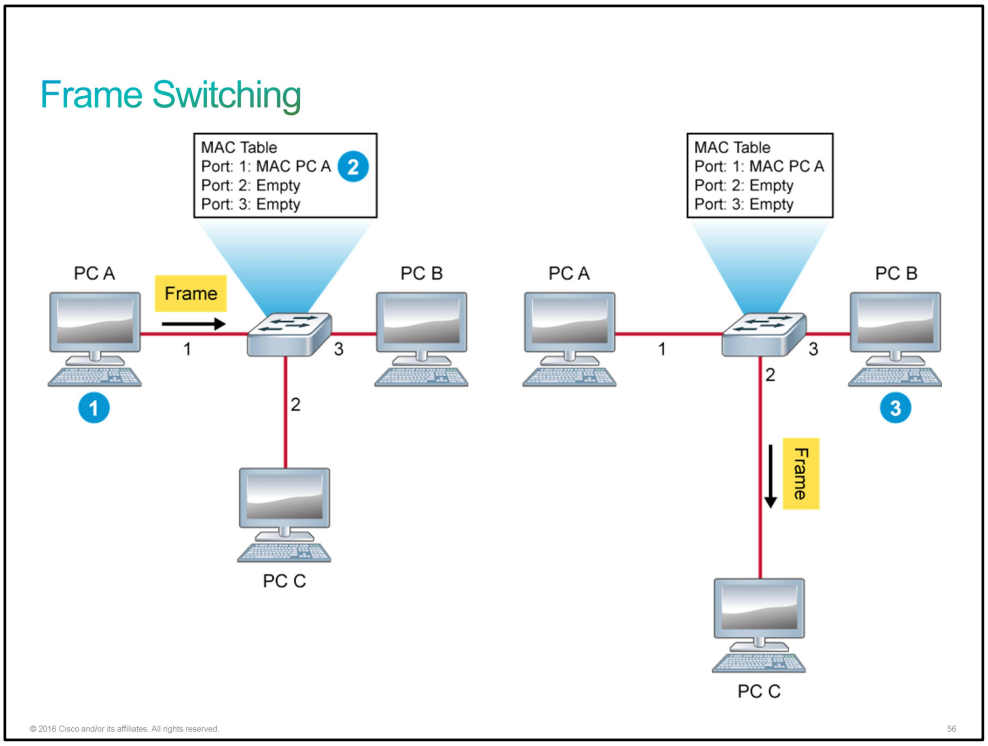


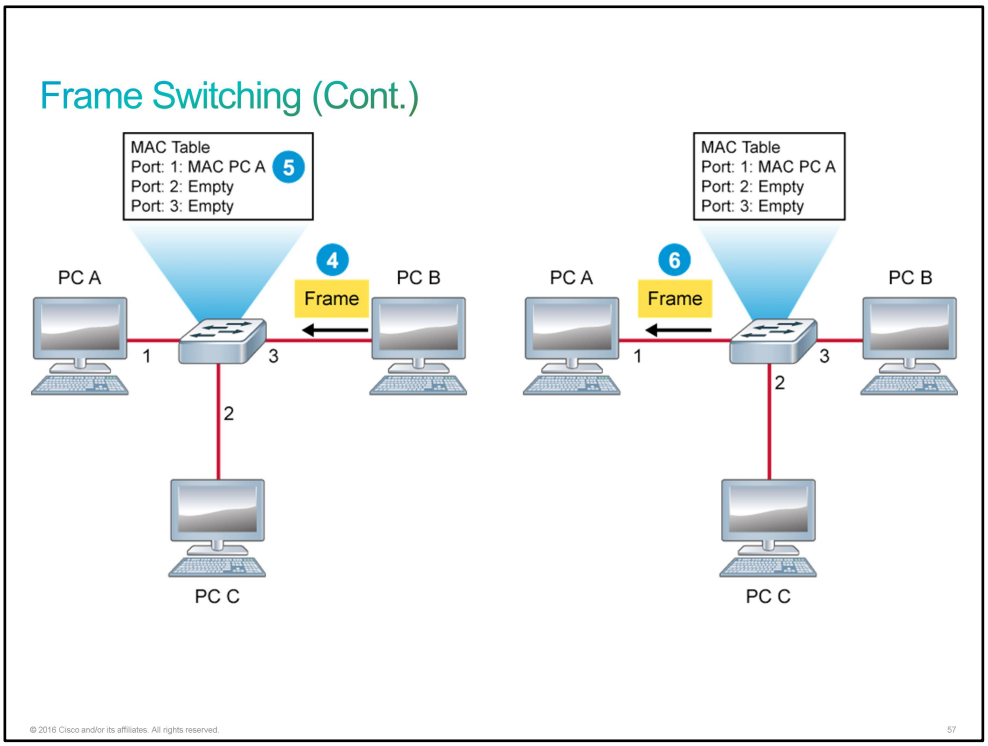
There are three major types of network communications:

- **Unicast:** Communication in which a frame is sent from one host and is addressed to one specific destination. In a unicast transmission, there is only one sender and one receiver. Unicast transmission is the predominant form of transmission on LANs and within the Internet.
- **Broadcast:** Communication in which a frame is sent from one address to all other addresses. In this case, there is only one sender, but the information is sent to all the connected receivers. Broadcast transmission is essential for sending the same message to all devices on the LAN.
- **Multicast:** Communication in which information is sent to a specific group of devices or clients. Unlike broadcast transmission, in multicast transmission, clients must be members of a multicast group to receive the information.

Frame Switching

The switch builds and maintains a table, which is called the MAC table, that matches the destination MAC address with the port that is used to connect to a node. The MAC table is stored in the CAM, which enables very fast lookups.





For each incoming frame, the destination MAC address in the frame header is compared to the list of addresses in the MAC table. Switches then use MAC addresses as they decide whether to filter, forward, or flood frames.

The switch creates and maintains a table using the source MAC addresses of incoming frames and the port number through which the frame entered the switch. When an address is not known, a switch learns the network topology by analyzing the source address of incoming frames from all the attached networks. The table describes the switching process:

Switching Frames Procedure

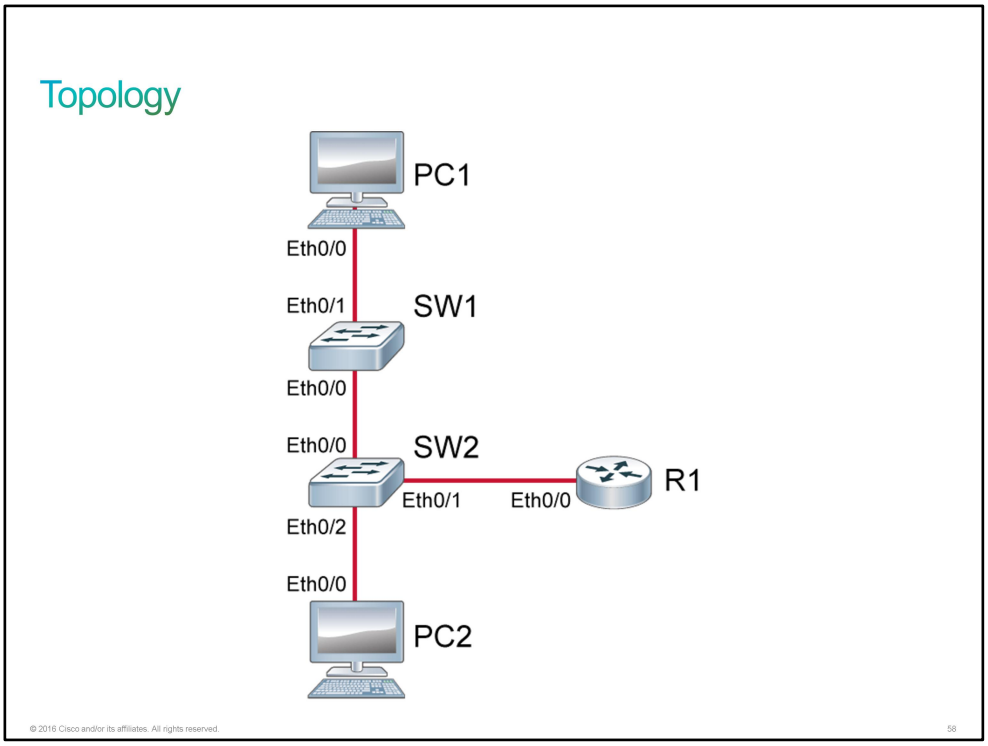
Step	Action
1	The switch receives a frame from PC A on port 1.
2	The switch enters the source MAC address and the switch port that received the frame into the MAC table.
3	The switch checks the table for the destination MAC address. Because the destination address is not known, the switch floods the frame to all the ports except the port on which it received the frame.
4	The destination device with the matching MAC address replies to the unicast with a unicast frame addressed to PC A.
5	The switch enters the source MAC address of PC B and the port number of the switch port that received the frame into the MAC table. The destination address of the frame and its associated port is found in the MAC table.
6	The switch can now forward frames between the source and destination devices without flooding because it has entries in the MAC table that identify the associated ports.

Discovery 3: Observe How a Switch Operates

Introduction

This discovery session will let you observe how a switch maintains its MAC address table, which it uses to control the forwarding of frames. The lab is prepared with the devices that are represented in the topology diagram with the IP addresses as depicted in the table. All devices are fully configured.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2

Device	Characteristic	Value
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Observe How a Switch Operates

Activity

Step 1 First, determine the MAC addresses of the Ethernet0/0 interface on PC1, PC2, and R1.

The **show interface** command displays the MAC address of the interface along with a lot of other information. To reduce the amount of output, allowing you to focus on the line that contains the MAC address, you can pipe the **show interface** output to include filter as shown here.

One at a time, access the console connection to PC1, PC2, and R1 and execute the **show interface** command.

```
PC1# sh int e0/0 | include address
  Hardware is AmdP2, address is aabb.cc00.7600 (bia aabb.cc00.7600)
  Internet address is 10.10.1.10/24


PC2# sh int e0/0 | include address
  Hardware is AmdP2, address is aabb.cc00.7700 (bia aabb.cc00.7700)
  Internet address is 10.10.1.20/24


R1# sh int e0/0 | include address
  Hardware is AmdP2, address is aabb.cc00.7500 (bia aabb.cc00.7500)
  Internet address is 10.10.1.1/24
```

In the emulated environment of the lab, the MAC addresses are similar to each other. This similarity will make it easy to distinguish them as the steps of this discovery progress.

MAC addresses in your output may be different.

Step 2 Access the console of SW2, and enter the **show mac address-table** command.

On SW2, enter the following command:

```
SW2# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.7500    DYNAMIC     Et0/1
   1    aabb.cc00.7700    DYNAMIC     Et0/2
Total Mac Addresses for this criterion: 2
```

This output is consistent with the information from the Job Aid table. The MAC address that is associated with PC2 is seen on interface Ethernet0/2 and the MAC address that is associated with R1 is associated with interface Ethernet0/1.

PC2 and R1 are both directly connected to SW2 and forward frames very regularly. It is expected that their addresses will remain in the MAC address table almost constantly. You may also see the MAC address of PC1 in the table as well.

Step 3 In this step, be prepared to press **Up Arrow** to use the IOS command recall feature to quickly repeat the **show mac address-table** command after clearing the MAC address table. After clearing the MAC address, table you should find that the MAC address for PC2 and R1 (which are directly connected to SW2) will repopulate themselves quickly.

Clear the MAC address table and use command recall to repeatedly execute the **show mac address-table** command until both addresses are populated. On SW2, enter the following command:


```
SW2# clear mac address-table dynamic
SW2# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.7500    DYNAMIC     Et0/1
   1    aabb.cc00.7700    DYNAMIC     Et0/2
Total Mac Addresses for this criterion: 2
```

The directly connected systems will populate quickly as they send Ethernet frames to the switch.

The MAC address of PC1, which is one hop away, is not in the table yet.

Step 4 Repeat a similar process on SW1. Clear the MAC address table, and then observe the population of the table.

The MAC address of PC1 should populate in just a few seconds. On SW1, enter the following commands.

```
SW1# clear mac address-table dynamic
SW1# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
SW1# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.7600    DYNAMIC     Et0/1
Total Mac Addresses for this criterion: 1
```

Step 5 Generate some traffic from PC1 to R1 and PC2. This traffic will have to travel across both SW1 and SW2. Because the MAC address of PC1 is not known to SW2 and the MAC addresses of R1 or PC2 are not known to SW1, there will be flooding of initial Ethernet frames. This flooding will happen much too fast to recognize it in the lab. But the final result should be that all three endpoints (PC1, PC2, and R1) appear in the MAC address tables of both switches.

Access the console of PC1 and ping R1.

```
PC1# ping 10.10.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Access the console of PC1 and ping PC2.

```
PC1# ping 10.10.1.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.20, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/5 ms
```

Step 6 Access the console of SW1 and execute the **show mac address-table** command. Repeat the same on SW2.

On SW1, enter the following command:

```
SW1# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.7500    DYNAMIC     Et0/0
   1    aabb.cc00.7600    DYNAMIC     Et0/1
   1    aabb.cc00.7700    DYNAMIC     Et0/0
Total Mac Addresses for this criterion: 3
```

On SW2, enter the following command.

```
SW2# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.7500    DYNAMIC     Et0/1
   1    aabb.cc00.7600    DYNAMIC     Et0/0
   1    aabb.cc00.7700    DYNAMIC     Et0/2
Total Mac Addresses for this criterion: 3
```

In the MAC address table of SW1, the MAC addresses of PC2 and R1 are both associated with the interface Ethernet0/0. Interface Ethernet0/0 is the link to SW2. Any Ethernet frames that are destined for
either of these MAC addresses must be forwarded to SW2 for delivery.

In the MAC address table of SW2, the MAC address of PC1 is associated with the interface Ethernet0/0. Interface Ethernet0/0 is the link to SW1. Any Ethernet frames that are destined for this MAC address must be forwarded to SW1 for delivery.

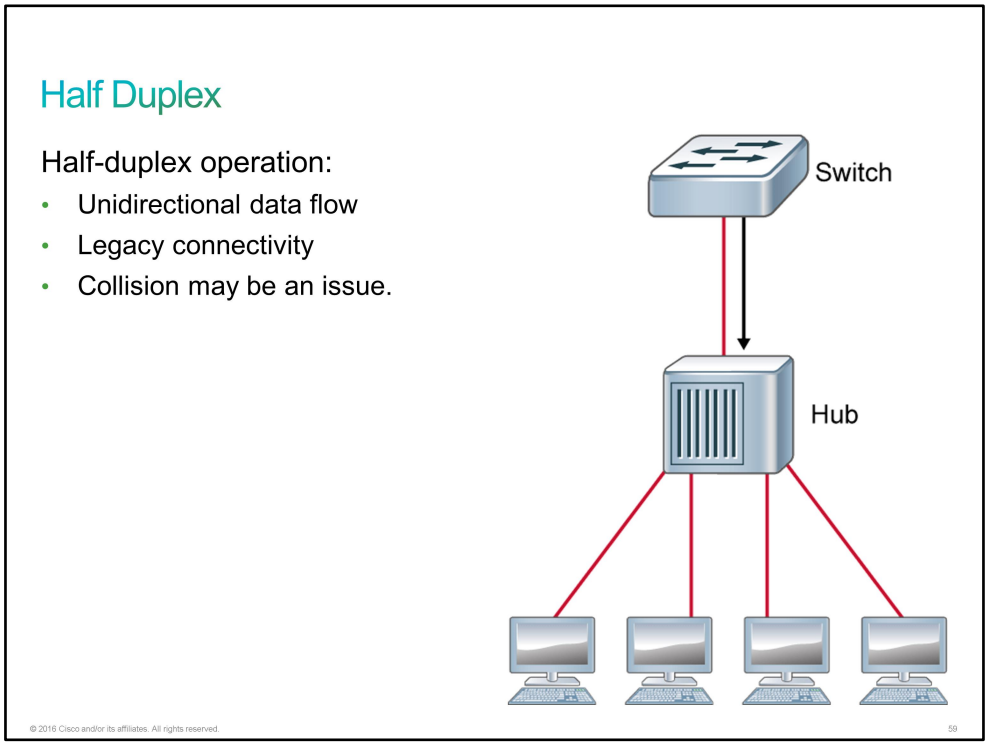
This is the end of the discovery lab.

Duplex Communication

The term *duplex communication* is used to describe a communications channel that can carry signals in both directions, as opposed to a simplex channel, which carries a signal in only one direction. There are two types of duplex settings that are used for communications on an Ethernet network, full duplex and half duplex.

Half Duplex

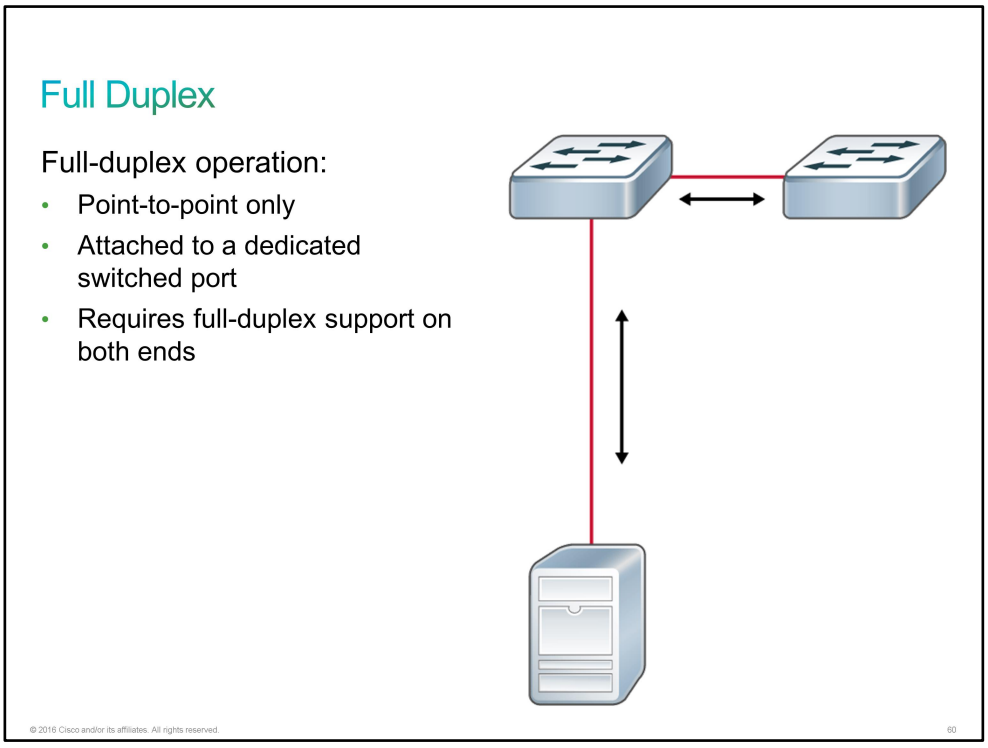
Half-duplex communication relies on a unidirectional data flow, which means that data can go only in one direction at a time. Sending and receiving data are not performed at the same time. Half-duplex communication is similar to communication with walkie-talkies or two-way radios, in which only one person can talk at a time. Because data can flow in only one direction at a time, each device in a half-duplex system must constantly wait its turn to transmit data. This constant waiting results in performance issues. As a result, full-duplex communication has replaced half duplex in more current hardware. Half-duplex connections are typically seen in older hardware, such as hubs.



If a device transmits while another is also transmitting, a collision occurs. Therefore, half-duplex communication implements Ethernet CSMA/CD to help reduce the potential for collisions and to detect them when they do happen. CSMA/CD allows a collision to be detected, which causes the offending devices to stop transmitting. Each device retransmits after a random amount of time has passed. Because the time at which each device retransmits is random, the possibility that they again collide during retransmission is very small.

Full Duplex

Full-duplex communication is like telephone communication, in which each person can talk and hear what the other person says simultaneously. In a full-duplex communication, the data flow is bidirectional, so data can be sent and received at the same time. The bidirectional support enhances performance by reducing the wait time between transmissions. Most Ethernet, Fast Ethernet, and Gigabit Ethernet NICs sold today offer full-duplex capability. In full-duplex mode, the collision-detection circuit is disabled. Frames that the two connected end nodes send cannot collide because the end nodes use two separate circuits in the network cable.



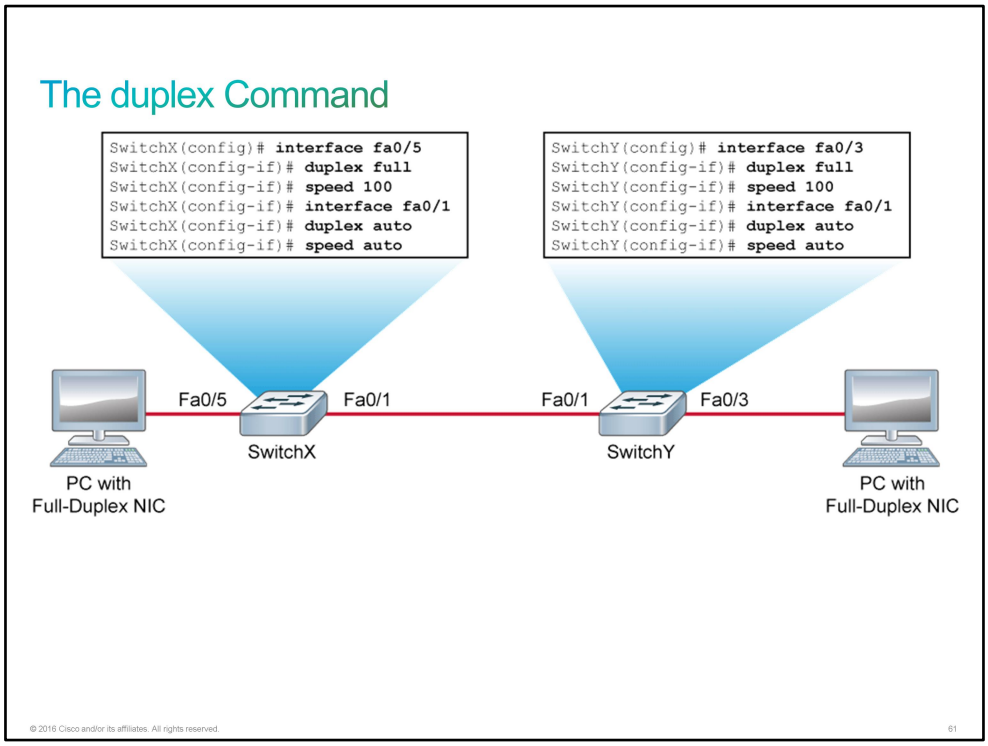
Each full-duplex connection uses only one port. Full-duplex communications require a direct connection between two nodes that both support full duplex. If one of the nodes is a switch, the switch port to which the other node is connected must be configured to operate in the full-duplex mode. The primary cause of duplex issues is mismatched settings on two directly connected devices. For example, the switch is configured for full duplex and the attached PC is configured for half duplex.

The duplex Command

The **duplex** command is used to specify the duplex mode of operation for switch ports. The **duplex** command supports the following options:

- The **full** option sets the full-duplex mode.
- The **half** option sets the half-duplex mode.
- The **auto** option sets autonegotiation of the duplex mode. With autonegotiation enabled, the two ports communicate to decide the best mode of operation.

The figure shows an example of duplex and speed configurations on the Fast Ethernet interfaces of two switches. To prevent mismatch issues, the settings on each interface are configured to match the settings of the directly connected interfaces. For example, interface Fa0/5 on SwitchX is configured for full duplex because it connects to a PC with a full-duplex NIC. The interface Fa0/1 on SwitchX is configured to autonegotiate speed and duplex settings with its neighbor, SwitchY.



For 100BASE-FX ports, the default option is **full**, and they cannot autonegotiate. 100BASE-FX ports operate only at 100 Mbps in full-duplex mode. For Fast Ethernet and 10/100/1000 ports, the default option is **auto**. The 10/100/1000 ports operate in either half- or full-duplex mode when their speed is set to 10 or 100 Mbps, but when their speed is set to 1000 Mbps, they operate only in the full-duplex mode.

Autonegotiation can at times produce unpredictable results. By default, when autonegotiation fails, a Cisco Catalyst switch sets the corresponding switch port to half-duplex mode. Autonegotiation failure happens when an attached device does not support autonegotiation. If the device is manually configured to also operate in the half-duplex mode, there is no problem. However, if the device is manually configured to operate in the full-duplex mode, there is a duplex mismatch. A duplex mismatch causes late collision errors at the end of the connection. To avoid this situation, manually set the duplex parameters of the switch to match the attached device.

You can use the **show interfaces** command in the privileged EXEC mode to verify the duplex settings on a switch. This command displays statistics and statuses for all interfaces or for the interface that you specify. The following example shows the duplex and speed settings of a Fast Ethernet interface.

```
SwitchX# show interfaces FastEthernet0/5
FastEthernet0/5 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 0022.91c4.0e01 (bia 0022.91c4.0e01)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 10/100BaseTX
  input flow-control is off, output flow-control is unsupported
  <... output omitted ...>
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     7289 packets input, 927927 bytes, 0 no buffer
     Received 184 broadcasts (1380 multicasts
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 1380 multicast, 0 pause input
     0 input packets with dribble condition detected
     39965 packets output, 7985339 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```


Challenge

1. Which type of physical medium is no longer use for Ethernet?

   A. coaxial
   B. twisted copper pair
   C. fiber-optic
   D. wireless

2. Match the UTP cable category with its description.

category 5	used in 10BASE-T networks—can transmit data at speeds of up to 10 Mbps
category 5e	capable of transmitting data at speeds of up to 100 Mbps
category 6a	used in networks running at speeds of up to 1000 Mbps (1 Gbps)
category 3	used in networks running at speeds up to 10 Gbps

3. Which is not an optical fiber connector type?

   A. threaded
   B. bayonet
   C. push-pull
   D. metal
   E. RJ-45

4. Place the Ethernet frame fields into the correct order.

2.	FCS
3.	preamble
6.	type
5.	source address
1.	destination address
4.	data and pad

5. Which three formats are correct MAC address formats? (Choose three.)

   A. 0000.0c43.2e08
   B. 00:00:0c:43:2e:08
   C. 00-00-0C-43-2E-08
   D. 00000C432E08
   E. 0000-0C43-2E08
   F. 00:00-0c:43-2e:08

6. Match the network communication type with its description.

multicast	communication in which a frame is sent from one host and addressed to one specific destination
broadcast	communication in which a frame is sent from one address to all other addresses
unicast	communication in which information is sent to a specific group of devices or clients

7. Which three characteristics are full-duplex operation characteristics? (Choose three.)

   A. unidirectional data flow
   B. point-to-point only
   C. legacy connectivity
   D. attached to a dedicated switched port
   E. collision may be an issue
   F. requires full-duplex support on both ends

Answer Key

Challenge

1. A
2.

category 3	used in 10BASE-T networks—can transmit data at speeds of up to 10 Mbps
category 5	capable of transmitting data at speeds of up to 100 Mbps
category 5e	used in networks running at speeds of up to 1000 Mbps (1 Gbps)
category 6a	used in networks running at speeds up to 10 Gbps

3. E
4.

6.	FCS
1.	preamble
4.	type
3.	source address
2.	destination address
5.	data and pad

5. A, B, C
6.

unicast	communication in which a frame is sent from one host and addressed to one specific destination
broadcast	communication in which a frame is sent from one address to all other addresses
multicast	communication in which information is sent to a specific group of devices or clients

7. B, D, F

Module 2: Establishing Internet Connectivity

Introduction

This module explains all necessary technologies to successfully establish an Internet connection and provide Internet access to local area users. Internet protocol and IP addressing are explained. Subnets and steps needed to perform subnetting are introduced. Transport layer protocols, TCP and UDP, are briefly described, but details are avoided; the focus is to provide enough knowledge to later understand how NAT operates. The router, its role and operation, is explained. Basic configuration steps are shown with configuration examples. The packet delivery process is illustrated, and the ARP protocol is introduced. Access control lists are introduced at an overview level, but only a standard ACL is explained at this point due to its use in NAT. Other use cases and configuration options are discussed in the next module. Finally, NAT is explained. Configuration examples are used, with the focus on configuring PAT.

Lesson 1: Understanding the TCP/IP Internet Layer

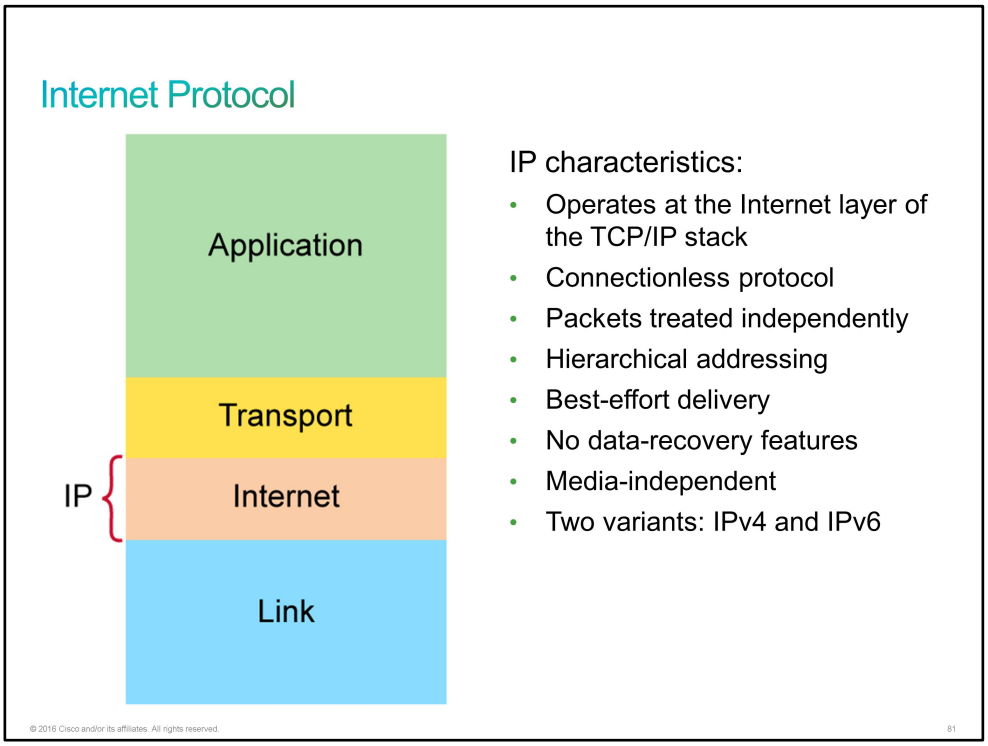
Introduction

CCS is happy with your work involving switches and wants to start assigning you to deployments involving routers and Internet connectivity. Before they do, you will need to demonstrate your knowledge of IPv4 and IPv4 addressing.

You will need to understand the Internet Protocol and its general characteristics, IP addressing, its structure (network and the host portion of addresses), and the IPv4 address fields. You will go through Internet address classes and the types of reserved IP addresses, where you will focus on relevant types (network address, broadcast address). You will also mention the problem of memorizing IP addresses and how DNS is introduced as a solution. At the end, you will explain how to verify of IP settings on end host devices.

Internet Protocol

The IP component of TCP/IP determines where packets of data are routed, based on their destination addresses. IP has certain characteristics that are related to how it manages this function.



IP uses packets to carry information through the network. A packet is a self-contained, independent entity that contains data and sufficient information to be routed from the source to the destination without reliance on earlier exchanges.

IP has these characteristics:

- IP operates at Layer 3 of the OSI model (network layer) and at the Internet layer of the TCP/IP stack.
- IP is a connectionless protocol, in which a one-way datagram is sent to the destination without advance notification to the destination device. The destination device receives the data and does not return any status information to the sending device.
- Each packet is treated independently, which means that each packet can travel a different way to the destination.
- IP uses hierarchical addressing, in which the network ID is the equivalent of a street, and the host ID is the equivalent of a house or an office building on that street.
- IP provides service on a best-effort basis and does not guarantee packet delivery. A packet can be misdirected, duplicated, or lost on the way to its destination.
- IP does not provide any special features that recover corrupted packets. Instead, the end systems of the network provide these services.
- IP operates independently of the medium that is carrying the data.
- There are two types of IP addresses: IPv4 and IPv6.

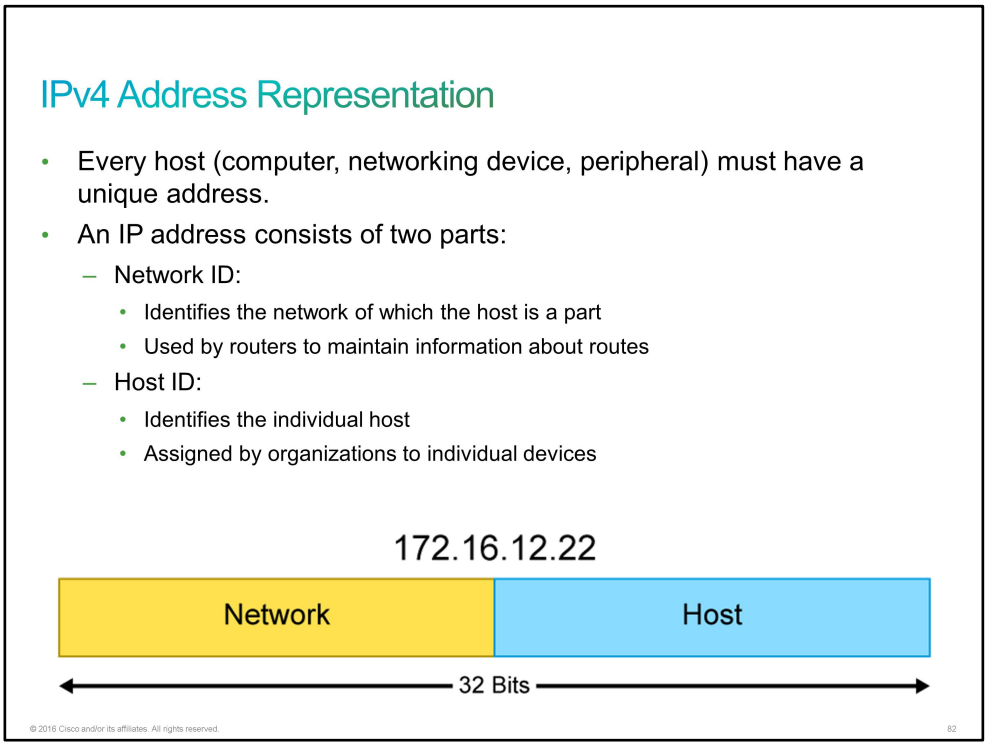
Example: Delivering a Letter Through a Postal Service

An analogy for IP services would be mail delivery by a postal service. For example, you live in San Francisco and your mother lives in New York. You write three letters to your mother. You seal each letter in a separate envelope, address each letter, and write your return address in the upper left-hand corner of each envelope.

You deposit the three letters in the outgoing mail slot at your local post office. The postal service makes its best attempt to deliver all three letters to your mother in New York. However, the postal service will not guarantee that the letters will arrive at their destination. It will not guarantee that all three letters will be processed by the same carrier or take the same route. And it will not guarantee that the letters will arrive in the order in which you mailed them.

IPv4 Address Representation

Every host must be assigned a unique address to communicate on an IP network. Common IP hosts include PCs, laptops, printers, web servers, smart phones, and tablets.



Physical street addresses are necessary to identify the locations of specific homes and businesses so that mail can reach them efficiently. In the same way, logical IP addresses are used to identify the location of specific devices on an IP network so that data can reach those network locations. Every host that is connected to the Internet has a unique 32-bit IP address that identifies it. Without a structure for allocating all those IP addresses, it would be impossible to route packets efficiently. Learning how IP addresses are structured and how they function in the operation of a network provides an understanding of how IP packets are forwarded over networks using TCP/IP.

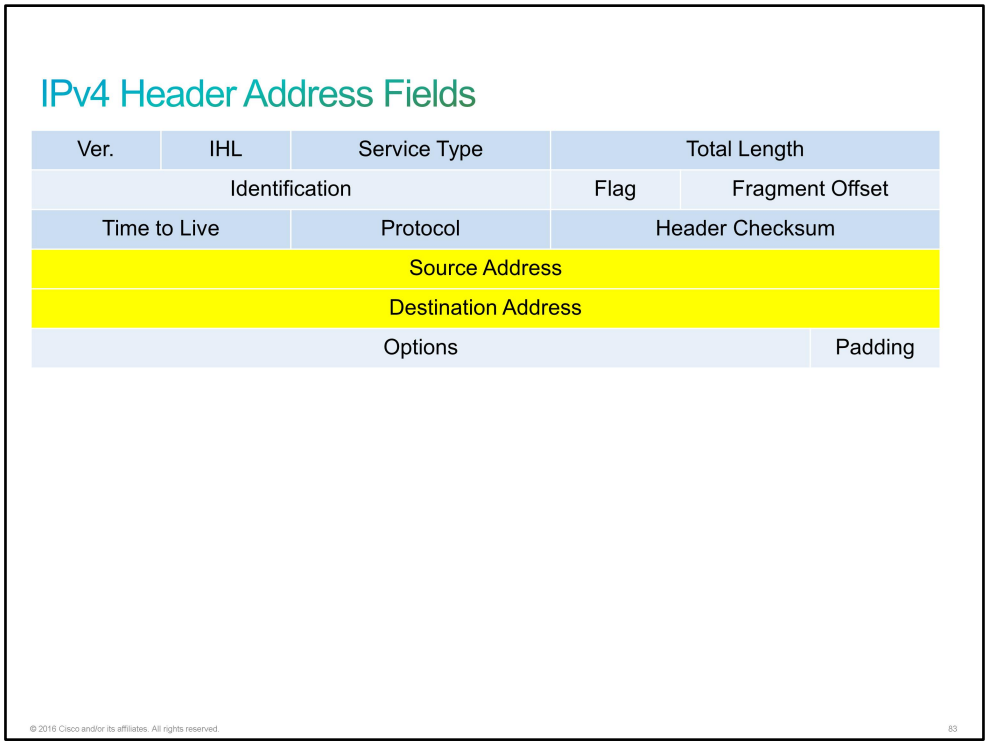
Two versions of the Internet Protocol are in use, IPv4 and IPv6. IPv4 is the most common type of address that is currently used on the Internet. It has been the mainstay protocol since the 1980s. The IPv6 address was designed to solve the problem of global IP address exhaustion. Adoption of IPv6 was initially very slow, but is now reaching wider deployment.

An IP address is hierarchical and consists of two parts:

- **The network address portion (network ID):** Describes the network of which this IP address is a part.
- **The host address portion (host ID):** Identifies a specific endpoint. These endpoints are the servers, computers, and other devices that are connected to the network. Host IDs are assigned to individual devices (end-user devices, printers, network devices, and so on).

IPv4 Header Address Fields

Before you can send an IP packet, there needs to be a format that all IP devices agree upon to route a packet from the source to the destination. All that information is contained in the IP header. The IPv4 header is basically a container for values that are required to achieve host-to-host IP communications. Some fields (such as the IP version) are static, and others, such as TTL, are modified continually in transit.



The IPv4 header has several fields. At this point, these two fields are most important for you:

- **Source Address:** Specifies the 32-bit binary value that represents the IP address of the sending endpoint
- **Destination Address:** Specifies the 32-bit binary value that represents the IP address of the receiving endpoint

Other fields in the header are the following:

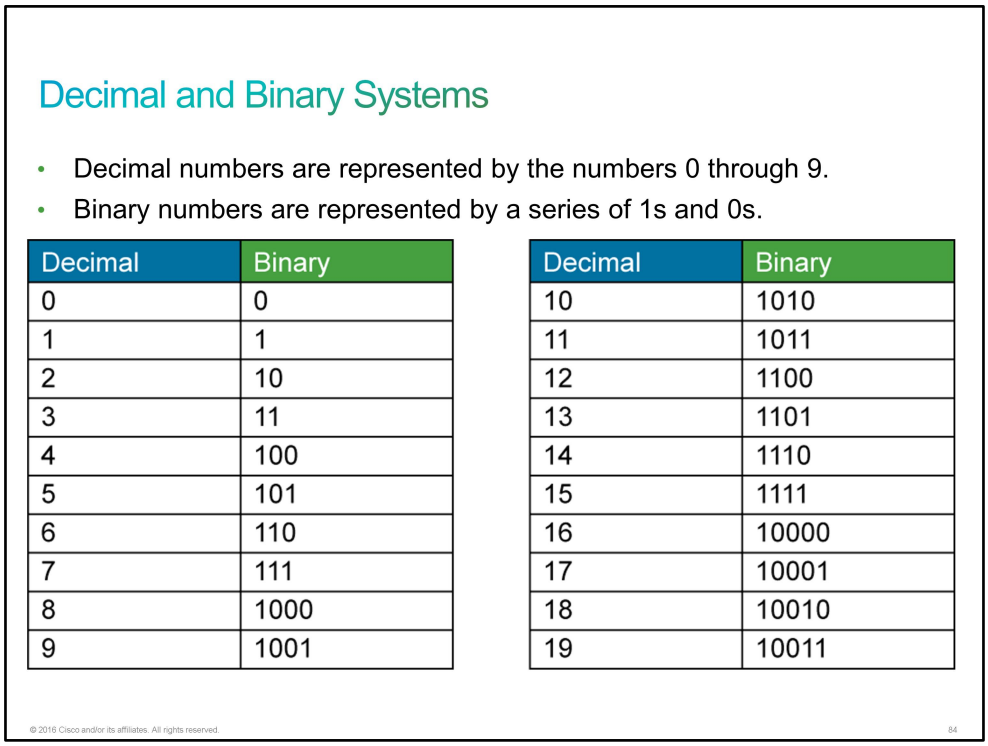
- **Version:** Describes the version of the Internet Protocol
- **IHL:** Internet Header Length; describes the length of the header
- **Service Type:** Provides information on the desired quality of service
- **Total Length:** Describes the length of a packet, including header and data
- **Identification:** Used for unique fragment identification
- **Flag:** Sets various control flags regarding fragmentation
- **Fragment Offset:** Indicates where specific fragment belongs
- **Time to Live:** Limits the lifetime of a packet
- **Protocol:** Indicates the protocol that is used in the data portion of an IP packet

- **Header Checksum:** Used for header error detection
- **Options:** Includes optional parameters
- **Padding:** Used to ensure that the header ends on a 32-bit boundary

If you would like to learn more about the IPv4 header fields, go to http://tools.ietf.org/html/rfc791.

Decimal and Binary Systems

The decimal (base 10) system is the numbering system that is used in everyday mathematics, and the binary (base 2) system is the foundation of computer operations. Network device addresses use the binary system to define their location on the network. The IP address is based on a dotted-decimal notation of a binary number. Having a basic understanding of the mathematical properties of a binary system helps you to understand networking.



Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

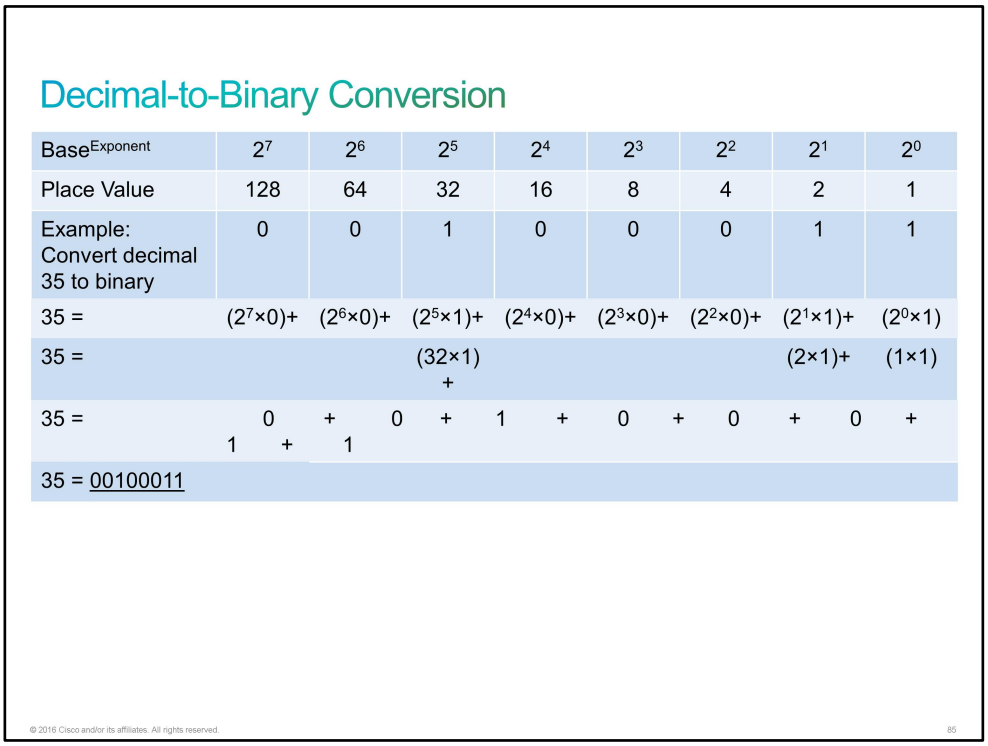
Decimal	Binary
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011

In the decimal system, the digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. When quantities higher than 9 are required, the decimal system begins with 10 and continues all the way to 99. Then the decimal system begins again with 100, and so on, with each column to the left raising the exponent by 1.

The binary system uses only the digits 0 and 1. Therefore, the first digit is 0, followed by 1. If a quantity higher than 1 is required, the binary system goes to 10, followed by 11. The binary system continues with 100, 101, 110, 111, then 1000, and so on. This figure shows the binary equivalent of the decimal numbers 0 through 19.

Decimal-to-Binary Conversion

You can convert decimal numbers to binary numbers through a specific process.



This figure shows a simple binary conversion of the decimal number 35. The base exponent line shows base-2 numbers and their exponents (2 × 2 = 4 × 2 = 8, and so on). The decimal value of the base exponent number is listed in the second row, and the binary number is displayed in the third row. The table describes the steps to determine the binary number. Note that the first 2 bits of the binary number are 0s. These zeros are known as leading 0s. In reality, the decimal number 35 would only be a 6-bit binary number. Because IP addresses are laid out as four sets of octets, the binary number is made into an octet by placing 0s to the left of the 6-bit number.

The table shows the steps for converting the number 35 to a binary number.

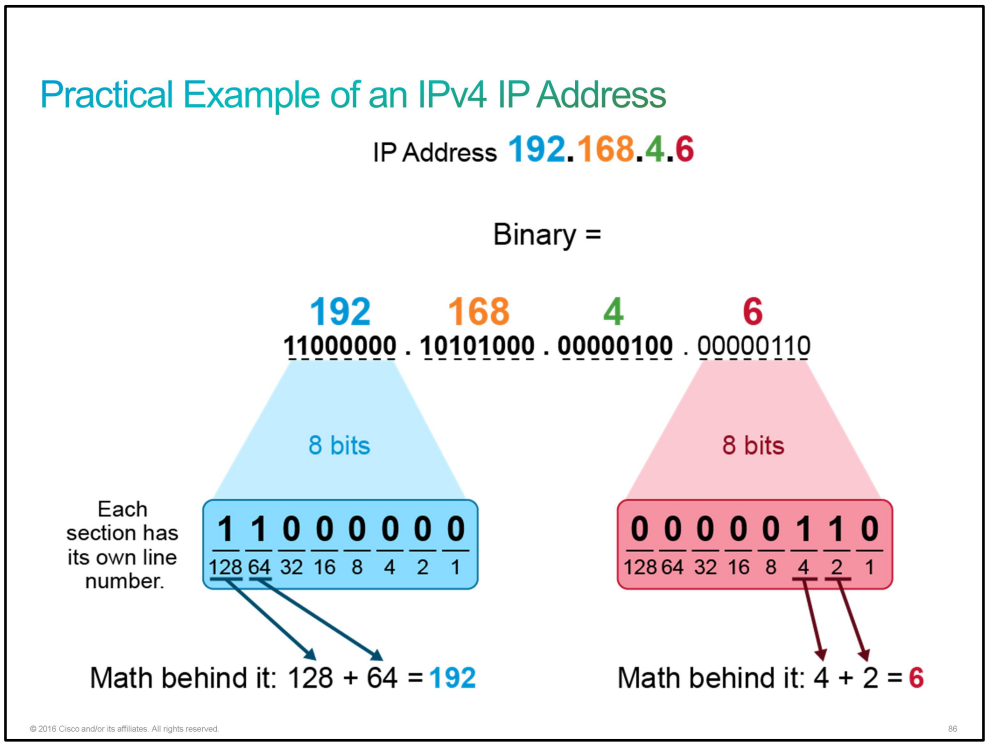
Procedure for Converting a Decimal Number to a Binary Number

Step	Action
1.	Looking at the table, what is the greatest power of 2 that is less than or equal to 35? 128 does not go into 35, so place a 0 in that column.
2.	64 does not go into 35, so place a 0 in that column.
3.	2^5 (32) is smaller than 35. 32 goes into 35 one time. Place a 1 in that column.
4.	Calculate how much is left over by subtracting 32 from 35. The result is 3.
5.	Check to see if 16 (the next lower power of 2) fits into 3. Because it does not, a 0 is placed in that column.

Step	Action
6.	The value of the next number is 8, which is larger than 3, so a 0 is placed in that column also.
7.	The next value is 4, which is still larger than 3, so it, too, receives a 0.
8.	The next value is 2, which is smaller than 3. Because 2 fits into 3 one time, place a 1 in that column.
9.	Subtract 2 from 3, and the result is 1.
10.	The decimal value of the last bit is 1, which fits into the remaining number. Therefore, place a 1 in the last column. The binary equivalent of the decimal number 35 is 00100011.

Practical Example of an IPv4 IP Address

An IP address is in the form of four sets of decimal numbers that are separated by dots. The decimal number in each and every set is in the range from 0 to 255. Each set is called an octet. There are four octets in an IP address. All computer systems understand IP addresses only in the binary form. The following example shows how an IP address is translated into the binary form.



An easy way how to begin with the translation is to create basic placeholders for your binary conversion. Place 32 underscores (_) under the IP address and separate them with dots. Why 32? Because you will translate each set into an 8-bit binary number; 4 × 8 = 32. Each underscore has a set value. Values from right to left are: 1, 2, 4, 8, 16, 32, 64, 128. Can you see a pattern? Each value increases by 2^n. Now you start adding these numbers so that the total equals your set. The figure shows the translation for the first and last set.

The value of the first set is 192. If you add 128 and 64 (first two underscore values) that equals 192. Every time that you add a number, write 1 on the underscore (each bit is represented by a 1 or a 0). When you finish, write 0 on the underscores that you did not use (those bits are set to off). The first octet is translated to 11000000.

The value of the last set is 6. If you use the second and third bit (from the right), you match that number, so the last set translates to 00000110. Try to translate the remaining sets on your own.

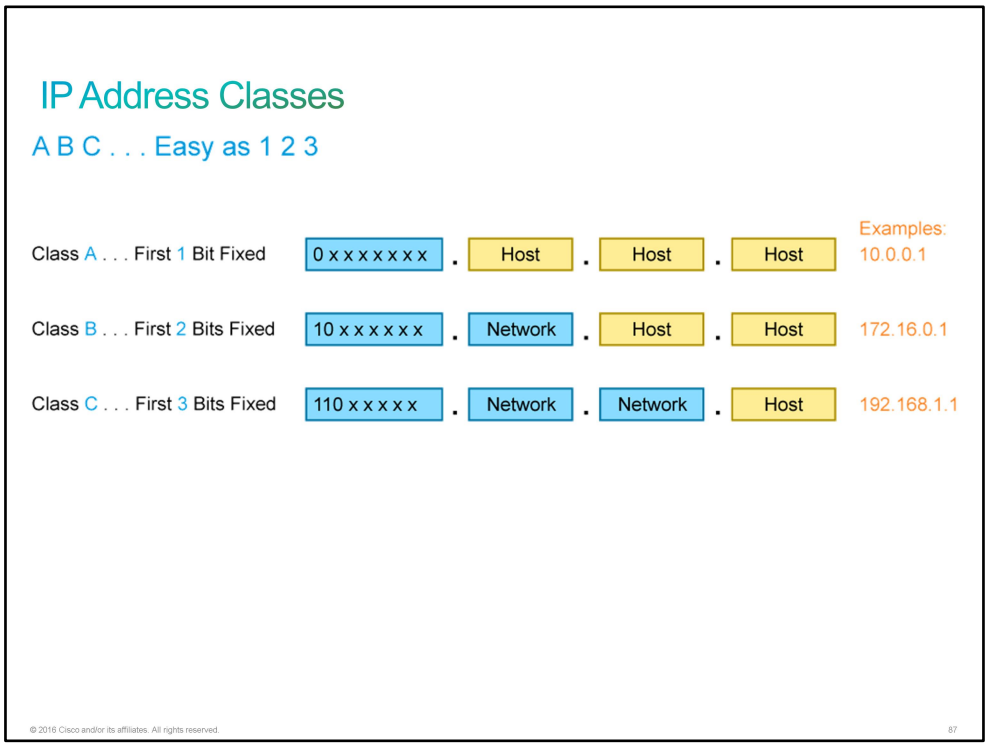
IP Address Classes

To accommodate networks of different sizes and help in classifying them, IP addresses are divided into categories that are called *classes*.

Assigning IP addresses to classes is known as *classful addressing*. During the early days of the Internet, the IANA determined the classes.

Each IP address is broken down into a network ID and a host ID. In addition, a bit or bit sequence at the start of each address determines the class of the address. The figure shows three of the five IP classes.

Note	IP hosts use only Class A, B, and C IP addresses for unicast (host-to-host) communications. Class D and Class E are included for completeness, but they are outside the scope of this discussion.



Class A

A Class A address block is designed to support extremely large networks with more than 16 million host addresses. The Class A address uses only the first octet (8 bits) of the 32-bit number to indicate the network address. The remaining 3 octets of the 32-bit number are used for host addresses. The first bit of a Class A address is always a 0. Because the first bit is a 0, the lowest number that can be represented is 00000000 (decimal 0), and the highest number that can be represented is 01111111 (decimal 127). However, these two network numbers, 0 and 127, are reserved and cannot be used as network addresses. Any address that has a value between 1 and 126 in the first octet of the 32-bit number is a Class A address.

Class B

The Class B address space is designed to support the needs of moderate to large networks with more than 65,000 hosts. The Class B address uses 2 of the 4 octets (16 bits) to indicate the network address. The remaining two octets specify host addresses. The first 2 bits of the first octet of a Class B address are always binary 10. Starting the first octet with binary 10 ensures that the Class B space is separated from the upper levels of the Class A space. The remaining 6 bits in the first octet may be populated with either 1s or 0s. Therefore, the lowest number that can be represented with a Class B address is 10000000 (decimal 128), and the highest number that can be represented is 10111111 (decimal 191). Any address that has a value in the range of 128 to 191 in the first octet is a Class B address.

Class C

The Class C address space is the most commonly available address class. This address space is intended to provide addresses for small networks with a maximum of 254 hosts. In a Class C address, the first three octets (24 bits) of the IP address identify the network portion, with the remaining octet reserved for the host portion. A Class C address begins with binary 110. Therefore, the lowest number that can be represented is 11000000 (decimal 192), and the highest number that can be represented is 11011111 (decimal 223). If an address contains a number in the range of 192 to 223 in the first octet, it is a Class C address.

Class D

Class D (multicast) IP addresses are dedicated to multicast applications such as streaming media. Multicasts are a special type of broadcast, in that only hosts that request to participate in the multicast group will buffer the traffic to the IP address of that group. Unlike IP addresses in Classes A, B, and C, multicast addresses are always the destination address and never the source. A Class D address begins with binary 1110. Therefore, the lowest number that can be represented is 11100000 (decimal 224), and the highest number that can be represented is 11101111 (decimal 239). If an address contains a number in the range of 224 to 239 in the first octet, it is a Class D address.

Class E

Class E (reserved) IP addresses are reserved by the IANA as a block of experimental addresses. Class E IP addresses should never be assigned to IP hosts. A Class E address begins with binary 1111. Therefore, the lowest number that can be represented is 11110000 (decimal 240), and the highest number that can be represented is 11111111 (decimal 255). If an address contains a number in the range of 240 to 255 in the first octet, it is a Class E address.

This table shows the IP address range of the first octet (in decimal and binary) for Class A, B, and C IP addresses, and also the number of host addresses that are available for each class of addresses.

IP Address Classes (Cont.)

Class A, B, and C First Octet Binary and Decimal Ranges

IP Address Class	First Octet Binary Range	First Octet Decimal Range	Maximum Number of Hosts per Subnet
Class A	00000001 to 01111110	1–126	16,777,214
Class B	10000000 to 10111111	128–191	65,534
Class C	11000000 to 11011111	192–223	254
Class D (Multicast)	11100000 to 11101111	224–239	—
Class E (Reserved)	11110000 to 11111111	240–255	—

Note Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used. This range is reserved for loopback and diagnostic functions.

Each class of network allows a fixed number of hosts. In a Class A network, the first octet is assigned to the network, leaving the last three octets to be assigned to hosts. The first host address in each network (all 0s) is reserved for the actual network address, and the final host address in each network (all 1s) is reserved for broadcasts.

In a Class A network, the last three octets are used as host addresses. An octet is 8 bits, so 3 octets is 24 bits. So the maximum number of hosts in a Class A network is $2^{24} – 2$ (subtracting the network and broadcast reserved addresses), or 16,777,214.

In a Class B network, the first 2 octets are assigned to the network. The final 2 octets (16 bits) are assigned to hosts. The maximum number of hosts in a Class B network is $2^{16} – 2$, or 65,534.

In a Class C network, the first 3 octets are assigned to the network. The final octet (8 bits) can be assigned to hosts, so the maximum number of hosts is $2^{8} – 2$, or 254.

Class D and Class E IP addresses are special cases. They are never assigned to hosts as source IP addresses.

Reserved IPv4 Addresses

Certain IP addresses are reserved and cannot be assigned to individual devices on a network. Reserved IP addresses include a network address, which is used to identify the network itself, and a broadcast address, which is used for broadcasting packets to all the devices on a network.

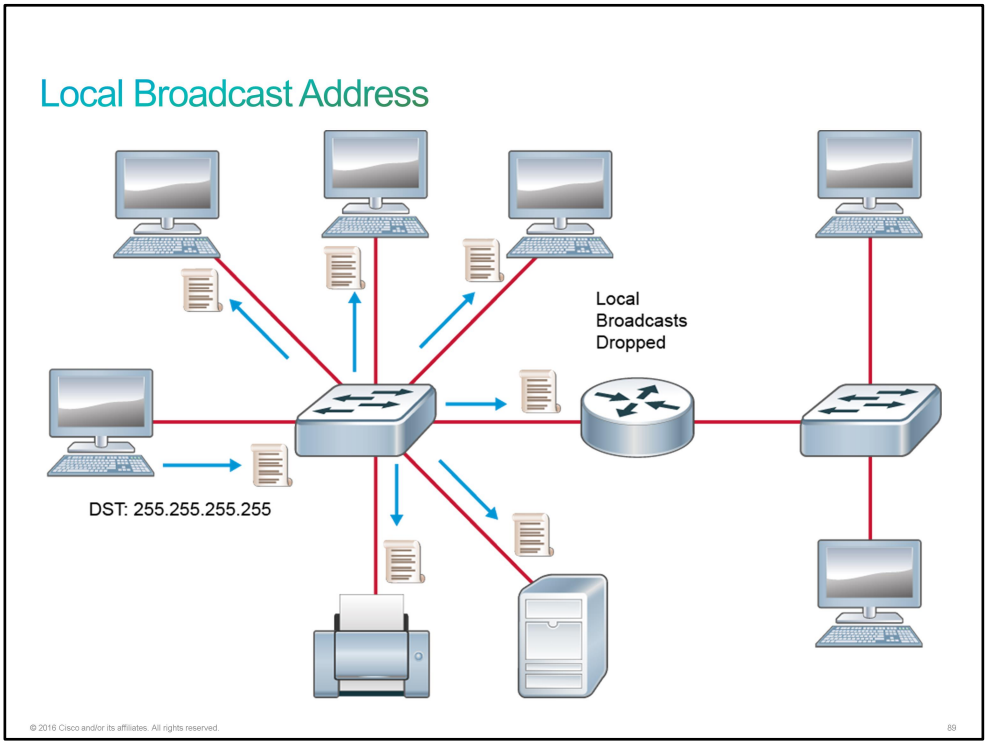
Network Address

The network address is a standard way to refer to a network. An IP address that has binary 0s in all the host bit positions is reserved for the network address. For example, in a Class A network, 10.0.0.0 is the IP address of the network containing the host 10.1.2.3. All hosts in 10.0.0.0 will have the same network bits. The IP address 172.16.0.0 is a Class B network address, and 192.16.1.0 is a Class C network address. A router uses the network IP address when it searches its IP routing table for the destination network location.

In a Class B network address, the first two octets are the network portion. The last two octets contain 0s because those 16 bits are for host numbers and are used for devices that are attached to the network. In the IP address 172.16.0.0, the first two octets are reserved for the network address and are never used as an address for any device that is attached to it. An example of an IP address for a device on the network is 172.16.16.1. In this example, 172.16 is the network address portion and 16.1 is the host address portion.

Local Broadcast Address

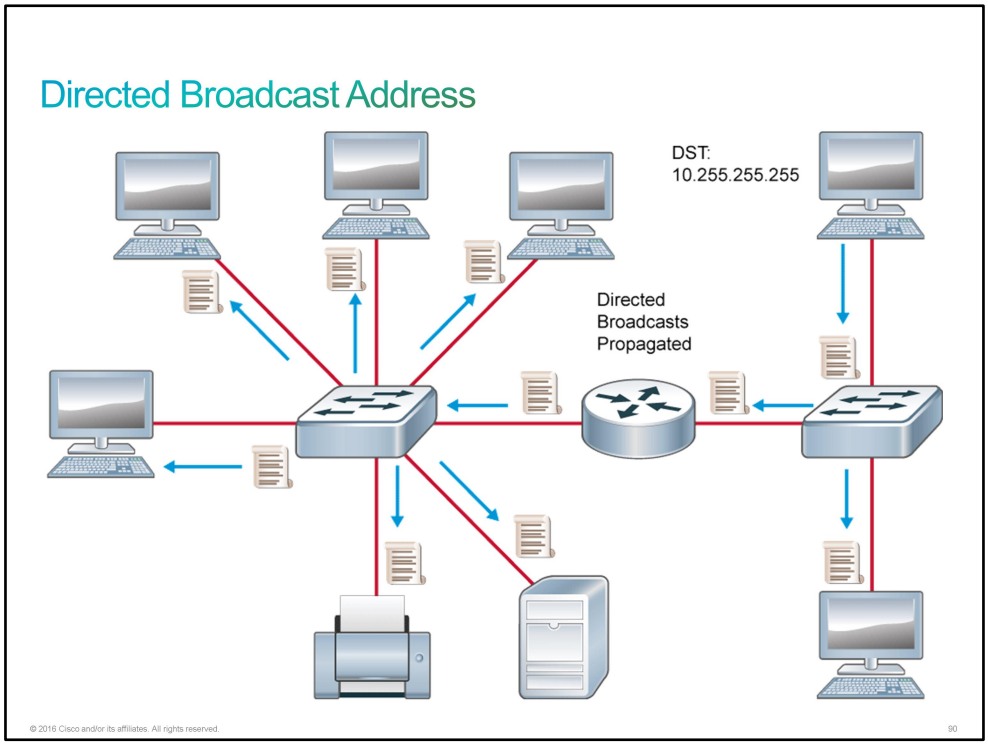
If an IP device wants to communicate with all the devices on the local network, it sets the destination address to all 1s (255.255.255.255) and transmits the packet. For example, hosts that do not know their network number and are asking a server for it may use this address. The local broadcast is never routed beyond the local network (subnet).



Directed Broadcast Address

The broadcast IP address of a network is a special address for each network that allows communication to all the hosts in that network. To send data to all the hosts in a network, a host can send a single packet that is addressed to the broadcast address of the network. The broadcast address uses the highest address in the network range, which is the address in which the bits in the host portion are all 1s. For the network 10.0.0.0, with 8 network bits, the broadcast address would be 10.255.255.255. This address is also referred to as the *directed broadcast*. Assuming a hypothetical network where every Class A IP host address was in use, a ping to 10.255.255.255 would receive a response from all 16,777,214 hosts.

For the network address 172.16.0.0, the last 16 bits make up the host field (or host part of the address). The broadcast that would be sent out to all the devices on that network would include a destination address of 172.16.255.255.



Note	The directed broadcast address can be routed over your company's intranet and over the Internet. In the 1990s, a popular DoS attack referred to as a Smurf used directed broadcasts to send so much traffic to an intended victim that they could not send or receive any legitimate traffic. For this reason, Cisco IOS defaults to disallowing directed broadcasts. This capability can be restored with the **ip directed-broadcast** command in the global configuration mode. It is a best practice to leave directed broadcasts disabled unless you have a specific use case. Routers began using the **no ip directed-broadcast** command as a platform default starting with Cisco IOS Release 12.0.

Local Loopback Address

A local loopback address is used to let the system send a message to itself for testing. The loopback address creates a shortcut method for TCP/IP applications and services that run on the same device to communicate with one another. A typical local loopback IP address is 127.0.0.1. On a Microsoft Windows 7 host, you can ping any IP address in the 127.0.0.0/8 range.

Network ID

The network portion of an IP address is also referred to as the *network ID*. A network ID is important because most hosts on a network can directly communicate only with devices in the same network. If the hosts need to communicate with devices that have interfaces that are assigned to another network ID, they must go through a network device that can route data between the networks. This holds true even when the devices share the same physical media segment. The network ID cannot be assigned to a host. For example, 10.0.0.0 cannot be assigned because it is the network ID for that Class A network.

A network ID enables a router to transmit an IP packet onto the appropriate network segment.

All Zeros Address

The address 0.0.0.0 indicates the host in "this" network and is used only as a source address. An example use case is the DHCP assignment process before the host has a valid IP address.

For more information about reserved IPv4 addresses, refer to RFC 5735, at http://tools.ietf.org/html/rfc5735.

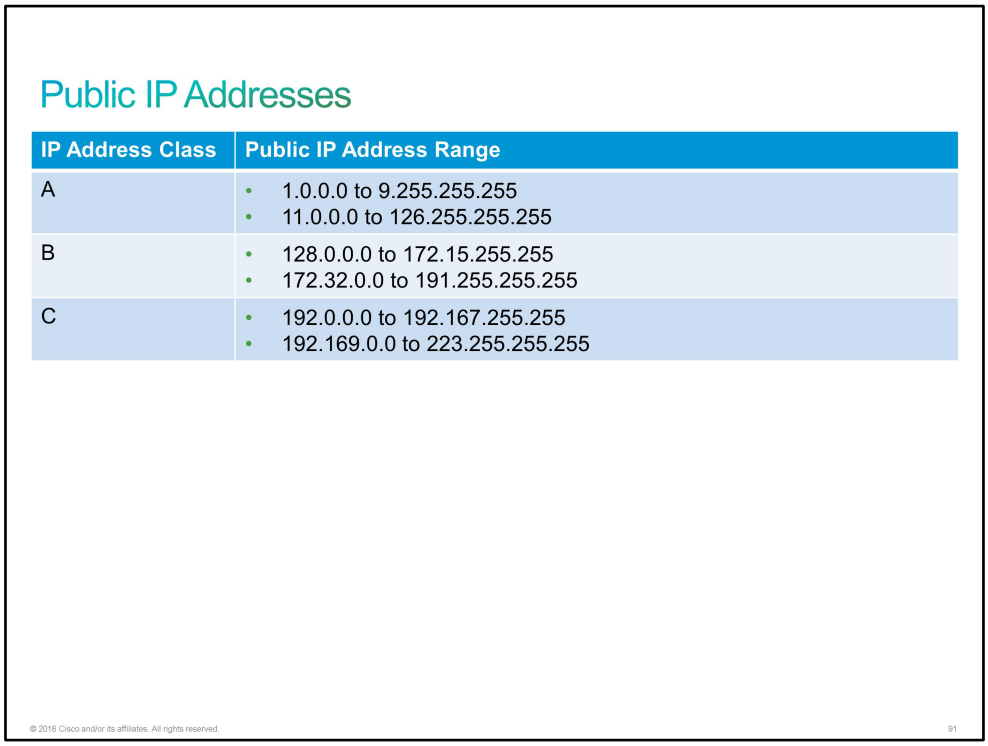
Private vs. Public IP Addresses

As the Internet began to grow exponentially in the 1990s, it became clear that if the current growth trajectory continued, eventually there would not be enough IP addresses for everyone that wanted one. Work began on a permanent solution, which would become IPv6, but in the interim, several other solutions were developed. These included NAT, CIDR, private IP addressing, and VLSM.

Public IP Addresses

Hosts that are publicly accessible over the Internet require public IP addresses. Internet stability depends directly on the uniqueness of publicly used network addresses. Therefore, a mechanism is needed to ensure that addresses are, in fact, unique. The allocation of IP addresses is managed by the IANA.

With few exceptions, businesses and home Internet users receive their IP address assignment from their LIR, which is typically their ISP. These IP addresses are called "provider-dependent" because they are linked to the ISP. If you change ISPs, you will have to readdress your Internet-facing hosts.

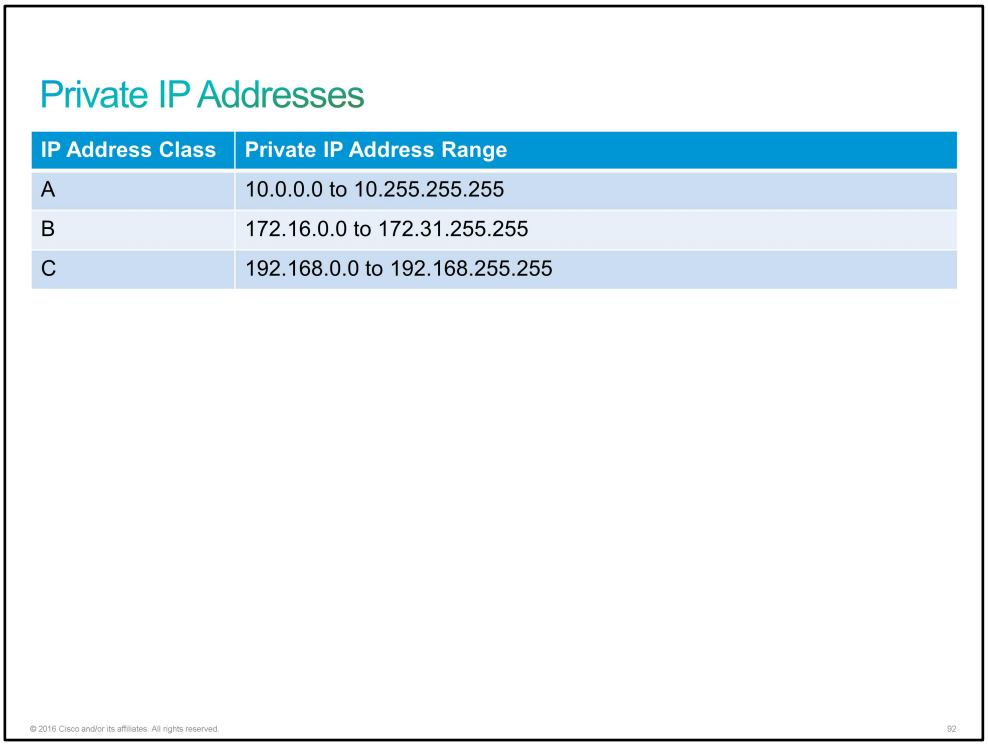


Private IP Addresses

Internet hosts require a globally unique IP address, but private hosts that are not connected to the Internet can use any valid address, as long as it is unique within the private network. However, because many private networks exist alongside public networks, deploying arbitrary IP addresses is strongly discouraged.

In February of 1996, the IETF published RFC 1918, "Address Allocation for Private Internets," to both ease the accelerating depletion of globally routable IP addresses and provide companies an alternative to using arbitrary IP addresses. Three blocks of IP addresses (one Class A network, 16 Class B networks, and 256 Class C networks) are designated for private, internal use.

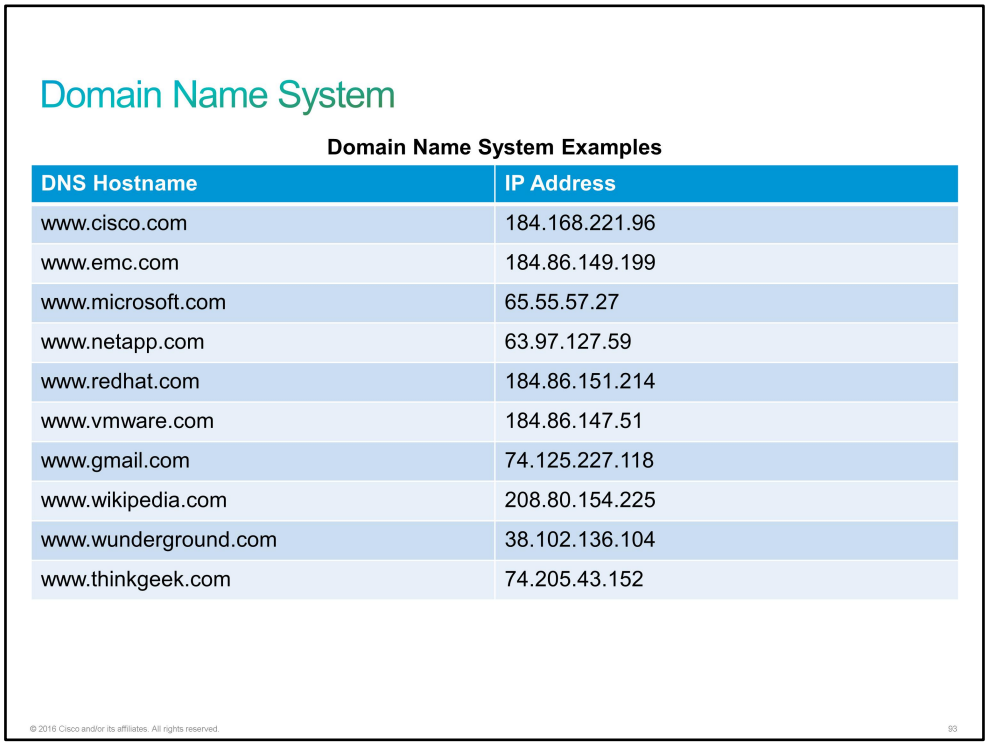
Addresses in these ranges are not routed on the Internet backbone. Internet routers are configured to discard private addresses. In a private intranet, these private addresses can be used instead of globally unique addresses. When a network that is using private addresses requires Internet connectivity, it is necessary to translate the private addresses to public addresses. This translation process is called NAT. A router or firewall is often the network device that performs NAT.



Domain Name System

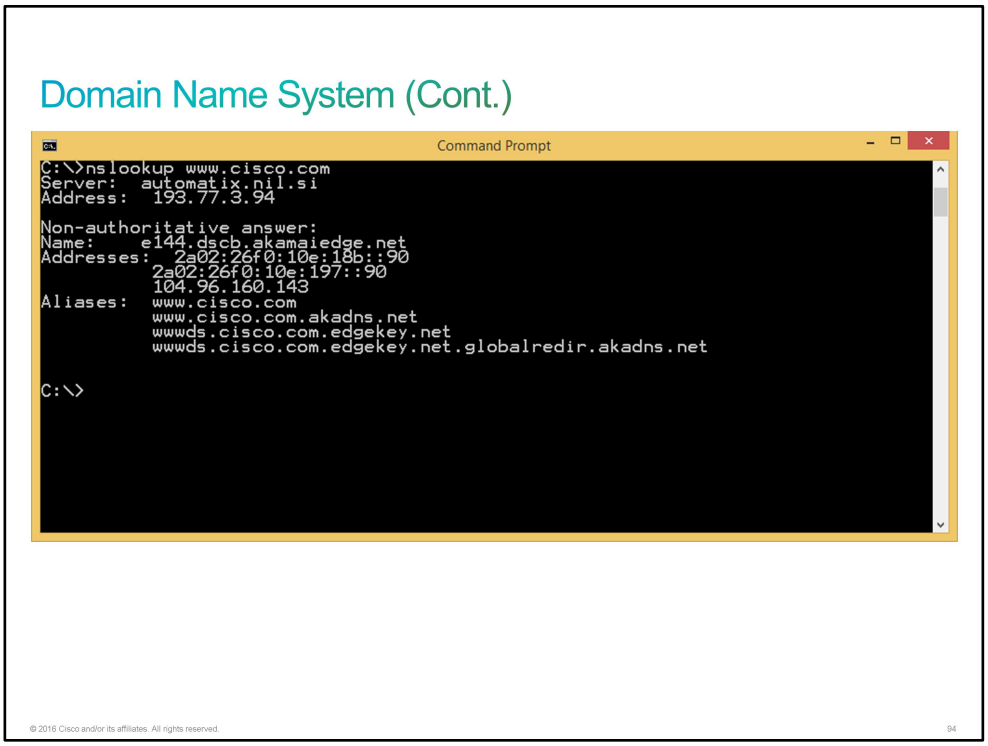
The DNS provides an efficient way to convert human-readable names of IP end systems into machine-readable IP addresses that are necessary for routing.

On TCP/IP networks, hosts are assigned their unique 32-bit IP addresses in the familiar dotted quad notation (x.x.x.x) so that they can send and receive messages over the local network and the Internet. Although not every IPv4 address has been assigned, there are billions of possible destinations. If there were no DNS, you would have to remember the IP address of every host that you would like to reach. Imagine having to remember the IP addresses for even the top 10 websites that you visit.



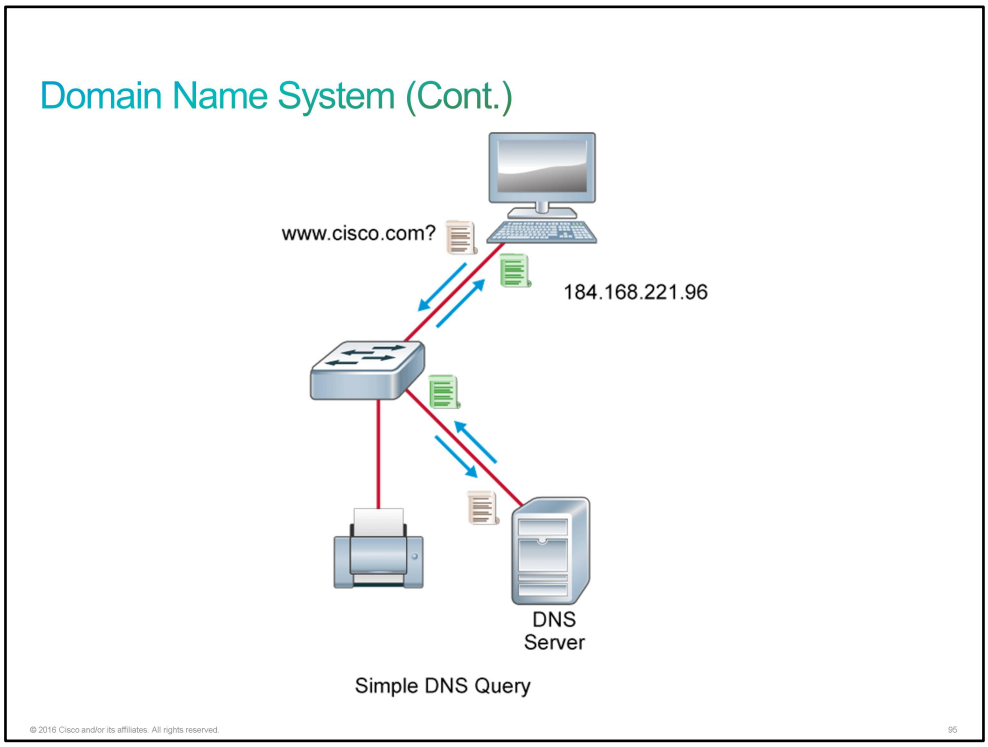
DNS uses a distributed database that is hosted on several servers, which are located around the world, to resolve the names that are associated with dotted-decimal IP addresses. The DNS protocol defines an automated service that matches resource names with the required numeric network address.

An easy way to observe DNS in action can be performed in a command window in Microsoft Windows, Apple MacOS X, or your favorite Linux distribution. When the command window is open, enter **nslookup www.cisco.com**. This command tells your IP host to make a DNS query. The result will appear below your query.



Note	The DNS transaction that is represented in the illustration is a simplification for the purposes of demonstration. In practice, there are DNS transactions, which are external to the local DNS server, that are necessary to receive an answer to the host query.

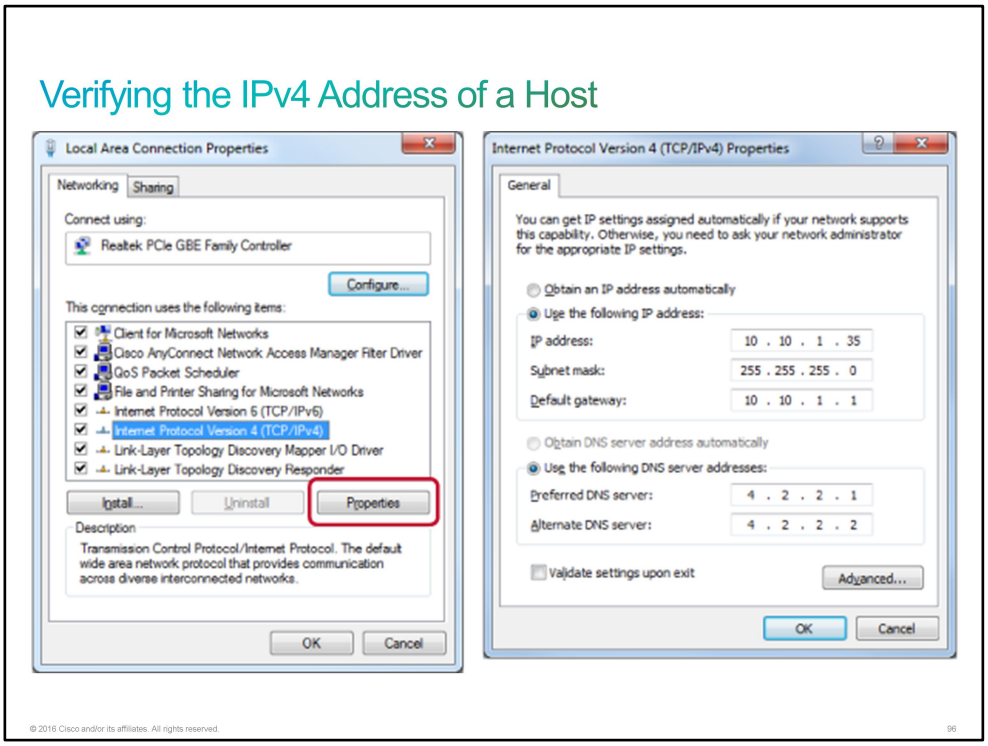
Your host sends a DNS query for the IP address of www.cisco.com. If your DNS server has the answer cached, it returns the answer directly.



Verifying the IPv4 Address of a Host

All operating systems that are capable of TCP/IP communications include utilities for configuring, managing, and monitoring the IP networking configuration. Operating systems such as Microsoft Windows, Apple Mac OS X, and most Linux variants include both CLI and GUI tools.

On a PC running Microsoft Windows 7 Enterprise, the Networking tab of a given adapter allows you to view and set the IP address that is associated with that adapter. In this example, the PC is manually configured with a static IP address.



Note	Navigating to the TCP/IP network settings varies widely depending on the operating system that is installed.

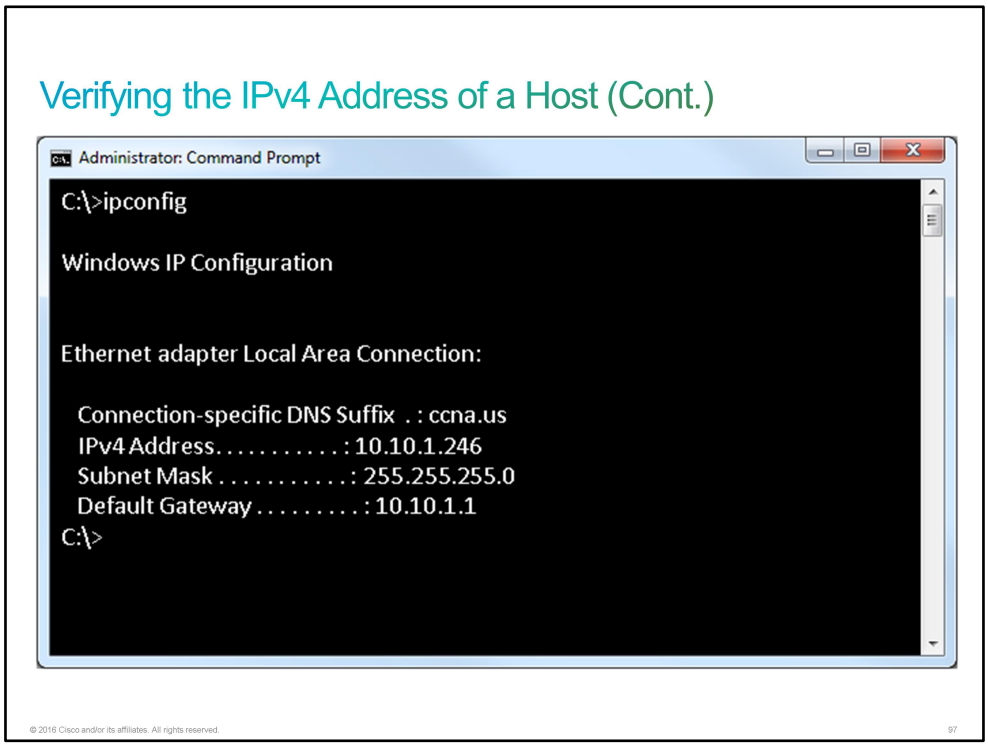
You use the **ipconfig** command to display all current TCP/IP network configuration values at the command line of a Windows computer. Using different command options, you can also use the **ipconfig** command to view and refresh DHCP and DNS settings. Used without command options, the **ipconfig** command displays the IP address, subnet mask, and default gateway for all adapters.

The following is the syntax for the **ipconfig** command:

```
ipconfig [/ all] [/ renew [adapter]] [/ release [adapter]] [/displaydns] [/flushdns]
```

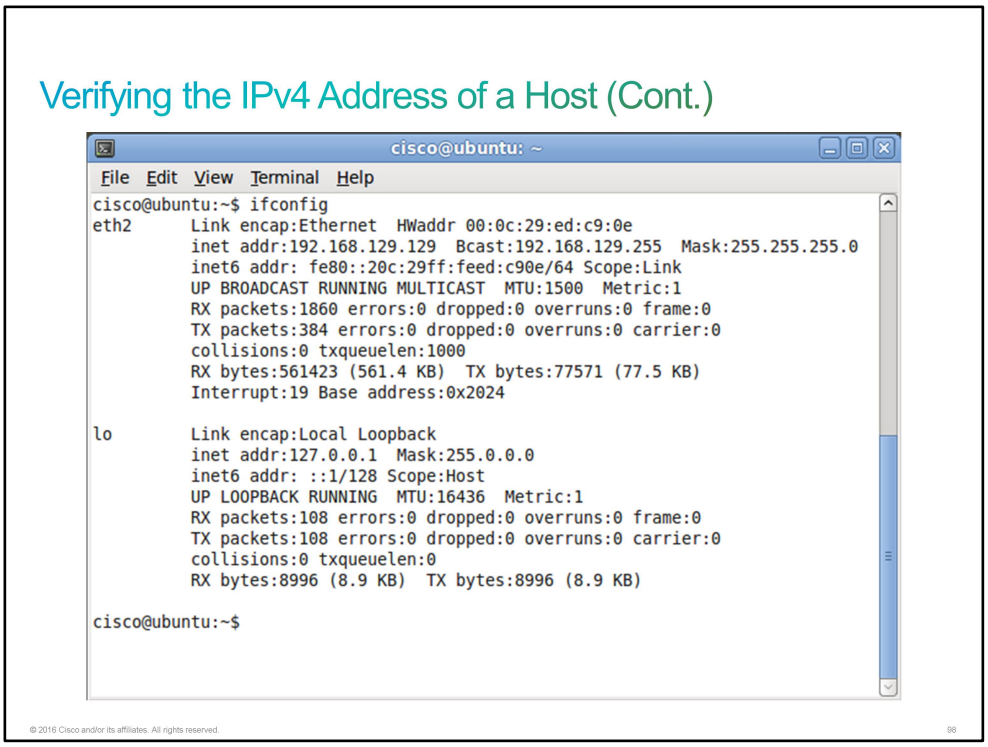

These command options are commonly used:

- **/all**—This option displays the complete TCP/IP configuration for all adapters, including DHCP and DNS configuration. Without this parameter, the **ipconfig** command displays only the IP address, subnet mask, and default gateway values for each adapter. Adapters can represent physical interfaces, such as installed network adapters, or logical interfaces, such as dial-up connections.
- **/renew [***adapter***]**—This option renews DHCP configuration for all adapters (if an adapter is not specified) or for a specific adapter if the adapter parameter is included. This parameter is available only on computers with adapters that are configured to obtain an IP address automatically. To specify an adapter name, enter the adapter name that appears when you use **ipconfig** without parameters.
- **/release** [*adapter*]—This option sends a DHCPRELEASE message to the DHCP server to release the current DHCP configuration and discard the IP address configuration for either all adapters (if an adapter is not specified) or for a specific adapter if the adapter parameter is included. This parameter disables TCP/IP for adapters that are configured to obtain an IP address automatically. To specify an adapter name, enter the adapter name that appears when you use **ipconfig** without parameters.
- **/displaydns**—This option displays the contents of the host DNS cache. When an IP host makes a DNS query for a hostname, it caches the result to avoid unnecessary queries.
- **/flushdns**—This option deletes the host DNS cache. This option is useful if the IP address that is associated with a hostname has changed, but the host is still caching the old IP address.
- **/?**—This option displays help at the command prompt.

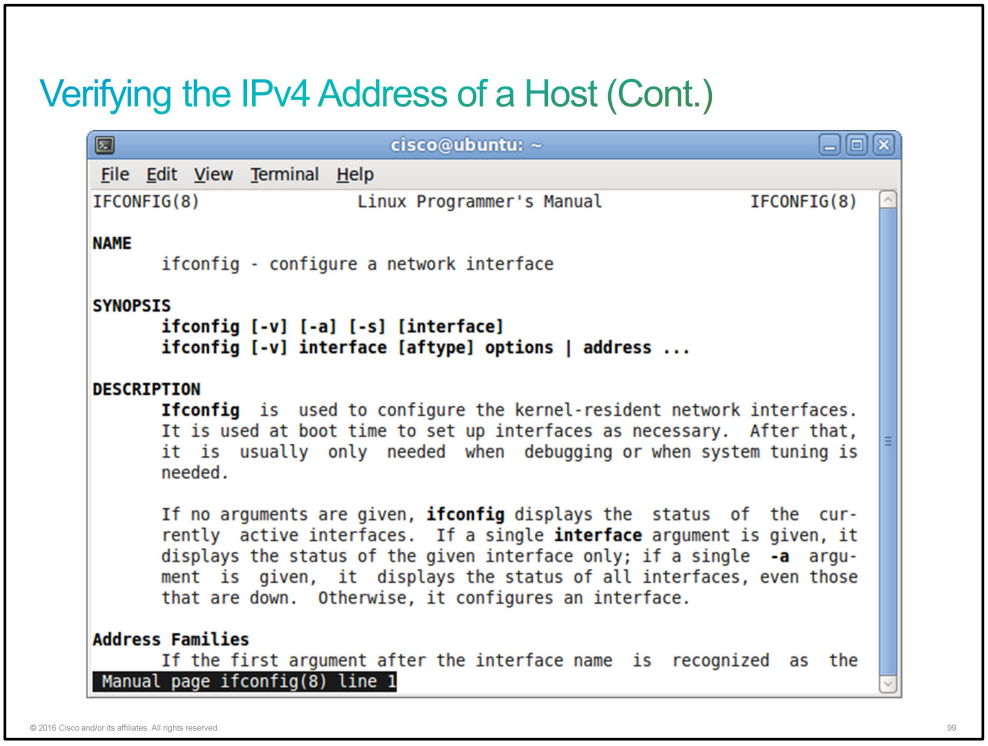


Note	For additional information about **ipconfig** and the command syntax, use your favorite search engine and search for this string: **microsoft technet dd197434 site:microsoft.com**

On most Linux operating systems, the **ifconfig** command is used to perform the same tasks that **ipconfig** performs on Microsoft Windows operating systems.



On Linux systems, you can get the details of specific syntax for just about any command using the **man** (manual) command. In this example, **man ifconfig** was entered:



Challenge

1. Where in the TCP/IP stack does IP operate?

   A. Internet layer
   B. application layer
   C. transport layer
   D. network access layer

2. What are two characteristics of a network ID from an IP address? (Choose two.)

   A. identifies the network of which the host is a part
   B. used by routers to maintain information about routes
   C. identifies the individual host
   D. assigned by organizations to individual devices
   E. identifies the individual subnet

3. What are two characteristics of a host ID from an IP address? (Choose two.)

   A. identifies the network of which the host is a part
   B. used by routers to maintain information about routes
   C. identifies the individual host
   D. assigned by organizations to individual devices
   E. identifies the individual subnet

4. What is the length of the source address in the IPv4 header?

   A. 8 bits
   B. 8 bytes
   C. 32 bytes
   D. 4 bytes

5. What is the correct binary representation of the IPv4 address 192.168.16.101?

   A. 11000000.10101000.00000100.01100101
   B. 11000000.00010000.10101000.01100101
   C. 11000000.10101000.00010000.01100101
   D. 11000000.10101000.00010000.00110101

6. What is the range of the Class C private IP addresses?

   A. 192.0.0.0 to 192.167.255.255
   B. 192.169.0.0 to 223.255.255.255
   C. 192.168.0.0 to 192.168.255.255
   D. 192.168.0.0 to 192.168.0.255

7. What is the range of the Class A public IP addresses?

A. 1.0.0.0 to 9.255.255.255, 11.0.0.0 to 126.255.255.255
B. 1.0.0.0 to 126.255.255.255
C. 0.0.0.0 to 10.255.255.255
D. 0.0.0.0 to 10.0.0.255

Answer Key

Challenge

1. A
2. A, B, E
3. C, D
4. D
5. C
6. C
7. A

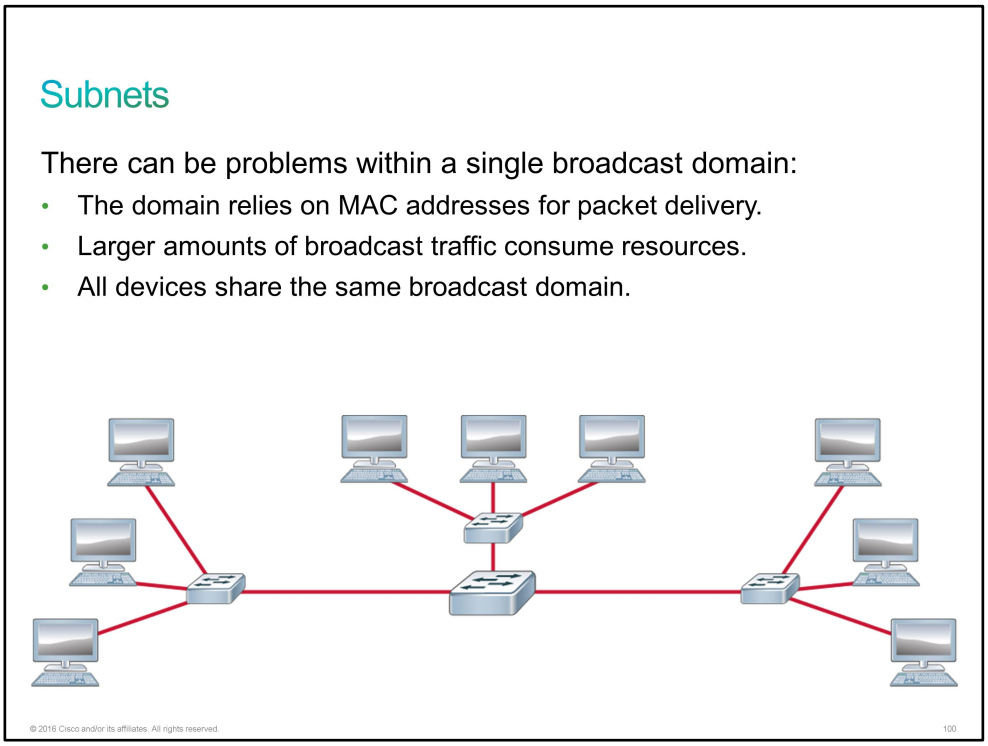
Lesson 2: Understanding IP Addressing and Subnets

Introduction

You did well demonstrating your knowledge on the Internet protocol. But before you are ready to go on deployments involving routers and Internet connectivity, you also need to demonstrate your knowledge of subnetting and VLSM.

Subnets

Network administrators often need to divide networks, especially large networks, into subnetworks, or subnets, to provide scalability.



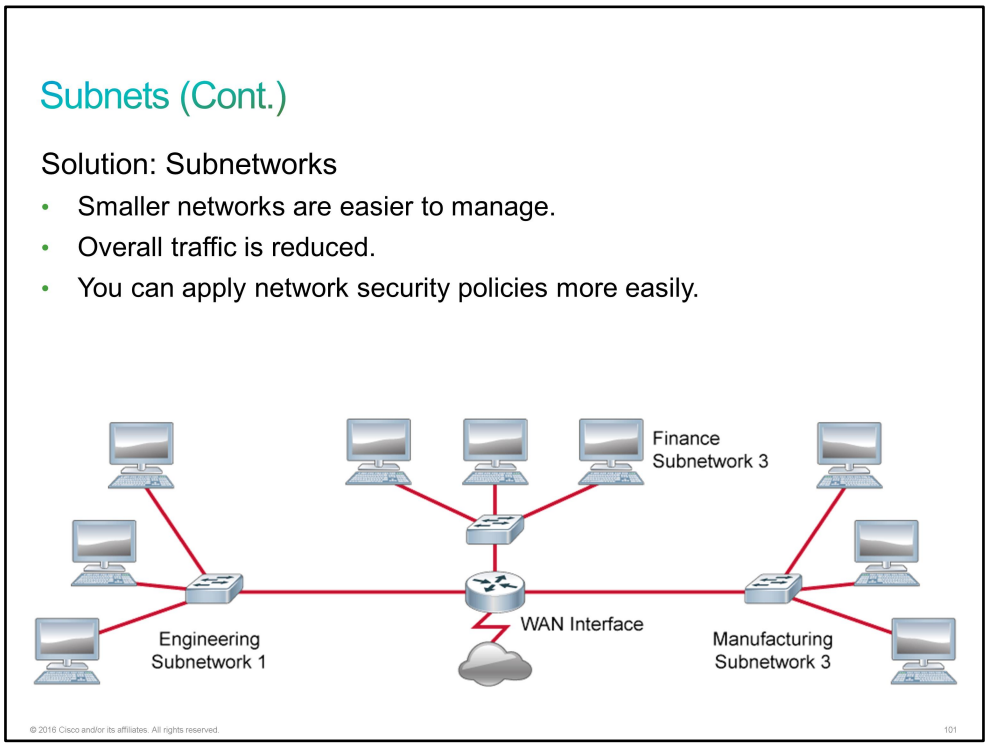
A company that occupies a three-story building might have a network that is divided by floors, with each floor divided into offices. Think of the building as the network, the floors as the three subnets, and the offices as the individual host addresses.

A subnet segments the hosts within the network. Without subnets, the network has a flat topology. A flat topology relies on MAC addresses to deliver packets. MAC addresses have no hierarchical structure. As the network grows, the use of the network bandwidth becomes less efficient.

There are other disadvantages to a flat network. All devices share the same broadcast domain, and it is difficult to apply security policies because there are no boundaries between devices.

Note	A broadcast domain is a network in which all devices can reach each other by broadcast.

On a switch-connected network, the host sees all the broadcasts in the broadcast domain. You can use routers to separate networks by breaking the network into multiple subnets or multiple broadcast domains.



The advantages of subnetting a network are as follows:

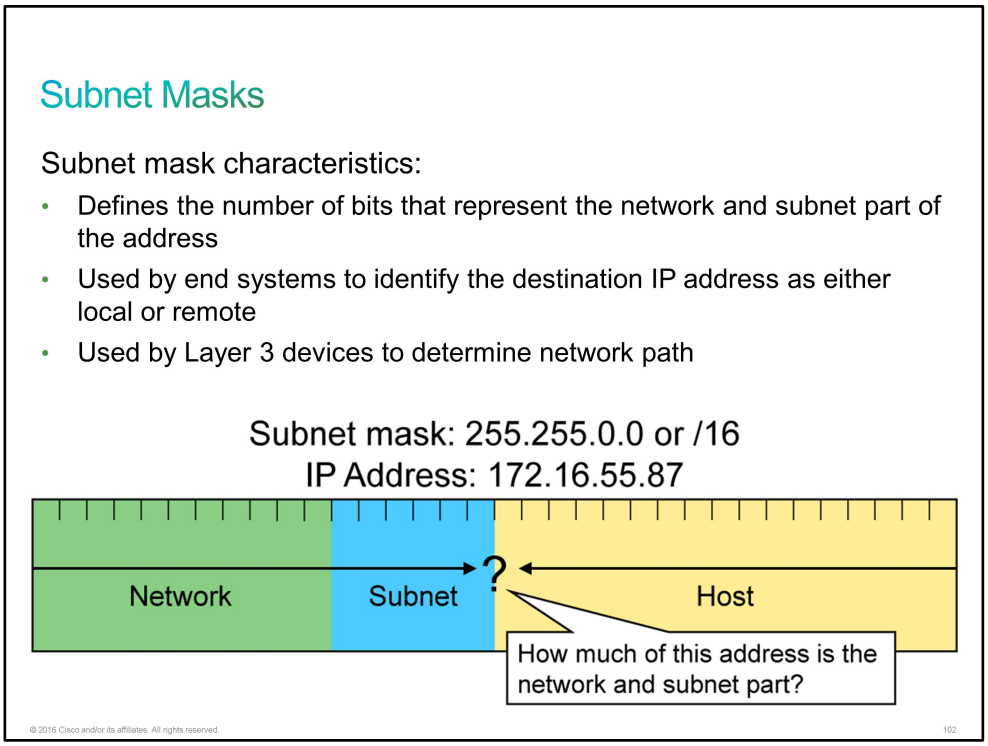
- Smaller networks are easier to manage and map to geographical or functional requirements.
- Subnetting enables you to create multiple logical networks from a single Class A, B, or C network address.
- Overall network traffic is reduced, which can improve performance.
- You can more easily apply network security measures at the interconnections between subnets than within a large single network.

In multiple-network environments, each subnetwork may be connected to the Internet by a single router. The figure shows one router connecting multiple subnetworks to the Internet. The details of the internal network environment and how the network is divided into multiple subnetworks are inconsequential to other IP networks.

The IP addressing that is used in the flat network must be modified to accommodate the required segmentation. A subnet mask identifies the network-significant portion of an IP address. The network-significant portion of an IP address is simply the part that identifies the network that the host device is on. This part is called the *network address* and defines every subnetwork. The use of segmentation is important for the routing operation to be efficient.

Subnet Masks

A subnet mask is a 32-bit combination that is used for routing traffic within a subnet. It describes which portion of an IP address refers to the subnet and which part refers to the host. As you already know, an IP address has two components; the network part and the host part. Subnetting enables the network administrator to further divide the host part. The first part identifies the subnetwork (subnet) to which are device belongs. The other part identifies the host.



How do you know how many bits represent the network portion of the address and how many bits represent the host portion? When you express an IPv4 network address, you add a prefix length to the network address. The prefix length is the number of bits in the address that give the network portion. For example, in 172.16.55.87 /20, /20 is the prefix length. It tells you that the first 20 bits are the network address. The remaining 12 bits, the last octet, is the host portion. The entity that is used to specify the network portion of an IPv4 address to the network devices is called the *subnet mask*. The subnet mask consists of 32 bits, just as the address does, and uses 1s and 0s to indicate which bits of the address are network bits and which bits are host bits. You express the subnet mask in the same dotted decimal format as the IPv4 address. The subnet mask is created by placing a binary 1 in each bit position that represents the network portion and placing a binary 0 in each bit position that represents the host portion. A /20 prefix is expressed as a subnet mask of 255.255.240.0 (11111111.11111111.11110000.00000000). The remaining bits (low order) of the subnet mask are zeroes, indicating the host address within the network.

The subnet mask is configured on a host with the IPv4 address to define the network portion of that address.

Networks are not always assigned the same prefix. Depending on the number of hosts on the network, the prefix that is assigned may be different. Having a different prefix number changes the host range and broadcast address for each network.

For example, take a look at the host 10.1.20.70/26:

- Address:
  - 10.1.20.70
  - 00001010.00000001.00010100.01000110
- Subnet mask:
  - 255.255.255.192
  - 11111111.11111111.11111111.11000000
- Network address:
  - 10.1.20.64
  - 00001010.00000001.00010100.01000000

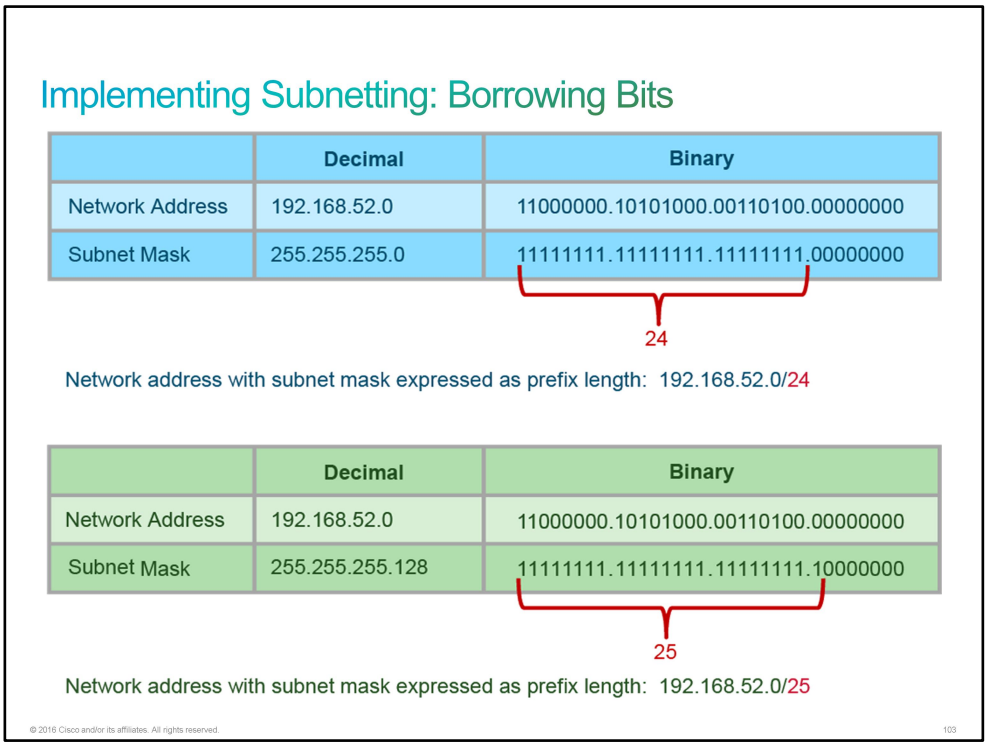
Implementing Subnetting: Borrowing Bits

To implement subnets, follow this procedure:

- Determine the IP address for your network as assigned by the registry authority.
- Based on your organizational and administrative structure, determine the number of subnets that are required for the network. Be sure to plan for growth.
- Based on the address class and required number of subnets, determine the number of bits that you need to borrow from the host ID.
- Determine the binary and decimal value of the new subnet mask that results from borrowing bits from the host ID.
- Apply the subnet mask to the network IP address to determine the subnet and host addresses. Also determine the network and broadcast addresses for each subnet.
- Assign subnet addresses to specific interfaces for all devices that are connected to the network.

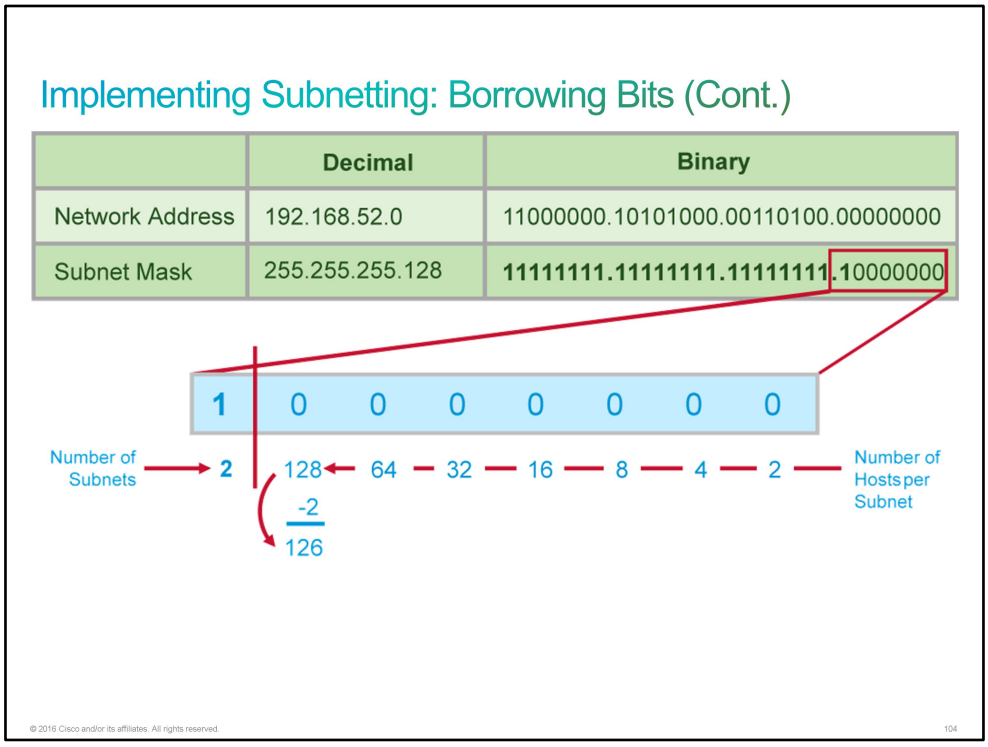
To subnet a network address, you must borrow host bits and use them as subnet bits, as the following figure shows. This approach ignores the boundaries of classful networks, on which the predefined classes of IP addresses (A, B, and C) are based, and introduces the classless network. Bits must be borrowed consecutively, starting with the first host bit on the left.

Take a look at the following figure. The top table shows a standard Class C network address that is not subnetted. The bottom table shows the same address after it is subnetted by borrowing one host bit. Notice that the prefix length has changed from 24 to 25. The network IP address itself is unchanged, although it is now considered a subnetwork (subnet) and is one of two subnets that have been created. The subnet mask has changed from 255.255.255.0 in decimal to 255.255.255.128 because the 128 bit is now turned on in the last octet.

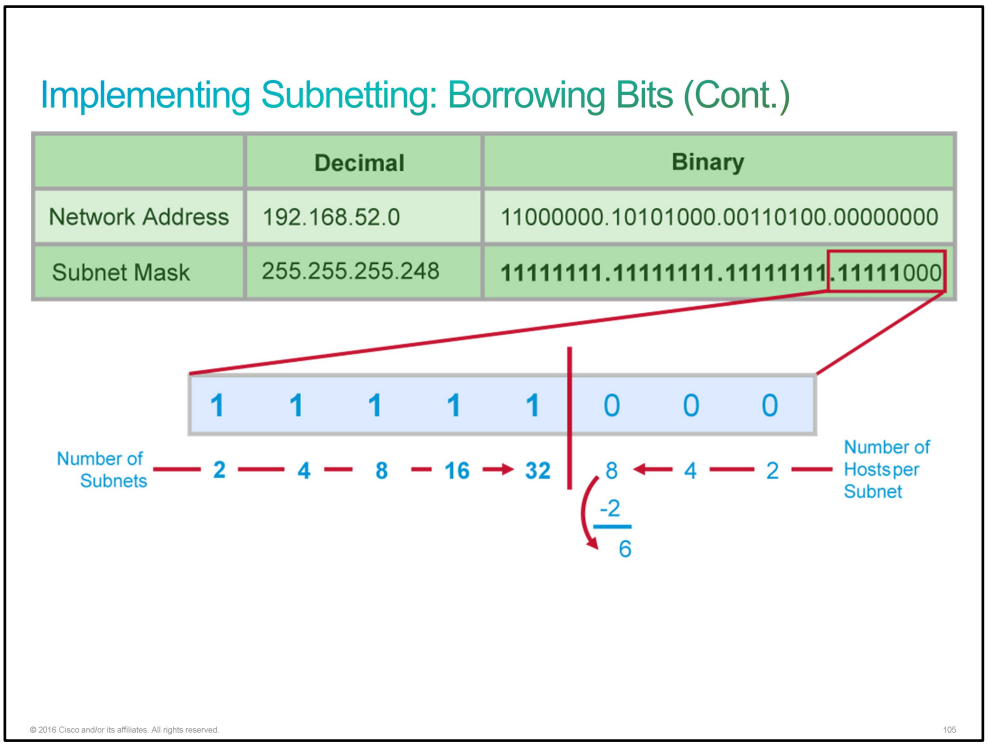


Each time that a bit is borrowed, the number of subnet addresses increases, and the number of host addresses that are available per subnet decreases. The algorithm that is used to compute the number of subnets and hosts uses powers of two. Therefore, borrowing one host bit enables you to create 2^1 (2) subnets, borrowing 2 bits gives you 2^2 (4) subnets, and so on.

As the following figure shows, you can also determine how many host addresses are available when you borrow a given number of bits by counting by powers of two. Starting with the far-right host bit, begin with 2^1 (2) and increase by powers of two. Then, subtract two. The figure shows that borrowing 1 bit for subnetting the address in the example leaves 7 bits for hosts. The formula to determine the number of hosts for this example is $2^7 – 2$, which calculates to 126 host addresses per subnet.



Here is another example, in which five host bits are borrowed for subnetting. In this example, 32 subnets are created, and only 6 host addresses are available for each subnet. The new subnet mask is 11111111.11111111.11111111.11111000, which equates to 255.255.255.248 in decimal.



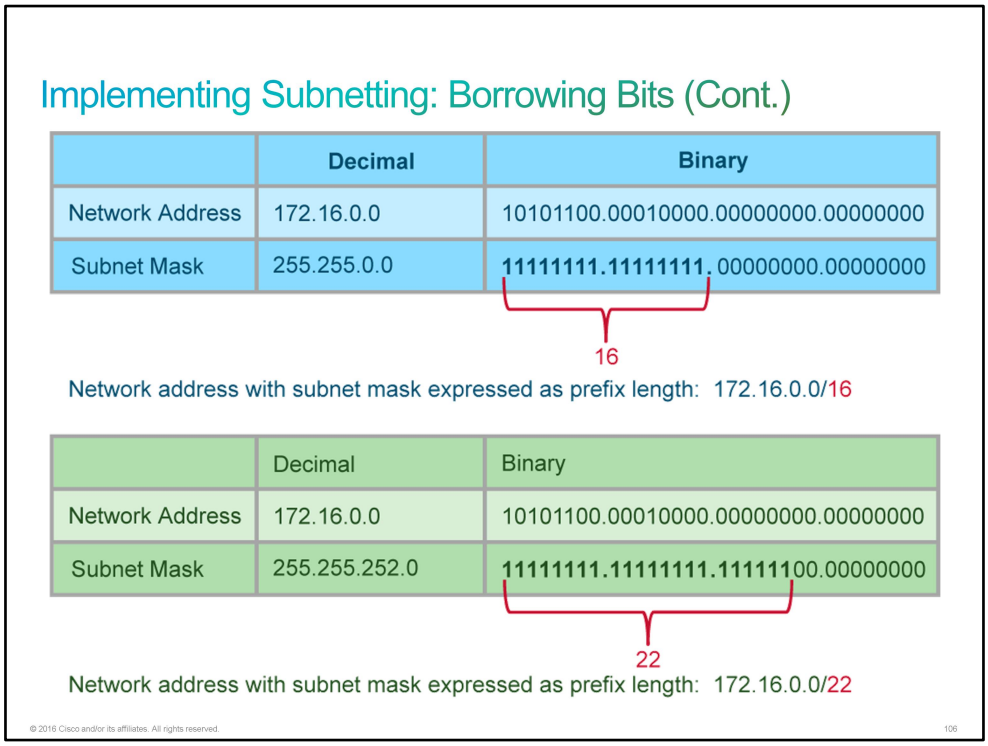
You can use the following formula to calculate the number of subnets that are created by borrowing a given number of host bits:

Number of subnets = 2^s (where *s* is the number of bits borrowed)

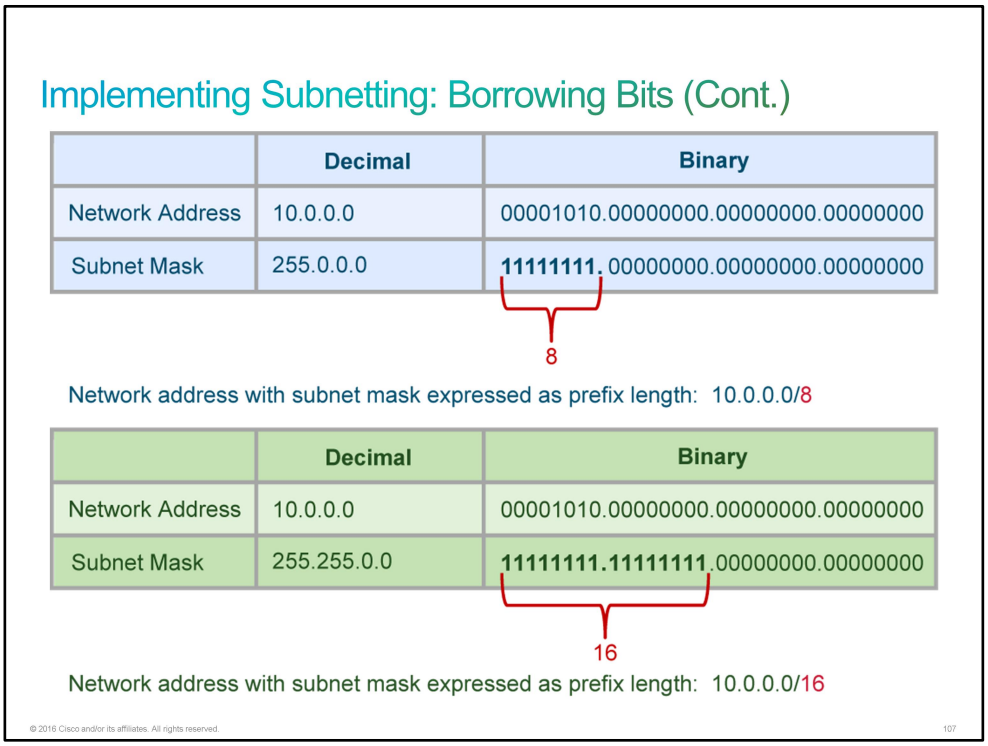
You can use a similar formula to calculate the number of host addresses that are available when a given number of host bits are borrowed:

Number of hosts = 2^h (where *h* is the number of host bits remaining after bits are borrowed) – 2

The following figure shows the subnetting of a Class B network address. The top table shows a network address with the default Class B subnet mask, 255.255.0.0. The second table shows the same address after it is subnetted by borrowing six host bits. Notice that the prefix length has changed from 16 to 22. The network IP address itself is unchanged, but the subnet mask has changed from 255.255.0.0 in decimal to 255.255.252.0.



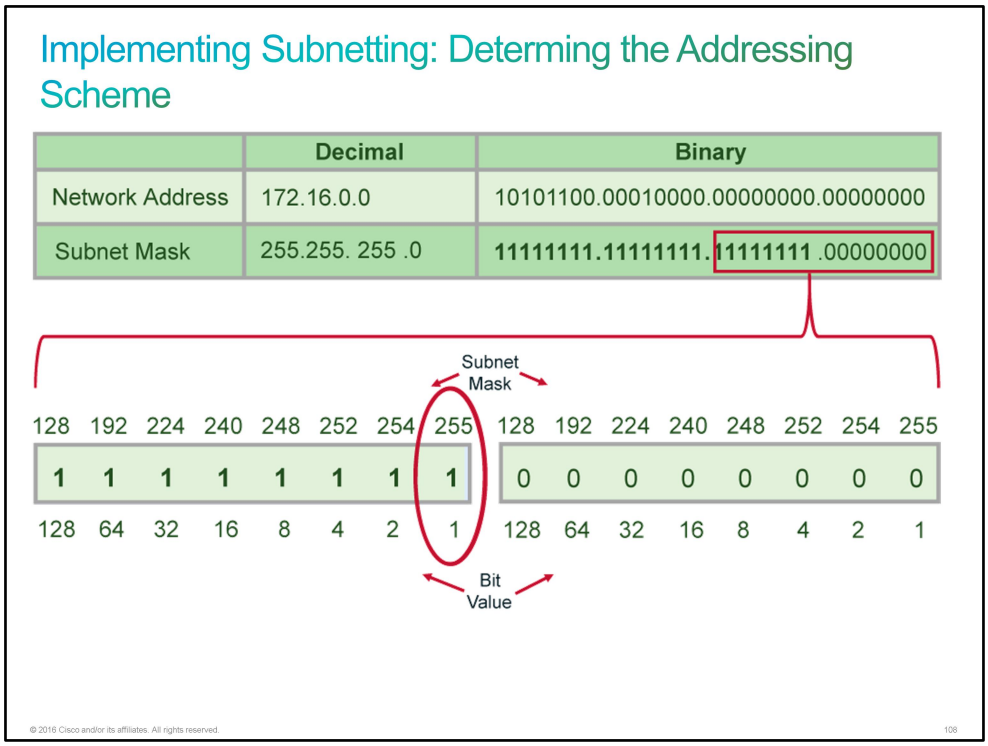
The next figure shows the subnetting of a Class A network address. The top table shows a network address with the default Class A subnet mask, 255.0.0.0. The bottom table shows the same address after it is subnetted by borrowing 8 host bits. Notice that the prefix length has changed from 8 to 16. The network IP address itself is unchanged, but the subnet mask has changed from 255.0.0.0 in decimal to 255.255.0.0.



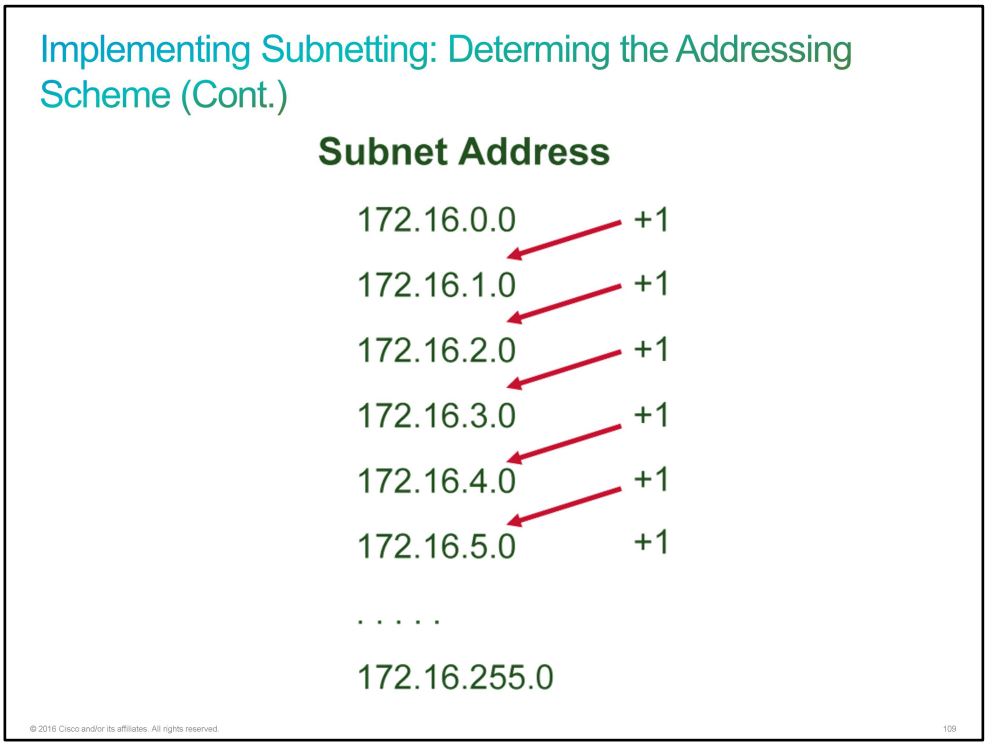
Implementing Subnetting: Determing the Addressing Scheme

If a network address is subnetted, the first subnet that is obtained after subnetting the network address is called subnet zero. To determine each subsequent subnet address, increase the network address by the bit value for the last bit that you borrowed.

In the following example, 8 bits are borrowed for subnetting the same network address, 172.16.0.0. The first subnet address is 172.16.0.0, the zero subnet. The last bit borrowed is the bit with the value of 1, so the next subnet address is 172.16.1.0.



The following figure shows the first six subnets and the last subnet that are created by borrowing the 8 bits.

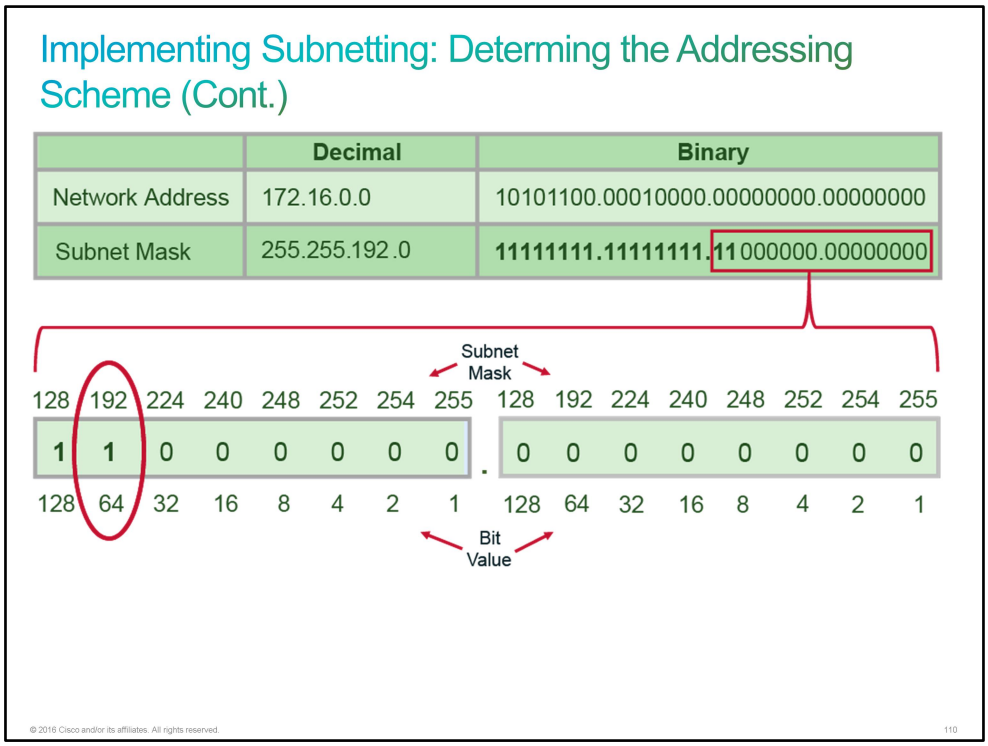


Here are the host addresses and broadcast addresses for those subnets.

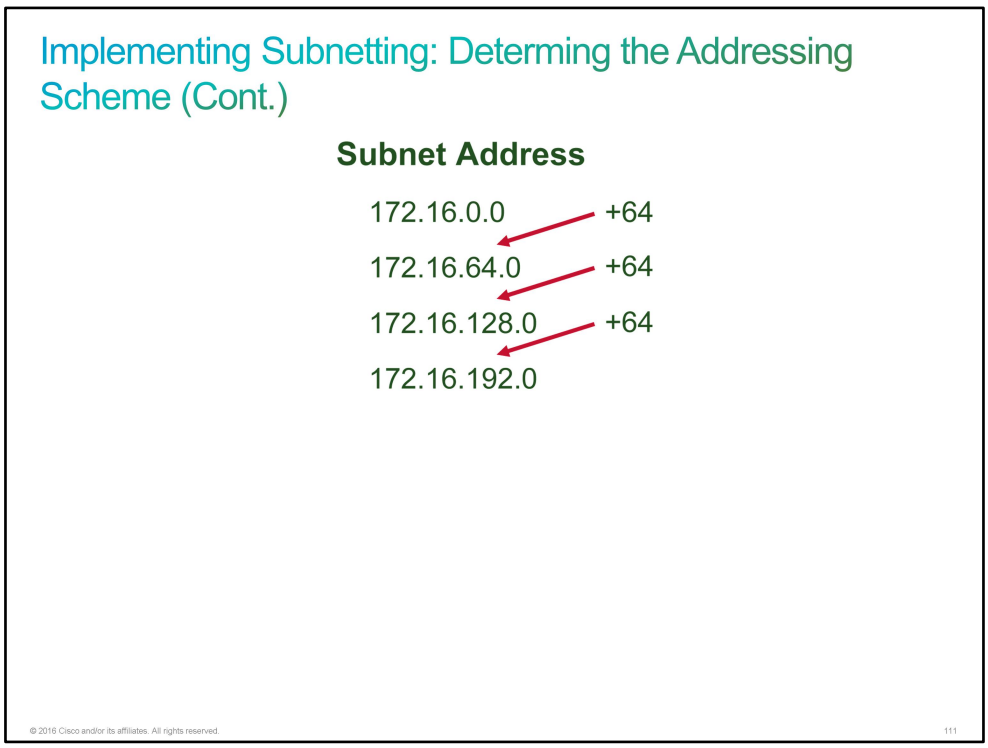
Host Addresses and Broadcast Addresses

Subnet Address	Host Address Range	Broadcast Address
172.16.0.0	172.16.0.1–172.16.0.254	172.16.0.255
172.16.1.0	172.16.1.1–172.16.1.254	172.16.1.255
172.16.2.0	172.16.2.1–172.16.2.254	172.16.2.255
172.16.3.0	172.16.3.1–172.16.3.254	172.16.3.255
172.16.4.0	172.16.4.1–172.16.4.254	172.16.4.255
172.16.5.0	172.16.5.1–172.16.5.254	172.16.5.255
...	...	...
172.16.255.0	172.16.255.1–172.16.255.254	172.16.255.255

In the following figure, Class B network address 172.16.0.0 has been subnetted by borrowing two host bits. The first subnet address is again 172.16.0.0, the zero subnet. The last bit borrowed is the bit with the value of 64, so the next subnet address is 172.16.64.0



The following figure shows all the subnets that are created by borrowing the 2 bits. The subnet 172.16.192.0 is the last subnet because 192 + 64 = 256, and the highest possible value for any given octet is 255.

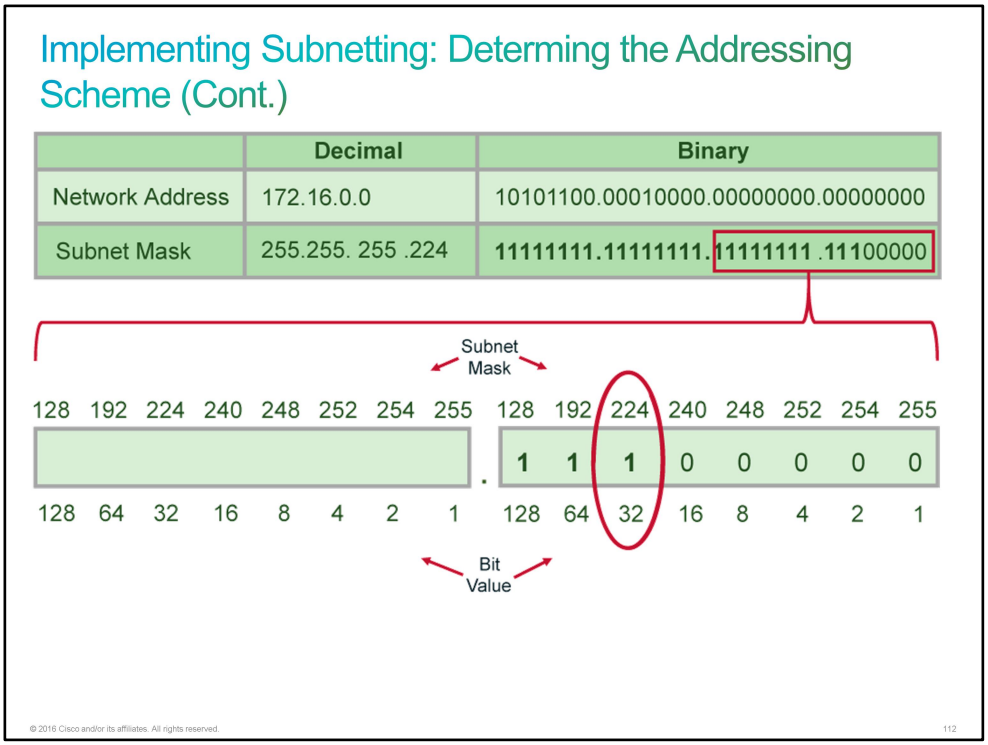


The following table shows the valid host addresses for each subnet that was created by borrowing 2 bits. You may recall the following statement from a previous topic: "You can determine how many host addresses are available when you borrow a given number of bits by counting by powers of two. Starting with the far-right host bit, begin with 2^1 (2) and increase by powers of two. Then subtract two." The reason that you have to subtract two is to allow for a broadcast address for each subnet and to avoid using the subnet address as a host address. The table shows the valid host IP address range for each network that you examined in the previous practice question.

Valid Host Addresses for Each Subnet Created by Borrowing 2 Bits

Subnet Address	Valid Host Address Range	Broadcast Address
172.16.0.0	172.16.0.1–172.16.63.254	172.16.63.255
172.16.64.0	172.16.64.1–172.16.127.254	172.16.127.255
172.16.128.0	172.16.128.1–172.16.191.254	172.16.191.255
172.16.192.0	172.16.192.1–172.16.255.254	172.16.255.255

Here is one more example of subnetting the same Class B network address, this time borrowing 11 host bits for subnetting. Again, the first subnet address is 172.16.0.0. The second subnet address is 172.16.0.32 because the last bit borrowed has a value of 32. Notice that this time, the last borrowed bit is in the fourth octet. Therefore, the increment of 32 (the value of the last borrowed bit) is applied in the fourth octet.



The following table shows the first 10 subnet addresses and the last subnet address (with the corresponding host addresses and broadcast addresses) that result from subnetting Class B network 172.16.0.0 by borrowing 11 host bits.

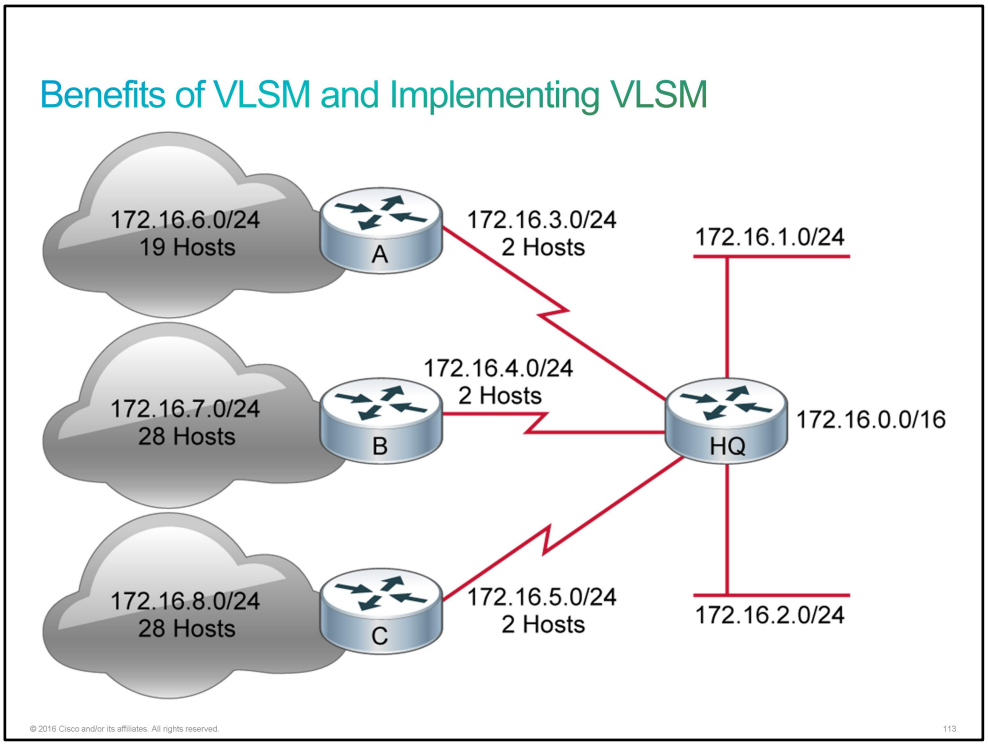
Subnet Address	Host Address Range	Broadcast Address
172.16.0.0	172.16.0.1–172.16.0.30	172.16.0.31
172.16.0.32	172.16.0.33–172.16.0.62	172.16.0.63
172.16.0.64	172.16.0.65–172.16.0.94	172.16.0.95
172.16.0.96	172.16.0.97–172.16.0.126	172.16.0.127
172.16.0.128	172.16.0.129–172.16.0.158	172.16.0.159
172.16.0.160	172.16.0.161–172.16.0.190	172.16.0.191
172.16.0.192	172.16.0.193–172.16.0.222	172.16.0.223
172.16.0.224	172.16.0.225–172.16.0.254	172.16.0.255
172.16.1.0	172.16.1.1–172.16.1.30	172.16.1.31
172.16.1.32	172.16.1.33–172.16.1.62	172.16.1.63
...	...	...
172.16.255.224	172.16.255.225–172.16.255.254	172.16.255.255

Benefits of VLSM and Implementing VLSM

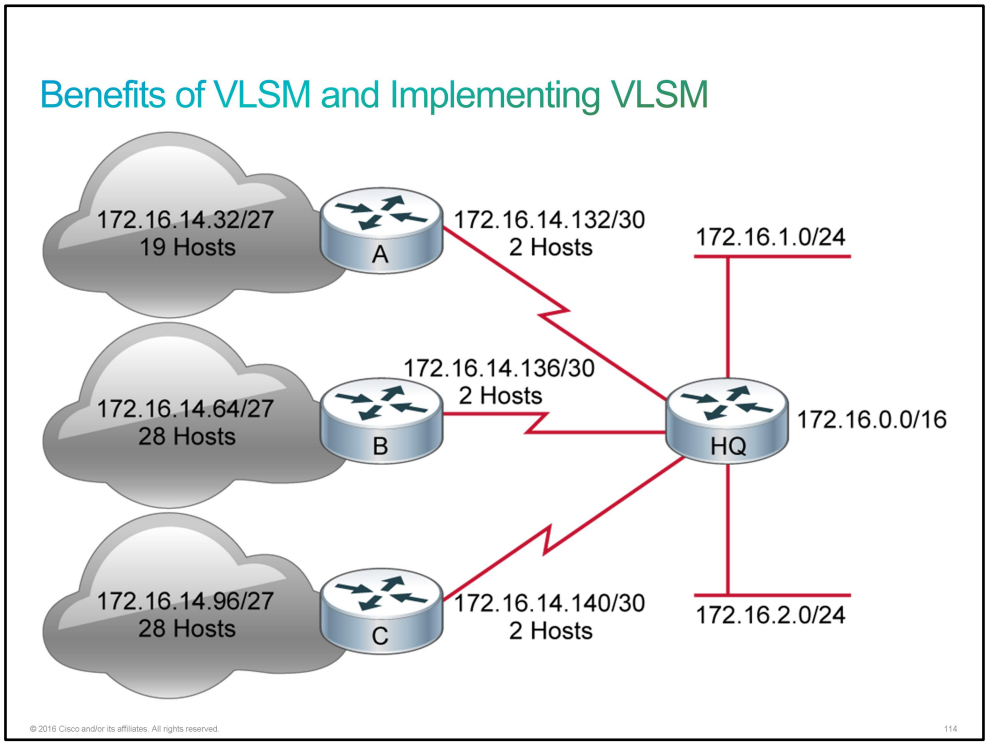
In all the subnetting examples in the previous topics, the same subnet mask was applied for all the subnets. This way, each subnet had the same number of available host addresses. You may need this approach sometimes, but, usually, having the same subnet mask for all subnets ends up wasting address space.

For example, in the following figure, Class B network 172.16.0.0 is subnetted by borrowing 8 host bits and applying a 24-bit subnet mask, which allows for 256 subnets with 254 host addresses each. In this example, many host addresses are wasted. Each WAN link needs only two host addresses, so 252 host addresses are wasted on each WAN link. Many host addresses are also wasted on other subnets. VLSM provides a solution.

VLSM allows you to include more than one subnet mask within a network to achieve more efficient use of IP addresses. Instead of using the same subnet mask for all subnets, you can use the most efficient subnet mask for each subnet. The most efficient subnet mask for a subnet is the mask that provides an appropriate number of host addresses for that individual subnet. For example, subnet 172.16.6.0 has only 19 hosts, so it does not need the 254 host addresses that the 24-bit mask allows. A 27-bit mask would provide 30 host addresses, which is much more appropriate for this subnet.



In the next figure, the 172.16.0.0/16 network is again divided into subnetworks using a 24-bit subnet mask. However, one of the subnetworks in this range, 172.16.14.0/24, is further divided into smaller subnetworks using a 27-bit mask to accommodate the subnets that have 19 or 28 hosts. These smaller subnetworks range from 172.16.14.0/27 to 172.16.14.224/27. Then, one of these smaller subnets, 172.16.14.128/27, is further divided using a 30-bit mask, which creates subnets with only two hosts to be used on the WAN links. The subnets with the 30-bit mask range from 172.16.14.128/30 to 172.16.14.156/30.



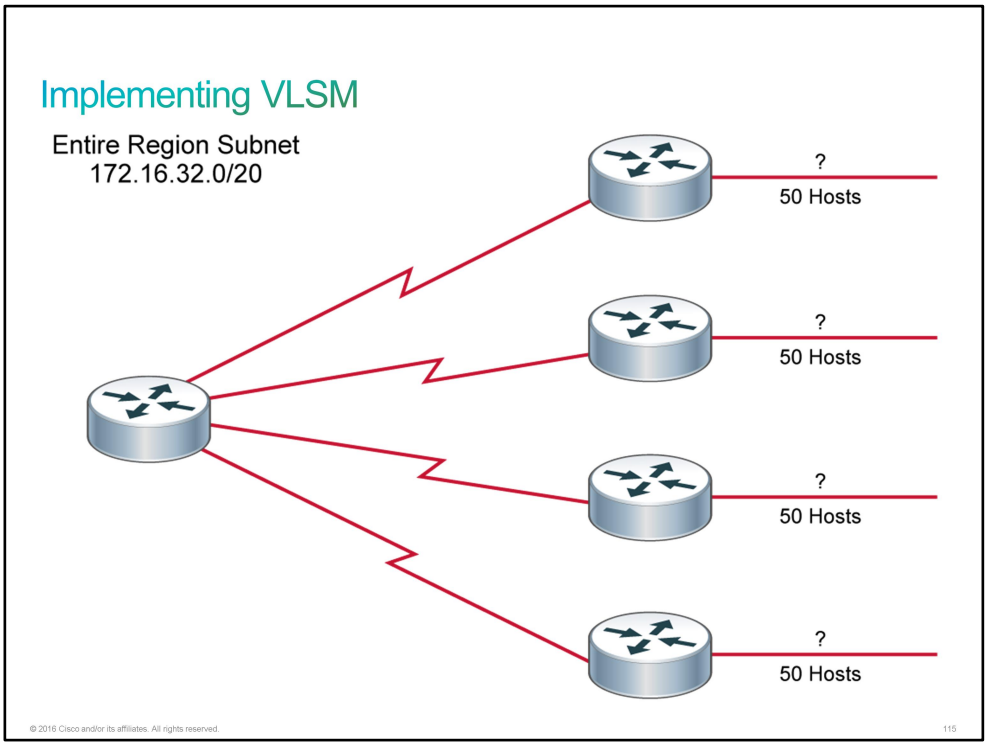
In addition to providing a solution to the problem of wasted IP addresses, VLSM has another important benefit: support for route summarization, which is also called route aggregation. The hierarchical addressing design of VLSM enables easier summarization of network addresses. Route summarization reduces the number of routes in routing tables by representing a range of network subnets in a single summary address. Smaller routing tables require less CPU time for routing lookups.

In the previous figure, the subnet 172.16.14.0/24 describes all the addresses that are further subnets of 172.16.14.0, including those addresses from subnet 172.16.14.0/27 to subnet 172.16.14.128/30.

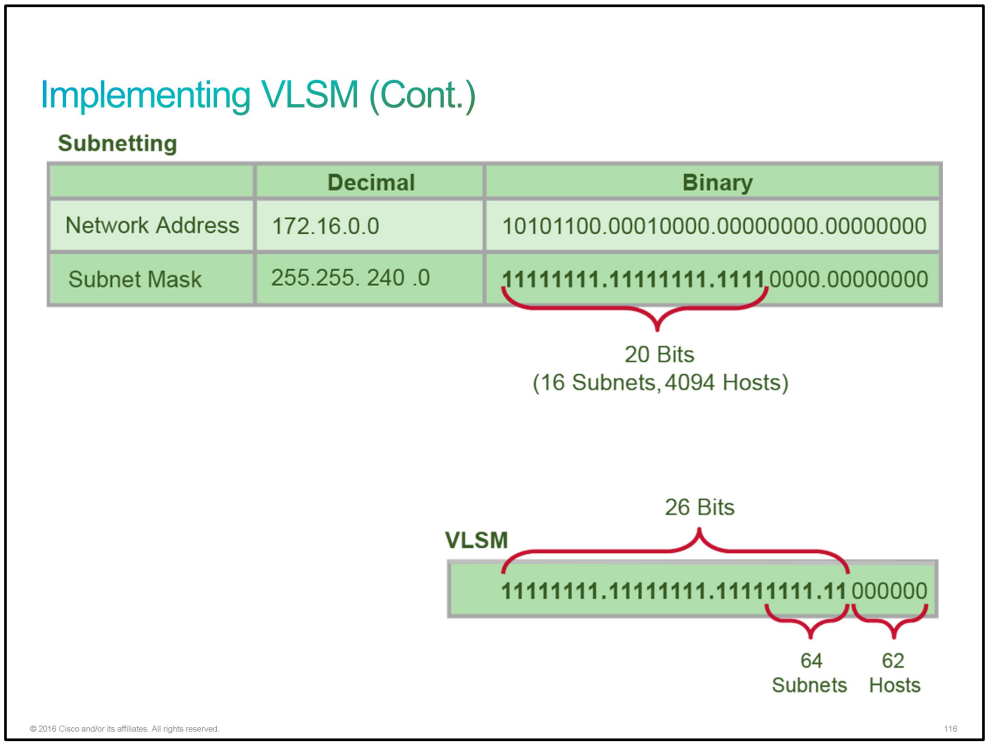
VLSM is an important technology in large routed networks. It can only be used in networks that run routing protocols that support VLSM. These protocols include RIPv2, OSPF, and EIGRP.

Implementing VLSM

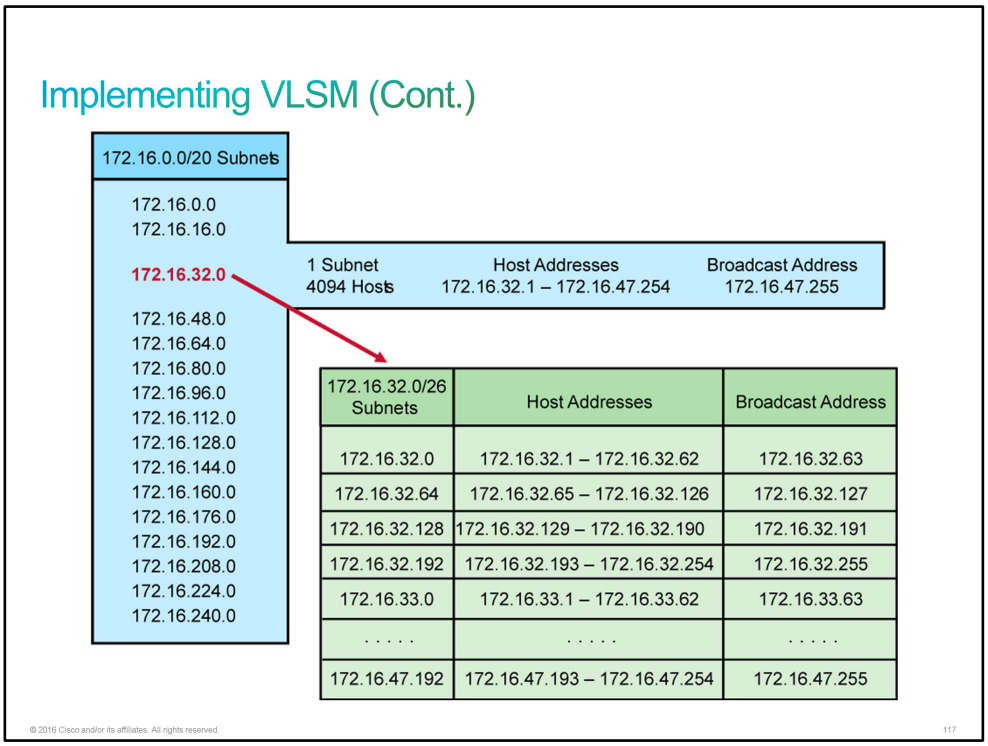
The network 172.16.0.0 has already been subnetted by applying a 20-bit subnet mask. One of the resulting subnet addresses, 172.16.32.0/20, is used for the region of the enterprise network that the following figure shows. This region needs to assign addresses to multiple LANs. Each LAN must have 50 hosts. You can use VLSM to further subnet the address 172.16.32.0/20 to give you more subnet addresses with fewer hosts per subnet.



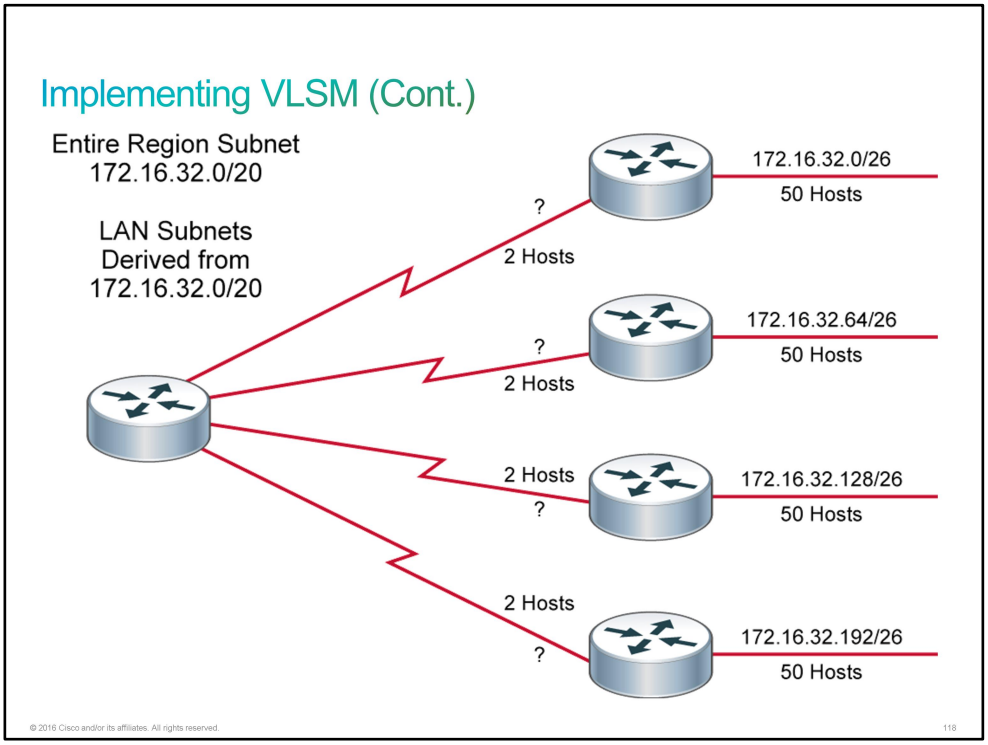
The next figure shows in binary the original subnetting of the 172.16.0.0/16 network by borrowing 4 host bits, which provided 16 subnets with 4094 host addresses each. The figure also shows how further subnetting with VLSM increases the number of subnets and provides the desired number of host addresses per subnet. Counting the additional host bits that are borrowed by powers of 2 from left to right shows that 64 subnets are created. Counting the host bits by powers of 2 from right to left and then subtracting 2 shows that 62 host addresses are available per subnet.



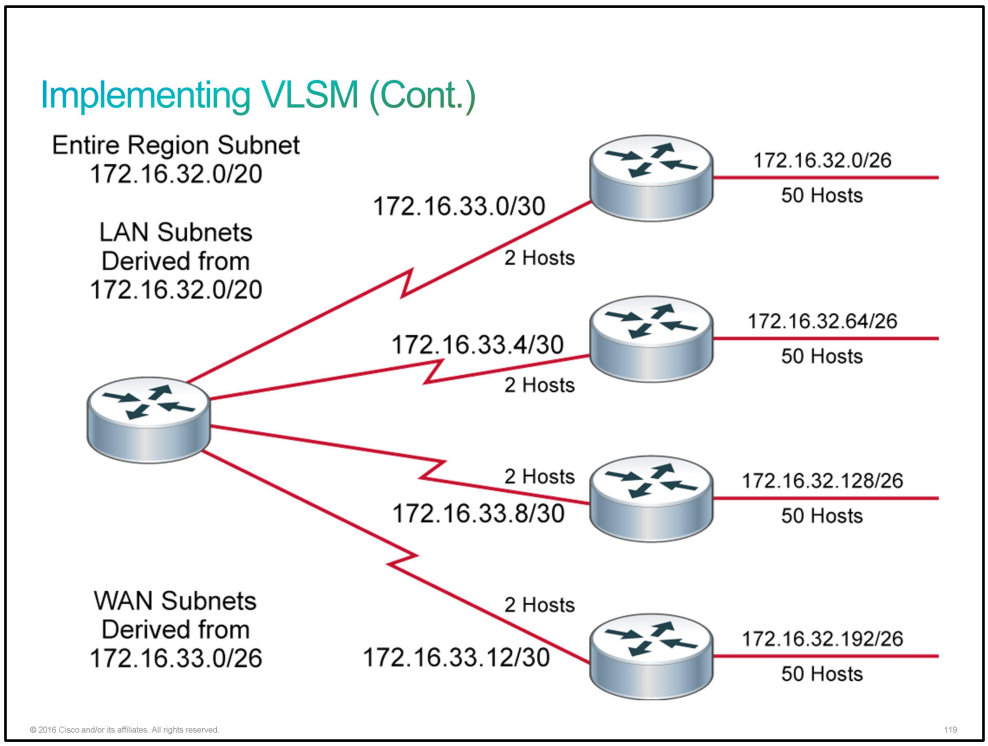
The next figure shows the subnet addresses and host addresses that are achieved by using VLSM. The subnet for the region in this example, subnet 172.16.32.0/20, is further subnetted by applying a 26-bit mask as the previous figure shows.



The following figure shows some of the new VLSM subnet addresses applied to the regional network.



To calculate the subnet addresses for the WAN links, further subnet one of the unused /26 subnets with a 30-bit subnet mask. For this example, subnet 172.16.33.0 will be further subnetted. The 30-bit subnet mask provides 16 (2^4) subnets with 2 ($2^2 - 2$) host addresses each.



As seen in this example, the easiest way to assign the subnets is to assign the largest first.

Challenge

1. What do subnetworks provide?

   A. scalability
   B. reachability
   C. redundancy
   D. load balancing

2. Which two aspects can present a problem with a single broadcast domain? (Choose two.)

   A. Larger amounts of broadcast traffic consume resources.
   B. All devices share the same broadcast domain.
   C. The domain relies on IP addresses for packet delivery.
   D. Larger amounts of multicast traffic consume resources.
   E. All PCs share the same broadcast domain.

3. Which two statements about a network that uses subnetworks are true? (Choose two.)

   A. It is more complex to apply network security policies.
   B. Smaller networks are easier to manage.
   C. Overall traffic is increased.
   D. Smaller networks are harder to manage.
   E. Overall traffic is reduced.

4. What is the decimal equivalent of the binary number 11000000?

   A. 224
   B. 240
   C. 128
   D. 192

5. You have subnetted your 192.168.36.0 network address with a 255.255.255.240 mask. How many usable subnets and hosts per subnet are available?

   A. 2 usable subnets and 126 hosts per subnet
   B. 4 usable subnets and 62 hosts per subnet
   C. 8 usable subnets and 30 hosts per subnet
   D. 16 usable subnets and 14 hosts per subnet
   E. 32 usable subnets and 6 hosts per subnet

6. How many valid host addresses are available for each subnet after subnetting network 192.168.0.0 by borrowing 2 host bits?

   A. 62
   B. 4094
   C. 8190
   D. 16382

7. What does VLSM stand for?

   A. Virtual LAN Subnet Mask
   B. Variable LAN Subnet Mask
   C. Virtual LAN Same Mask
   D. Variable Length Subnet Mask

Answer Key

Challenge

1. A
2. A, B
3. B, E
4. D
5. D
6. A
7. D

Lesson 3: Understanding the TCP/IP Transport Layer

Introduction

Bob notifies you via email that to join CCS, you will also need to demonstrate your understanding of TCP/IP transport layer functionality. Bob wants to validate that you know the differences between TCP and UDP, and common applications that use TCP and UDP as a transport.

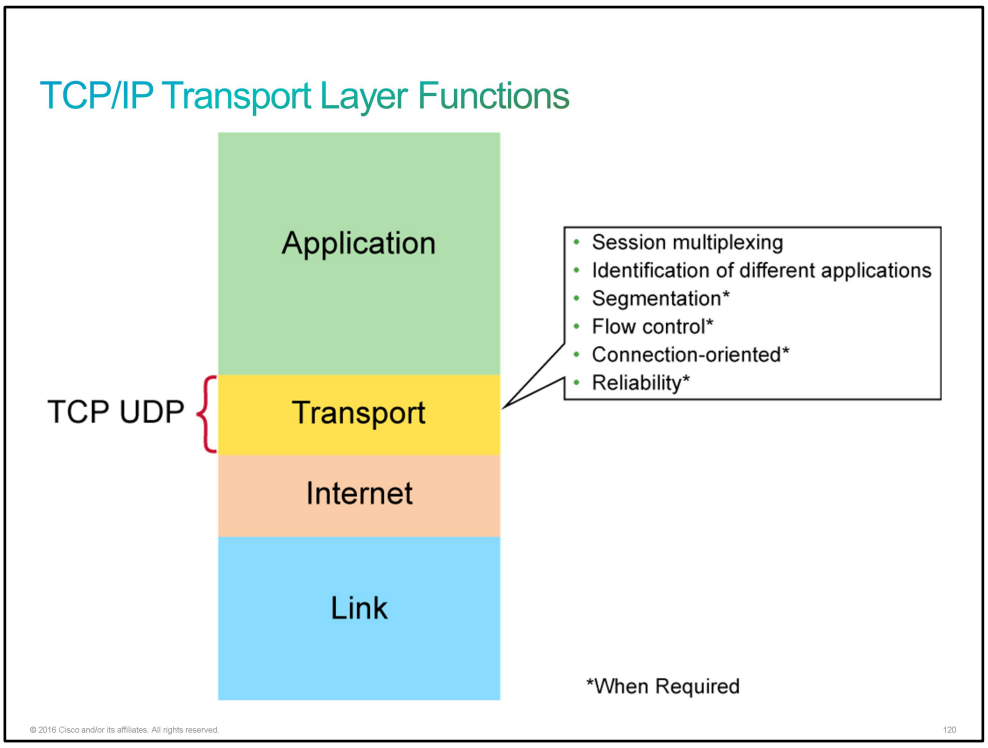
You will first start with the transport layer functions, and contrast reliable and unreliable transport. Compare TCP and UDP side by side. You will need to explain to Bob basic characteristics of UDP and describe its header. You will also discuss with him the TCP three-way handshake.

TCP/IP Transport Layer Functions

Residing between the application and Internet layers of the TCP/IP protocol stack, the transport layer is fundamental to the operation of the TCP/IP layered network architecture. The TCP/IP Internet layer directs information to its destination, but it cannot guarantee that the information will arrive in the correct order, free of errors, or even that it will arrive at all. The two most common transport layer protocols of the TCP/IP protocol suite are TCP and UDP. Both protocols manage the communication of multiple applications and provide communication services directly to the application process on the host.

The basic service that the transport layer provides is tracking individual communication between applications on the source and destination hosts. This service is called session multiplexing, and it is performed by both UDP and TCP. A major difference between TCP and UDP is that TCP can ensure that the data is delivered, while UDP does not.

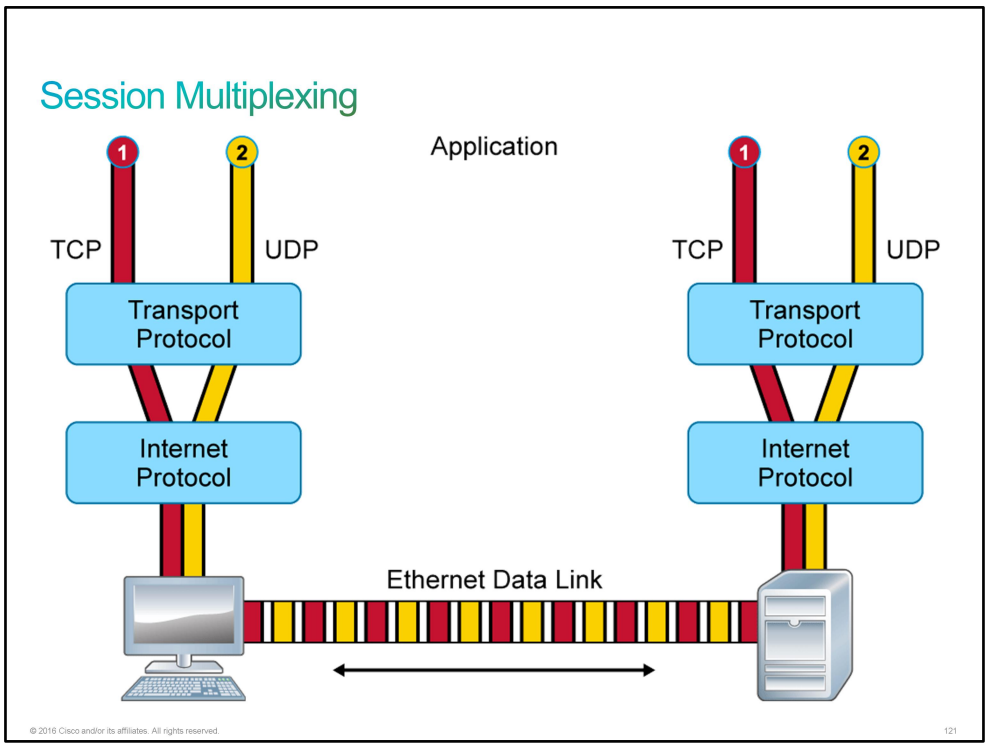
Note	Review of OSI and TCP/IP reference models: The transport layer of the TCP/IP protocol stack maps to the transport layer of the OSI model. The protocols that operate at this layer are said to operate at Layer 4 of the OSI model. If you hear someone use the term "Layer 4," they are referring to the transport layer of the OSI model.



Multiple communications often occur at once; for instance, you may be searching the web and using FTP to transfer a file at the same time. The transport tracks these communications and keeps them separate. This tracking is provided by both UDP and TCP. To pass data to the proper applications, the transport layer must identify the target application. If TCP is used, the transport layer has the additional responsibilities of establishing end-to-end operations, segmenting data and managing each piece, reassembling the segments into streams of application data, managing flow control, and applying reliability mechanisms.

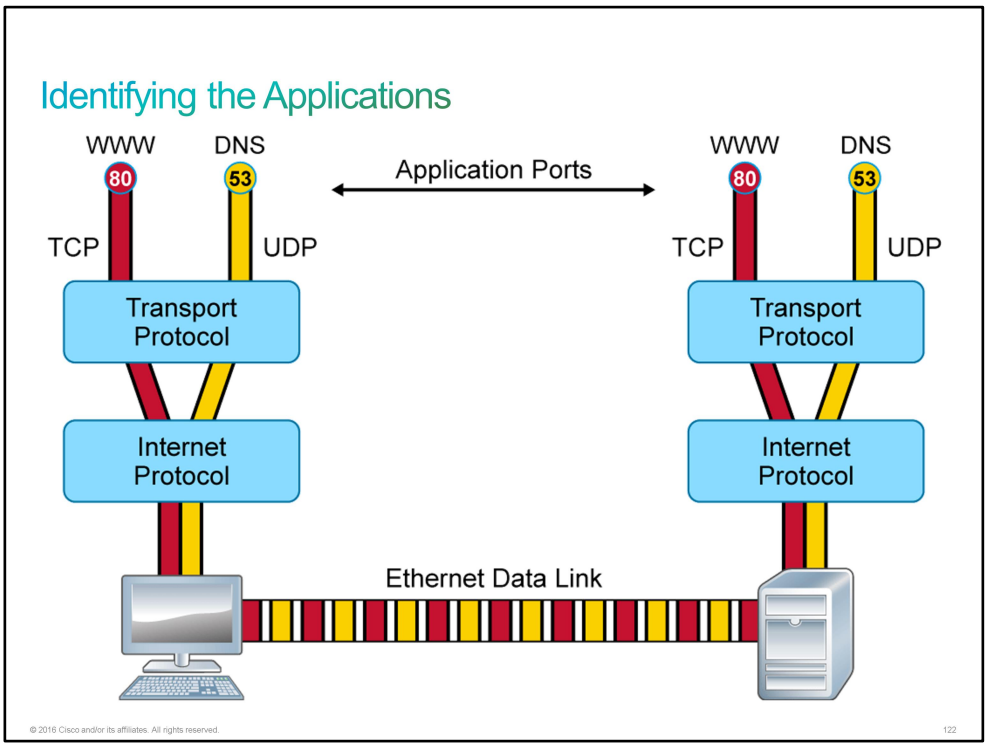
Session Multiplexing

Session multiplexing is the process by which an IP host is able to support multiple sessions simultaneously and manage the individual traffic streams over a single link. A session is created when a source machine needs to send data to a destination machine. Most often, this process involves a reply, but a reply is not mandatory.



Identifying the Applications

To pass data to the proper applications, the transport layer must identify the target application. TCP/IP transport protocols use port numbers to accomplish this task. Each application process that needs to access the network is assigned a port number (often called an application identifier) that is unique in that host. The port number is used in the transport layer header to indicate which application that piece of data is associated with.



Segmentation

TCP takes arbitrarily sized data chunks from the application layers and prepares them for transport onto the network. The application relies on TCP to ensure that each chunk is broken up into smaller segments that will fit the MTU of the underlying network layers. UDP does not provide segmentation services. UDP instead expects the application process to perform any necessary segmentation and supply it with data chunks that do not exceed the MTU of lower layers.

Note	The MTU of the IP protocol is 1500 bytes. Larger MTUs are possible, but 1500 bytes is the normal size.

Flow Control

If a sender transmits packets faster than the receiver can receive them, the receiver drops some of the packets and requires them to be retransmitted. TCP is responsible for detecting dropped packets and sending replacements. A high rate of retransmissions introduces latency in the communication channel. To reduce the impact of retransmission-related latency, flow control methods work to maximize the transfer rate and minimize the required retransmissions.

Basic TCP flow control relies on acknowledgments that are generated by the receiver. For every data chunk that is sent, the sender waits for this acknowledgment from the receiver before sending the next part. However, if the RTT is significant, the overall transmission rate may slow to an unacceptable level. To increase network efficiency, a mechanism called *windowing* is combined with basic flow control. Windowing allows a receiving computer to advertise how much data it is able to receive before transmitting an acknowledgment to the sending computer.

Windowing allows avoidance of congestion in the network.

Connection-Oriented Transport Protocol

Within the transport layer, a connection-oriented protocol establishes a session connection between two IP hosts and then maintains the connection during the entire transmission. When the transmission is complete, the session is terminated. The TCP protocol provides connection-oriented reliable transport for application data.

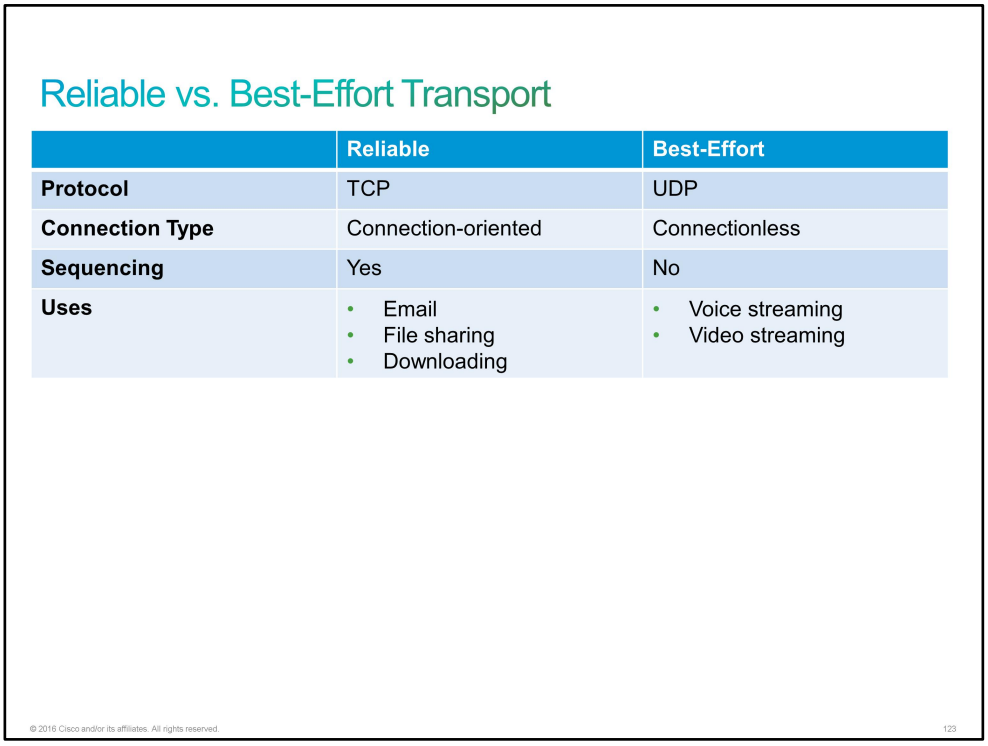
Reliability

TCP reliability has these three main objectives:

- Detection and retransmission of dropped packets
- Detection and remediation of duplicate or out-of-order data
- Avoidance of congestion in the network

Reliable vs. Best-Effort Transport

The terms *reliable* and *best effort* are terms that describe two types of connections between computers. TCP is a connection-oriented protocol that is designed to ensure reliable transport, flow control, and guaranteed delivery of IP packets. For this reason, it is labeled a "reliable" protocol. UDP is a connectionless protocol that relies on the application layer for sequencing and detection of dropped packets and is considered "best effort." Each protocol has strengths that make them useful for particular applications.



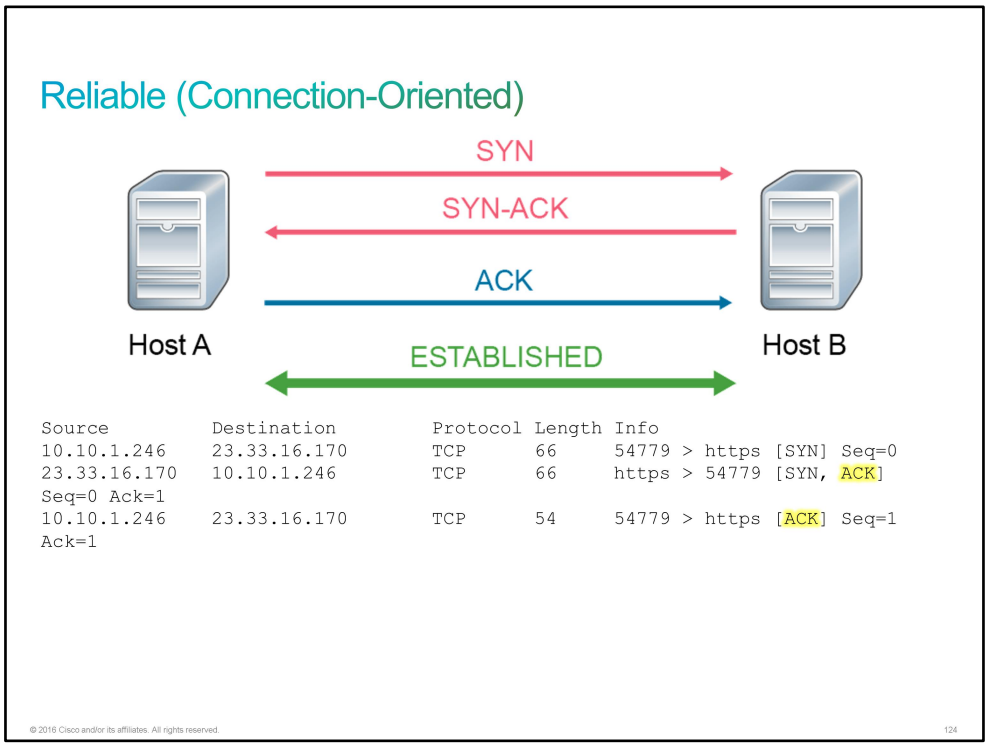
Reliable (Connection-Oriented)

Some types of applications require a guarantee that packets arrive safely and in order. Any missing packets could cause the data stream to be corrupted. Consider the example of using your web browser to download an application. Every piece of that application must be assembled on the receiver in the proper binary order, or it will not execute. File transfer is an application where the use of a connection-oriented protocol like TCP is indicated.

Note	TCP employs a three-way handshake that is initiated by the IP host that is making a connection to an application. For more information on TCP session setup, please use your favorite search engine to locate a copy of RFC 793 (Transmission Control Protocol).

TCP uses a three-way handshake when setting up a connection. You can think of it as being similar to a phone call. The phone rings, the called party says "hello," and the caller says "hello." Here are the actual steps:

1. The source of the connection sends a SYN packet to the destination requesting a session. The Sequence Number (or SN) in this case is zero.
2. The destination responds to the SYN with a SYN-ACK and increments the initiator SN by 1.
3. If the source accepts the SYN-ACK, it sends an ACK packet to complete the handshake.



Here you can see some common applications that use TCP:

- Web browsers
- Email
- FTP
- Network printing
- Database transactions

To support reliability, a connection is established between the IP source and destination to ensure that the application is ready to receive data. During the initial process of connection establishment, information is exchanged about the capabilities of the receiver, and starting parameters are negotiated. These parameters are then used for tracking data transfer during the connection.

When the sending computer transmits data, it assigns a sequence number to each packet. The receiver then responds with an acknowledgment number that is equal to the next expected sequence number. This exchange of sequence and acknowledgment numbers allows the protocol to recognize when data has been lost, or duplicated, or has arrived out of order.

Best Effort (Connectionless)

Reliability (guaranteed delivery) is not always necessary (or even desirable). For example, if one or two segments of a video stream fail to arrive, it would only create a momentary disruption in the stream. This disruption might appear as a momentary distortion of the image, but the user may not notice. In real-time applications, such as voice and video streaming, dropped packets can be tolerated as long as the overall percentage of dropped packets is low.

Here you can see some common applications that use UDP:

- DNS
- Streaming video
- VoIP
- TFTP

UDP provides applications with best-effort delivery and does not need to maintain state information about previously sent data. As a benefit, UDP does not need to establish any connection with the receiver and is termed connectionless. There are many situations in which best-effort delivery is more desirable than reliable delivery. A connectionless protocol is desirable for applications that require faster communication without verification of receipt.

TCP vs. UDP Analogy

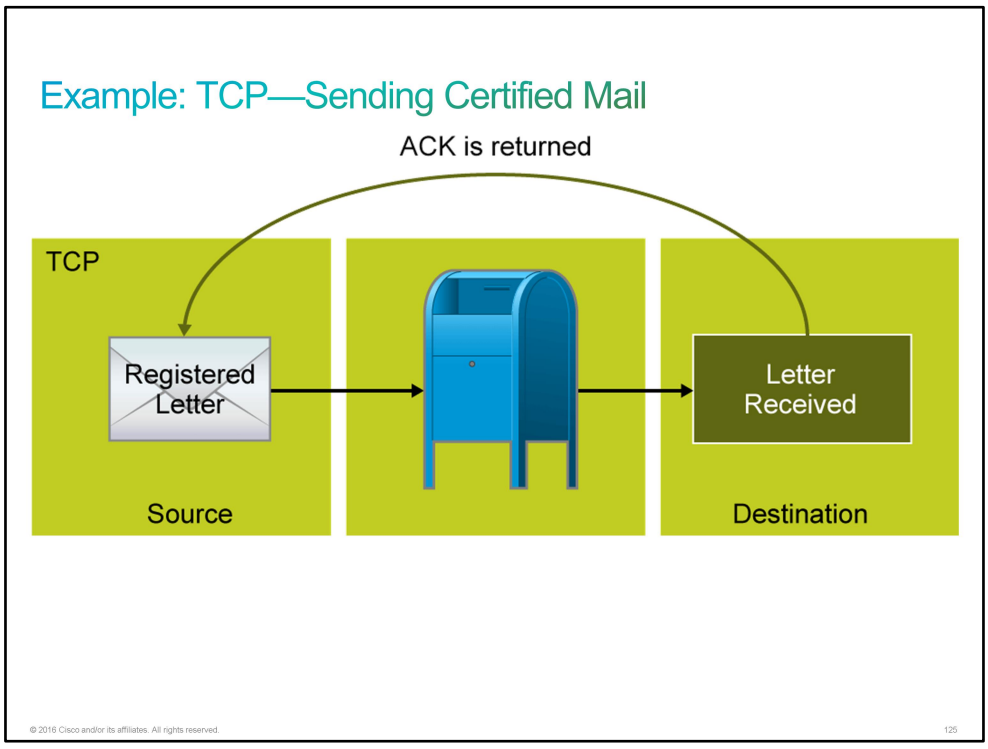
The postal service has been used as an analogy to illustrate the differences between connection-oriented TCP and connectionless services that UDP provides.

Example: TCP—Sending Certified Mail

Imagine that you are a popular author in Seattle. Your editor in Indianapolis is very anxious to publish your next novel and demands that you mail her each page as you finish one. You print each page of the book as you write them and put each page in a separate envelope. To ensure that your editor reassembles the book correctly, you put a page number on each envelope (a sequence number). You address the envelope and send the first one as certified mail. The postal service delivers it by any truck and any route, but because it is certified, the carrier who delivers it must get a signature from your editor and return a certificate of delivery to you.

Your contract with the publisher specifies that each page must be in a separate envelope. But having to go to the post office to send each letter individually is too time-consuming, so you send several envelopes together. The postal service again delivers each envelope by any truck and any route. Your editor signs a separate receipt for each envelope in the batch as she receives them. If one envelope is lost in transit, you will not receive a certificate of delivery for that numbered envelope, and you will need to resend that page. As your editor is receiving your envelopes, she uses the sequence numbers to assemble the book in the proper order.

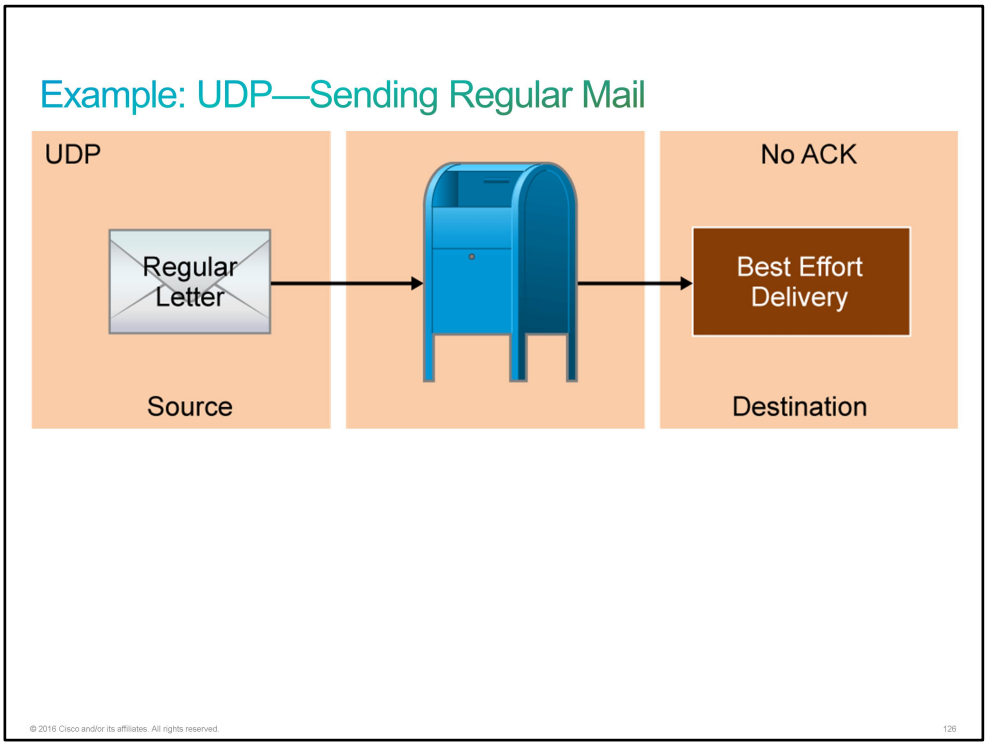
Like certified mail, TCP offers sequencing, acknowledgements, and retransmission.



Example: UDP—Sending Regular Mail

UDP services can be compared to using the postal service to pay your bills. You address each bill payment to a specific company address, stamp the envelope, and include your return address. The postal service guarantees its best effort to deliver each payment. The postal service does not guarantee delivery, and it is not responsible for telling you that delivery was successful or unsuccessful.

Like standard mail, UDP is a simple process that provides basic data-transfer services.

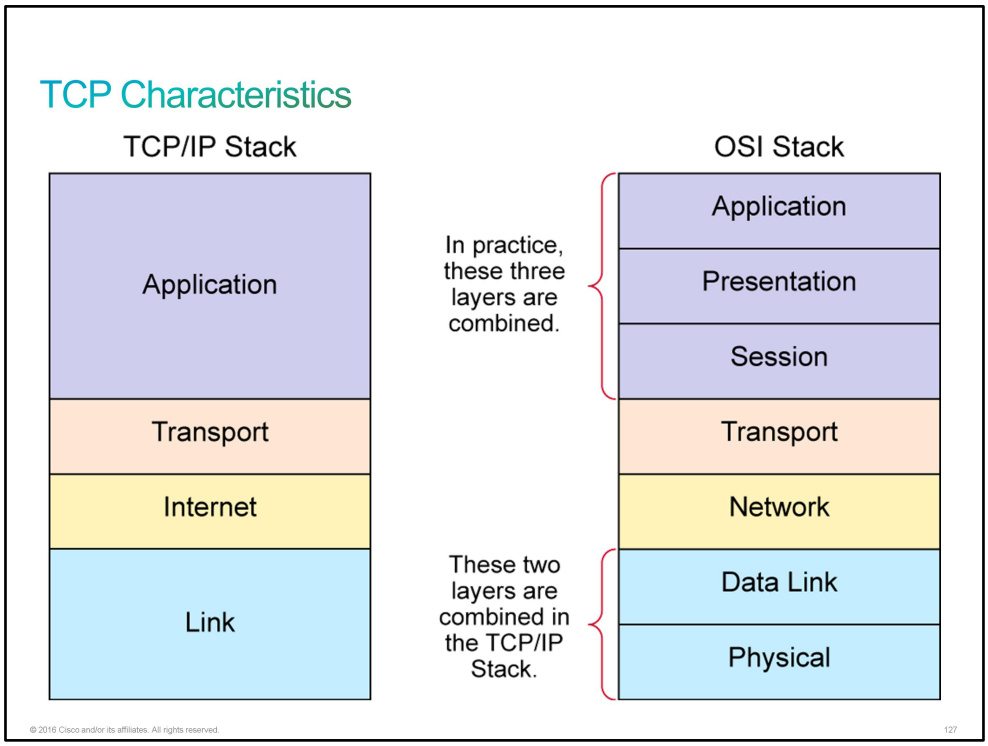


TCP Characteristics

TCP is a core protocol in the TCP/IP protocol suite. Applications leverage the connection-oriented services of TCP to provide data reliability between hosts. TCP includes several important features that provide for reliable data transmission.

TCP can be characterized as follows:

- TCP operates at the transport layer (OSI Layer 4) of the TCP/IP stack.
- TCP provides application access to the network layer (OSI Layer 3), where application data is routed from the source IP host to the destination IP host.



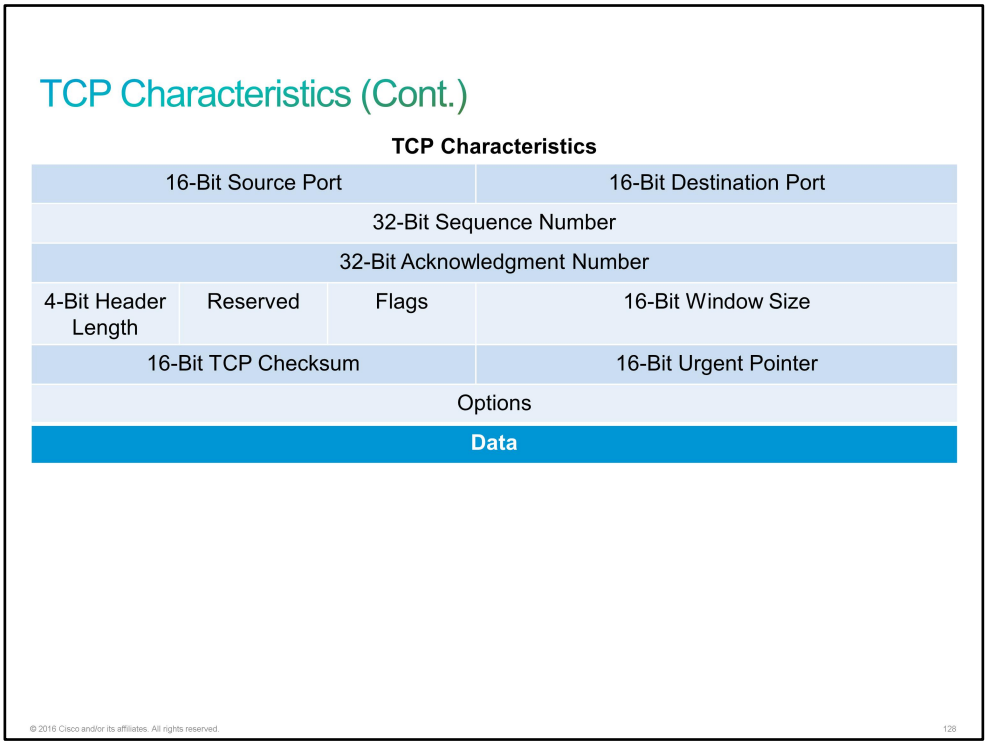
- TCP is connection-oriented and requires that network devices set up a connection to exchange data. The end systems synchronize with one another to manage packet flows and adapt to congestion in the network.
- TCP provides error checking by including a checksum in the IP datagram to verify that the TCP header information is not corrupt.
- A TCP connection is a pair of virtual circuits, one in each direction, so it operates in the full-duplex mode.
- TCP segments are numbered and sequenced so that the destination can reorder segments and determine if data is missing.
- Upon receipt of one or more TCP segments, the receiver returns an acknowledgment to the sender to indicate that it received the segment. Acknowledgments form the basis of reliability within the TCP session. When the source receives acknowledgment, it knows that the data has been successfully delivered. If the source does not receive acknowledgment within a predetermined period, it retransmits that data to the destination. The source may also terminate the connection if it determines that the receiver is no longer on the connection.

- TCP provides recovery services in which the receiver can request retransmission of a segment.
- TCP provides mechanisms for flow control. Flow control assists the reliability of TCP transmission by adjusting the effective rate of data flow between the two services in the session.

Reliable data delivery services are critical for applications such as file transfers, database services, transaction processing, and other mission-critical applications in which delivery of every packet must be guaranteed. TCP segments are sent by using IP packets. The TCP header follows the IP header and supplies information that is specific to the TCP protocol. Flow control, reliability, and other TCP characteristics are achieved by using fields in the TCP header. Each field has a specific function.

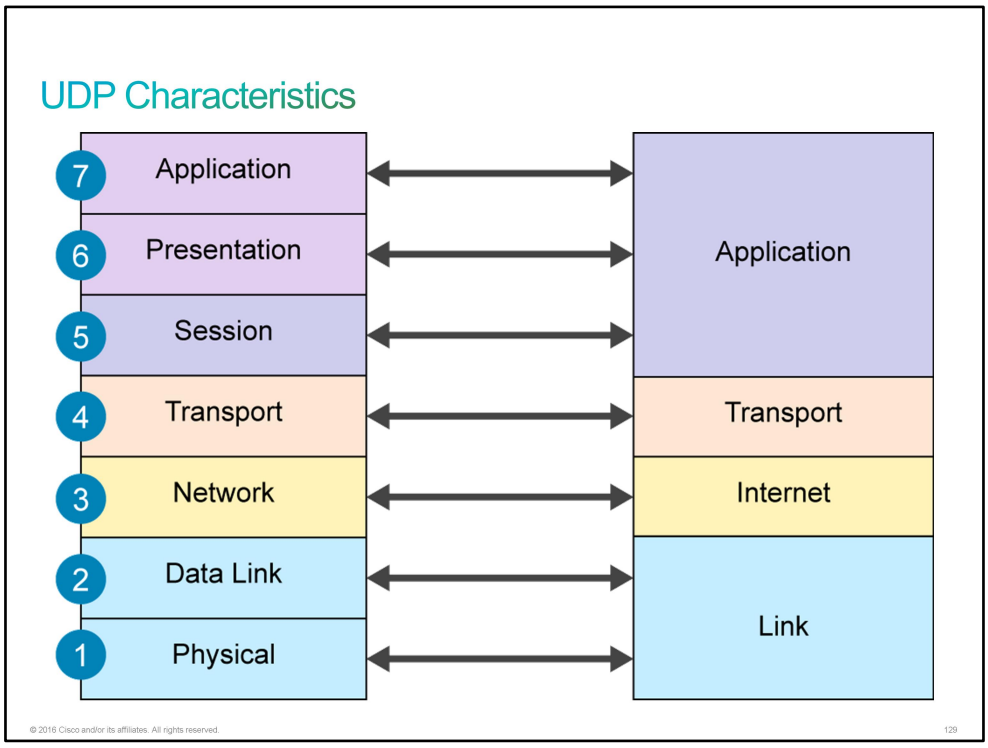
The fields of the TCP header include the following:

- **Source Port:** Number of the calling port (16 bits)
- **Destination Port:** Number of the called port (16 bits)
- **Sequence Number** and **Acknowledgment Number:** Used for reliability and congestion avoidance (32 bits each)
- **Header Length:** Size of the TCP header (4 bits)
- **Reserved:** For future use
- **Flags** or control bits (9 bits)
- **Window size:** Number of the window size (16 bits)
- **Checksum:** Calculated checksum of the header and fields that are used for error checking (16 bits)
- **Urgent Pointer:** If the URG flag is set, this field is an offset from the sequence number indicating the last urgent data byte (16 bits)
- **Options:** The length of this field is determined by the data offset field (from 0 to 320 bits)
- **Data:** ULP data (varies in size)



UDP Characteristics

UDP is a core protocol in the TCP/IP protocol suite. Applications leverage the connectionless services of UDP to provide high-performance, low-overhead data communications between hosts. UDP includes several features that provide for low-latency data transmission.



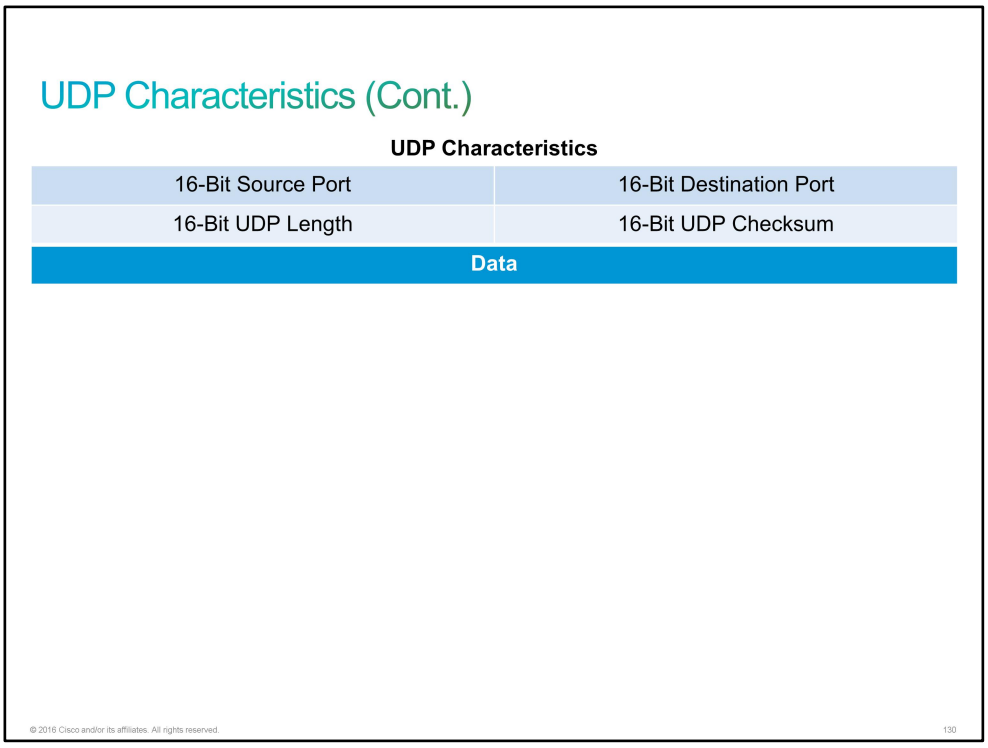
UDP is a simple protocol that provides basic transport layer functions:

- UDP operates at the transport layer of the TCP/IP stack (OSI Layer 4).
- UDP provides applications with access to the network layer without the overhead of reliability mechanisms.
- UDP is a connectionless protocol in which a one-way datagram is sent to a destination without advance notification to the destination device.
- UDP performs only limited error checking. A UDP datagram includes an optional checksum value, which the receiving device can use to test the integrity of the data.
- UDP provides service on a best-effort basis and does not guarantee data delivery, because packets can be misdirected, duplicated, or lost on the way to their destination.
- UDP does not provide any special features that recover lost or corrupted packets. UDP relies on applications that are using its transport services to provide recovery.
- Because of its low overhead, UDP is ideal for applications like DNS and NTP, where there is a simple request-and-response transaction.

An easy way to think of UDP is to use a postal service analogy. You are going to host a three-family garage sale next weekend, and you would like to send postcards that notify neighbors about the event, including the day, time, and location. You address each postcard with the name and address of your neighbors within a 6.2-mile (10-km) radius. The postal service delivers each postcard by any truck and any route. You have the option of paying additional postage for a delivery confirmation, but you decide that this additional expense is unnecessary because it is not important if a postcard is lost in transit, or if a neighbor acknowledges receipt of the message.

The low overhead of UDP is evident when you review the UDP header length of only 64 bits (8 bytes). So, the UDP header length is significantly smaller compared with the TCP minimum header length of 20 bytes. The following list describes the field definitions in the UDP segment:

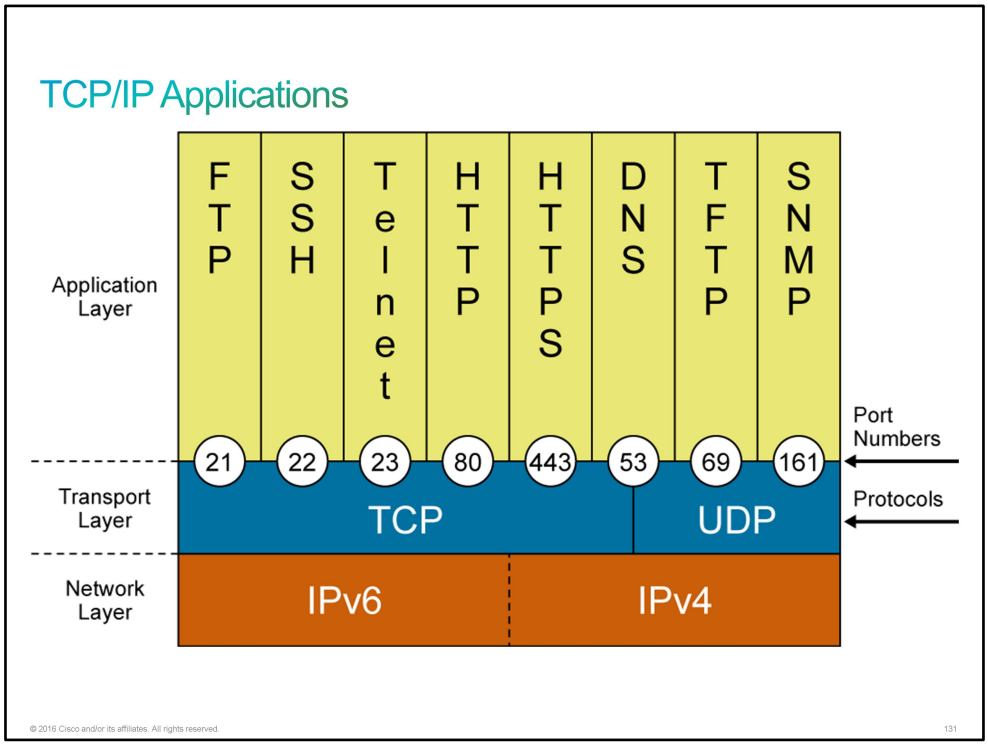
- **Source Port:** Number of the calling port (16 bits)
- **Destination Port:** Number of the called port (16 bits)
- **Length:** Length of UDP header and UDP data (16 bits)
- **Checksum:** Calculated checksum of the header and data fields (16 bits)
- **Data:** ULP data (varies in size)



Application layer protocols that use UDP include DNS, SNMP, DHCP, RIP, TFTP, NFS, online games, and streaming media.

TCP/IP Applications

UDP and TCP use internal software ports to support multiple conversations between various network devices. To differentiate the segments and datagrams for each application, TCP, and UDP both have header fields that uniquely identify these applications. These unique identifiers are the port numbers.



Some of the applications that TCP/IP supports:

- **FTP (port 21, TCP):** FTP is a reliable, connection-oriented service that uses TCP to transfer files between systems that support FTP. FTP supports bidirectional binary and ASCII file transfers. In addition to port 21, which is used for exchange of control, it also uses one additional port for data transmission.
- **SSH (port 22, TCP):** SSH provides the capability to remotely access other computers, servers, and networking devices. SSH enables a user to log in to a remote host and execute commands. SSH messages are encrypted.
- **Telnet (port 23, TCP):** Telnet is a predecessor to SSH. It sends messages in unencrypted cleartext. Most organizations now use SSH for remote communications.
- **HTTP (port 80):** HTTP defines how messages are formatted and transmitted and what actions browsers and web servers take in response to various commands. It uses TCP.
- **HTTPS (port 443, TCP):** HTTPS combines HTTP with a security protocol (SSL/TLS).
- **DNS (port 53, TCP, and UDP):** DNS is used to resolve Internet names to IP addresses. DNS uses a distributed set of servers to resolve names that are associated with numbered addresses.

- **TFTP (port 69, UDP):** TFTP is a connectionless service. Routers use TFTP to transfer configuration files and Cisco IOS images and other files between systems that support TFTP.
- **SNMP (port 161, UDP):** SNMP is an application layer protocol and it facilitates the exchange of management information between network devices. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth.

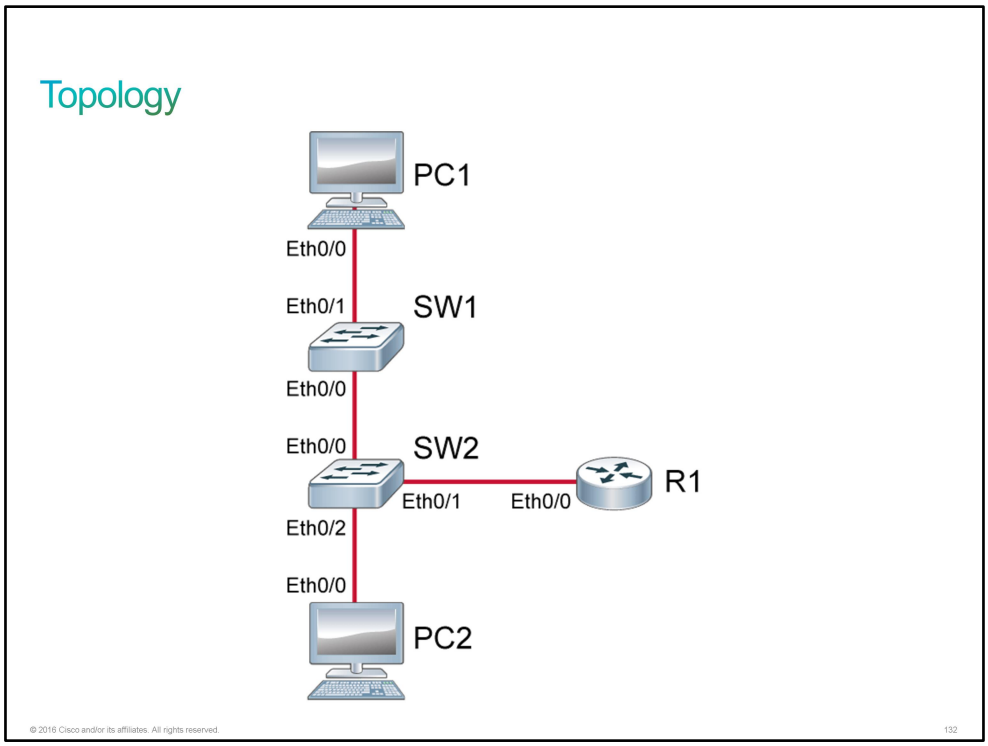
Here, you have seen only some applications with their port numbers. Go to the *Service Name and Transport Protocol Port Number Registry* for a complete list at http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

Discovery 5: Inspect TCP/IP Applications

Introduction

This discovery lab will help you explore TCP and UDP sockets. That is, how TCP and UDP servers listen on particular ports that are made available on particular interfaces, and how clients connect to the servers using their own IP addresses and their own ports. The lab is prepared with the devices represented in the topology diagram with the IP addresses as depicted in the table.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2

Device	Characteristic	Value
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Inspect TCP/IP Applications

Activity

Step 1 R1 has been configured to run several TCP services including Telnet, SSH, HTTP, and HTTPS. It has also been configured to run an NTP service. How and why these services may be configured on a router is beyond the scope of this discovery. For now, verify the services that are running on R1 by viewing its open ports. Access the console of R1 and execute the **show control-plane host open-ports** command.

There are several open TCP ports: 22 for SSH, 23 for Telnet, 80 for HTTP and 443 for HTTPS.


```
R1# show control-plane host open-ports
Active internet connections (servers and established)
Prot               Local Address             Foreign Address
Service    State
 tcp                        *:22                          *:0               SSH-
Server   LISTEN
 tcp                        *:23                          *:0
Telnet   LISTEN
 tcp                        *:80                          *:0
HTTP CORE   LISTEN
 tcp                        *:80                          *:0
HTTP CORE   LISTEN
 tcp                       *:443                          *:0
HTTP CORE   LISTEN
 tcp                       *:443                          *:0
HTTP CORE   LISTEN
 udp                       *:123                          *:0
NTP   LISTEN
```

These ports are in a listening state. That is, no foreign addresses are connected to them, but they are ready for connections to ensue.

Step 2 Access the console of PC1 and use Telnet to connect to R1. The password that is configured on R1 is "Cisco123."

The prompt changes from PC1 to R1 because you are now connected to R1 via Telnet from PC1.

```
PC1# telnet 10.10.1.1
Trying 10.10.1.1 ... Open

User Access Verification

Password: <Cisco123>
R1#
```

Step 3 Return to the console of R1 and review the open ports. You may want to use the Cisco IOS command recall feature to re-enter the command.

There is an additional line in the output when compared with the last execution. It shows a second line that is associated with TCP port 23. In this case, the foreign address is populated. The IP address is 10.10.1.10 (the IP address of PC1). The foreign port number might not be the same as what is shown in the example because it will be an ephemeral port.

```
R1# show control-plane host open-ports
Active internet connections (servers and established)
Prot                Local Address             Foreign Address
Service    State
 tcp                       *:22                      *:0                SSH-
Server   LISTEN
 tcp                       *:23                      *:0
Telnet   LISTEN
 tcp                       *:80                      *:0
HTTP CORE   LISTEN
 tcp                       *:80                      *:0
HTTP CORE   LISTEN
 tcp                      *:443                      *:0
HTTP CORE   LISTEN
 tcp                      *:443                      *:0
HTTP CORE   LISTEN
 tcp                       *:23           10.10.1.10:14044
Telnet ESTABLIS
 udp                      *:123                      *:0
NTP   LISTEN
```

Step 4 Return to the console of PC1 and use the **exit** command to disconnect the telnet session to R1.

The prompt returns to PC1 as you are no longer connected to the R1.

```
R1# exit

[Connection to 10.10.1.1 closed by foreign host]
PC1#
```

Alternatively, you could have used the **logout** command to disconnect from R1.

Step 5 Return to the console of R1 and review the open ports again.

All ports are in a listening state.

```
R1# show control-plane host open-ports
Active internet connections (servers and established)
Prot                Local Address             Foreign Address
Service    State
 tcp                       *:22                      *:0                SSH-
Server   LISTEN
 tcp                       *:23                      *:0
Telnet   LISTEN
 tcp                       *:80                      *:0
HTTP CORE   LISTEN
 tcp                       *:80                      *:0
HTTP CORE   LISTEN
 tcp                      *:443                      *:0
HTTP CORE   LISTEN
 tcp                      *:443                      *:0
HTTP CORE   LISTEN
 udp                      *:123                      *:0
NTP   LISTEN
```

This is the end of the discovery lab.

Challenge

1. What is a major difference between TCP and UDP?

   A. TCP can ensure that the data is delivered, while UDP does not.
   B. UDP can ensure that the data is delivered, while TCP does not.
   C. TCP exists in the transport layer of the TCP/IP model, while UDP exists in the Internet layer of the TCP/IP model.
   D. TCP exists in the Internet layer of the TCP/IP model, while UDP exists in the transport layer of the TCP/IP model.

2. What are the three main objectives of TCP reliability? (Choose three.)

   A. detection and retransmission of dropped packets
   B. detection and remediation of duplicate or out-of-order data
   C. avoidance of congestion in the network
   D. detection of dropped packets. The application does the retransmission.
   E. detection of duplicate or out-of-order data. The application does the remediation.
   F. avoidance of delay in the network

3. What is session multiplexing?

   A. a process by which an IP host is able to support multiple sessions simultaneously and manage the individual traffic streams over a single link.
   B. a process by which an IP host is able to support multiple sessions simultaneously over multiple links.
   C. a process that is used for congestion avoidance.
   D. a process that is used for packet dropping.

4. What are three common applications that use TCP? (Choose three.)

   A. DNS
   B. web browsers
   C. email
   D. streaming video
   E. FTP
   F. VoIP

5. What are three common applications that use UDP? (Choose three.)

   A. network printing
   B. TFTP
   C. database transactions
   D. streaming video
   E. FTP
   F. VoIP

6. Which OSI stack layers match the application layer of the TCP/IP stack?

   A. application layer, presentation layer, and session layer
   B. presentation layer, session layer, and transport layer
   C. application layer, presentation layer, session layer, and transport layer
   D. network layer and transport layer

7. Which OSI stack layers match the link layer of the TCP/IP stack?

   A. network layer, data link layer, and physical layer
   B. data link layer only
   C. data link layer and physical layer
   D. physical layer only

Answer Key

Challenge

1. A
2. A, B, C
3. A
4. B, C, E
5. B, D, F
6. A
7. C

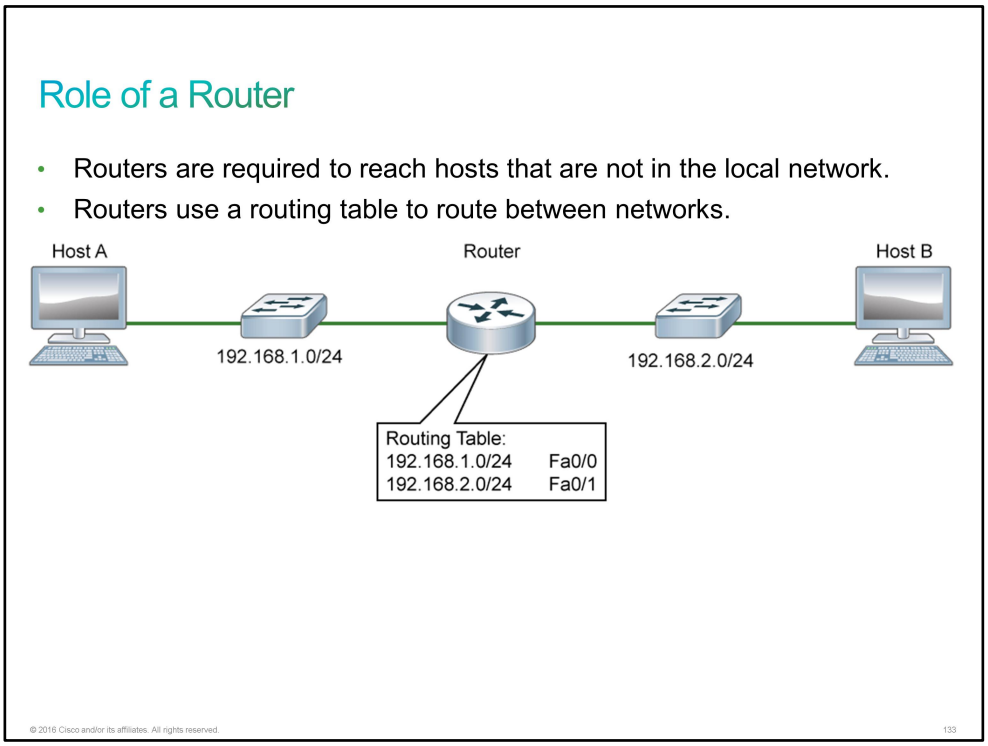
Lesson 4: Exploring the Functions of Routing

Introduction

Bob tells you that you will also need to understand routing functionality, including the types of routes and how dynamic routing protocols work. He tells you not to worry about the differences in link-state and distance vector routing protocols at this point and that you only need a high-level understanding of the functions of dynamic routing protocols.

Role of a Router

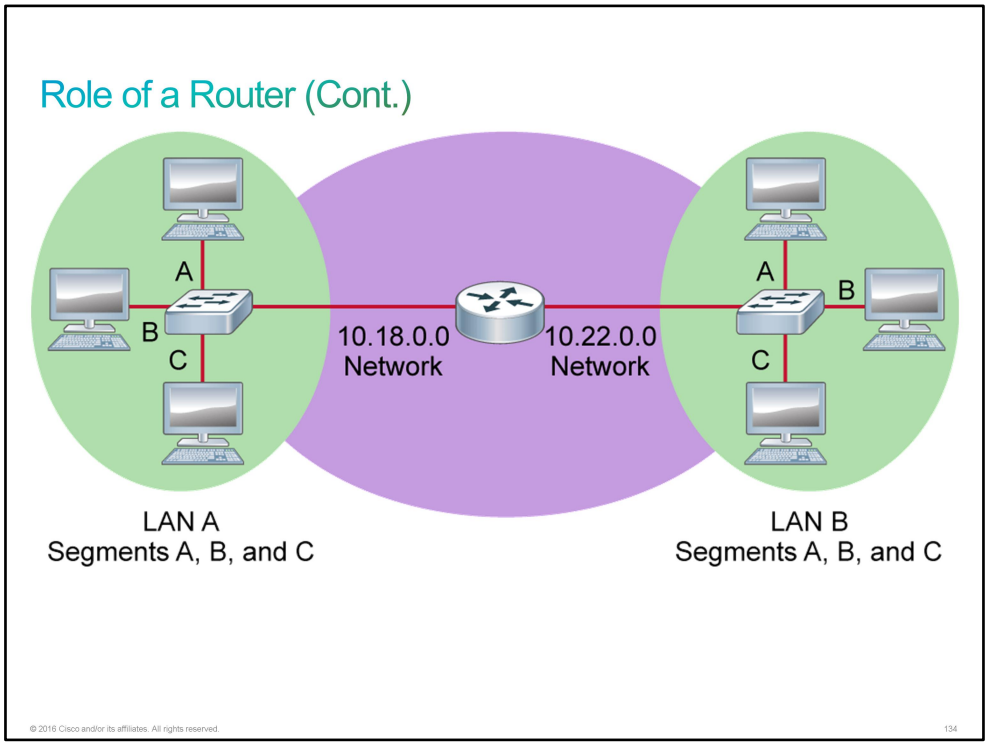
A router is a networking device that forwards packets between different networks or LANs.



While switches exchange data frames between segments to enable communication within a single network, routers are required to reach hosts that are not in the local LAN. Routers enable internetwork communication by placing the interface of each router in the network of the other routers. They use routing tables to route traffic between different networks.

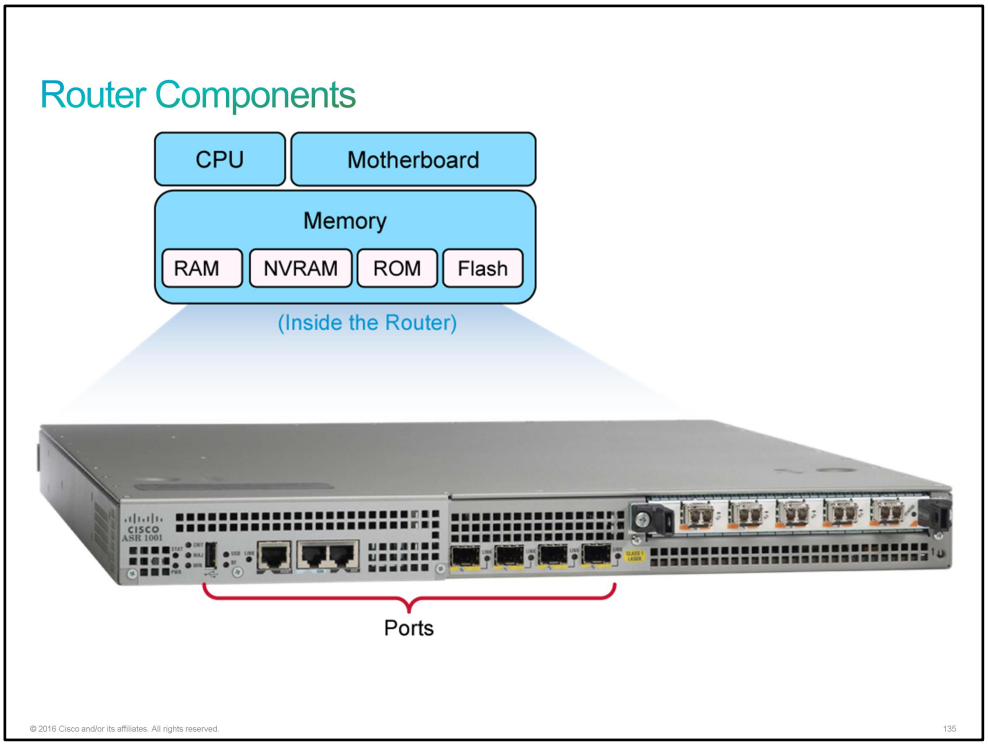
In the following figure, LAN A switches data frames between its segments—A, B, and C—to enable communication among hosts on those segments. In other words, the LAN A switch enables communication within a single network, LAN A, whose network IP address is 10.18.0.0/16. Likewise, the LAN B switch enables communication among the hosts on LAN B, whose network IP address is 10.22.0.0/16.

A host in LAN A cannot communicate with a host in LAN B without the router. Routers enable communications between hosts that are not in the same local LAN. Routers are able to do this function because they can be attached to multiple networks and have the ability to route between them. Routers are essential to large networks that use TCP/IP, because they can accommodate growth across wide geographical areas.



Router Components

Cisco offers many different routers, which come in many shapes and sizes. The various models offer various features that are suitable for an array of different environments. However, the core function of a router is to route packets, and for that reason, all routers have many common components.



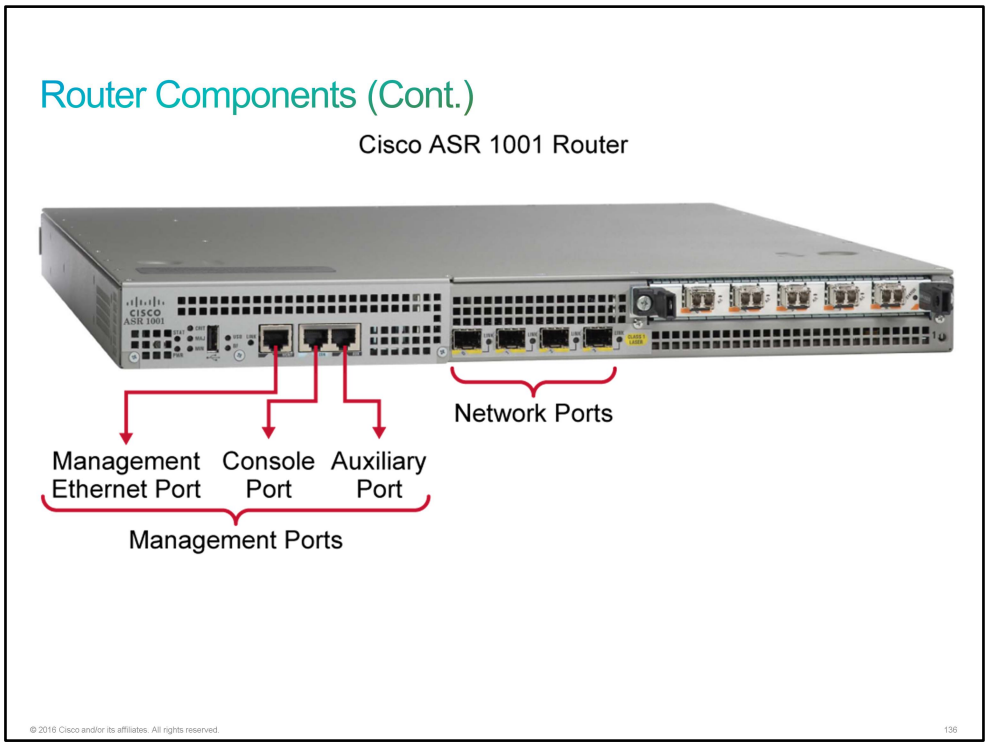
These components are as follows:

- **CPU:** CPU, or processor, is the chip that is installed on the motherboard that carries out the instructions of a computer program. For example, it processes all the information that is gathered from other routers or sent to other routers.
- **Motherboard:** The motherboard is the central circuit board, which holds critical electronic components of the system. The motherboard provides connections to other peripherals and interfaces.
- **Memory:** There are four primary types of memory:
  - **RAM:** RAM is memory on the motherboard that stores data during CPU processing. It is a volatile type of memory in that its information is lost when power is switched off. RAM provides temporary memory for the running configuration of the router while the router is powered on.
  - **NVRAM:** NVRAM retains content when the router is powered down. NVRAM stores the startup configuration file for most router platforms. It also contains the software configuration register, which is used to determine which image to use when booting the router.
  - **ROM:** ROM is read-only memory on the motherboard. The content of ROM is not lost when power is switched off. Data that is stored in ROM cannot be modified, or it can be modified only slowly or with difficulty. ROM sometimes contains a ROM monitor, which provides a user interface when the router cannot find a valid image, and bootloader software, which helps the router boot when it cannot find a valid Cisco IOS image in the flash memory.

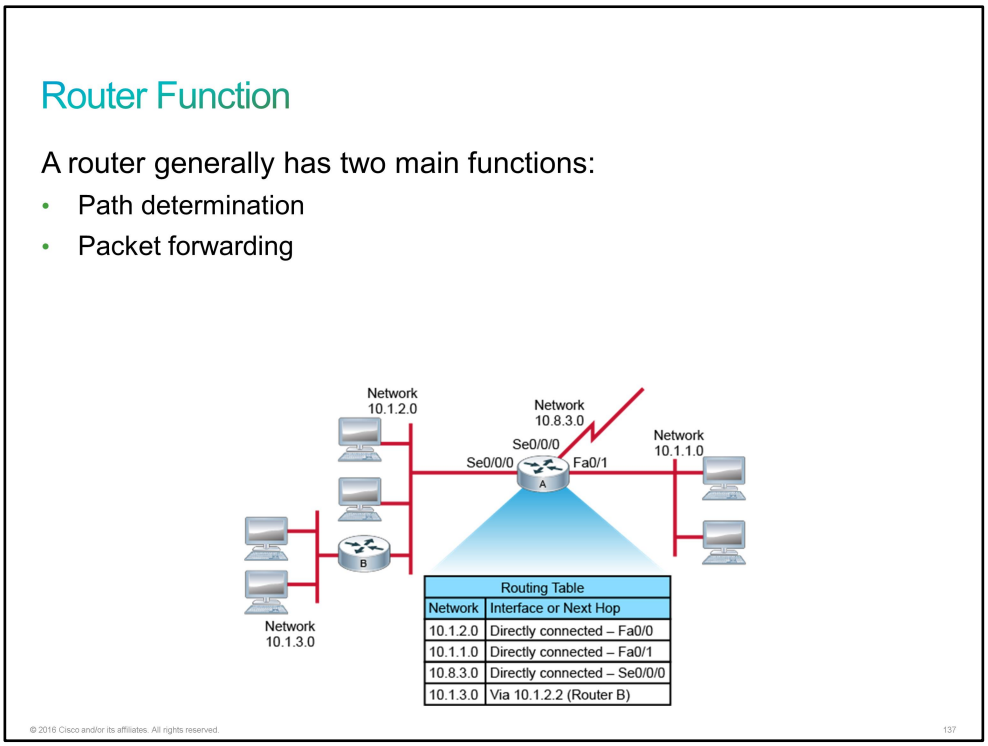
  - **Flash:** Flash memory is nonvolatile storage that can be electrically erased and reprogrammed. Flash memory stores the Cisco IOS Software image. On some platforms, it can also store configuration files or boot images.

- **Ports:** Ports are used to connect routers to other devices in the network. Routers can have these types of ports:
  - **Management ports:** Management ports are for the connection of a terminal that is used for management. Routers have a console port that can be used to attach to a terminal that is used for management, configuration, and control. High-end routers may also have a dedicated Ethernet port that can be used only for management. An IP address can be assigned to the Ethernet port, and the router can be accessed from a management subnet. The AUX interface on a router is used for remote management of the router. Typically, a modem is connected to the AUX interface for dial-in access. From a security standpoint, enabling the option to connect remotely to a network device carries with it the responsibility of vigilant device management.
  - **Network ports:** The router has many network ports, including various LAN or WAN media ports, which may be copper or fiber cable. IP addresses are assigned to network ports.

As an example, the following figure shows the ports on a Cisco ASR 1001 Router:



Router Function



Routers have these two important functions:

- **Path determination:** Routers use their routing tables to determine where to forward packets. Each router must maintain its own local routing table, which contains a list of all destinations that are known to the router, and information about how to reach those destinations. When a router receives an incoming packet, it checks the destination IP address in the packet and searches for the best match between the destination address and the network addresses in the routing table. A matching entry may indicate that the destination is directly connected to the router or that it can be reached via another router. This router is called the next-hop router and is on the path to the final destination. If there is no matching entry, the router sends the packet to the default route. If there is no default route, the router drops the packet.
- **Packet forwarding:** After a router determines the appropriate path for a packet, it forwards the packet through a network interface toward the destination network. As shown in the figure, each line of the routing table lists a destination network and its corresponding interface or next-hop address. If there is an interface on the router that has an IP address within the destination network, the destination network is considered "directly connected" to the router. For example, assume that router A receives a packet on its Serial0/0/0 interface that is destined for a host on network 10.1.1.0. Because the routing table indicates that network 10.1.1.0 is directly connected, router A forwards the packet directly to the host via its FastEthernet0/1 interface. If a destination network in the routing table is not directly connected, the packet must reach the destination network via the next-hop router. For example, assume that Router A receives a packet on its Serial0/0/0 interface and the destination host address is on the 10.1.3.0 network. In this case, it must forward the packet to the router B interface with the IP address 10.1.2.2. Routers support three packet-forwarding mechanisms:
  - **Process switching:** Process switching is the oldest forwarding mechanism that is available in Cisco routers. Every packet requires a full lookup in the routing table, which makes this mechanism very slow. It is typically not used in modern networks.

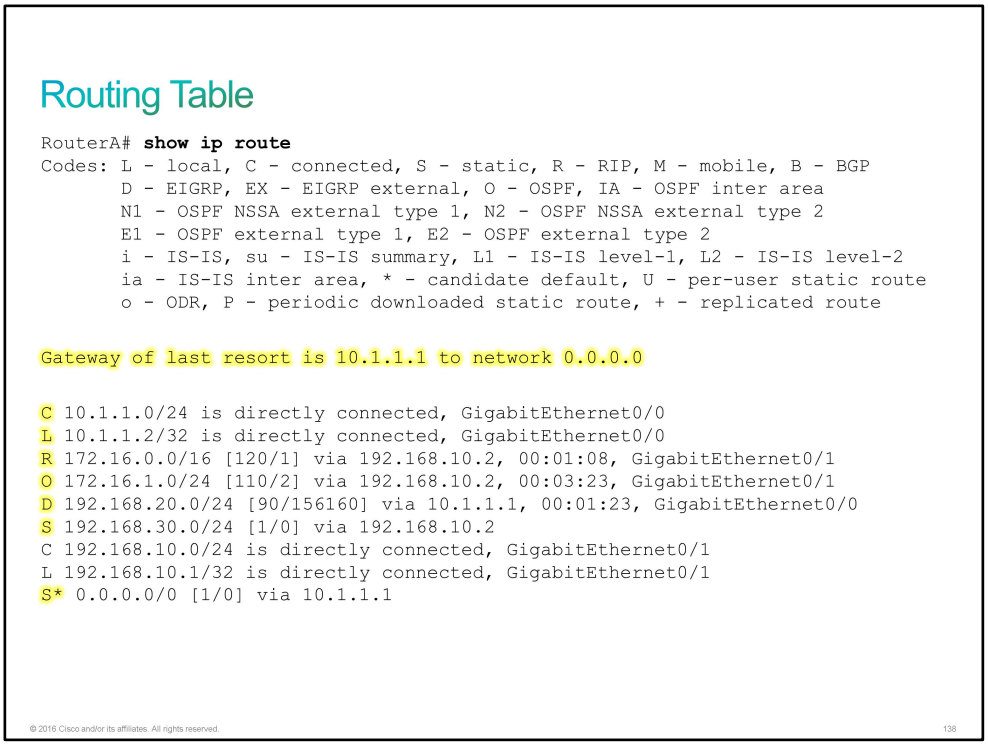
- **Fast switching:** To overcome the slow performance of process switching, Cisco IOS platforms support several switching mechanisms that use a cache to store the most recently used destinations. The first packet whose destination is not found in the fast-switching cache is process-switched, and an entry is created in the cache. Subsequent packets are then able to use fast switching.
- **Cisco Express Forwarding:** Cisco Express Forwarding is the most recent and preferred Cisco IOS packet-forwarding mechanism, which incorporates the best of the previous switching mechanisms. Changes in the network instead of packets trigger the generation of cache table entries. When something changes in the network topology, the change is also reflected in the cache table. All packets are switched using the Cisco Express Forwarding cache, which makes Cisco Express Forwarding the fastest forwarding mechanism and the preferred choice.

Routing Table

A routing table contains a list of all networks that are known to the router and information about how to reach those networks. Each line, or entry, of the routing table lists a destination network and the interface or next-hop address by which that destination network can be reached.

A routing table may contain the following types of entries:

- **Directly connected networks:** All directly connected networks are added to the routing table automatically. A directly connected network is a network that is directly connected to one of the interfaces on the local router. If the interface fails or is administratively shut down, the entry for that network is removed from the routing table.
- **Static routes:** Static routes are entries that you manually enter directly into the configuration of the router. Static routes can be effective for small, simple networks that do not change frequently. However, statically populating routing tables does not scale well and can lead to problems if the network topology changes.
- **Default routes:** A default route is an optional entry that is used when no explicit path to a destination is found in the routing table. You can manually configure the default route as a static route, or a routing protocol can enter it.
- **Dynamic routes:** The router learns dynamic routes automatically when a routing protocol is configured and a neighbor relationship to other routers is established. The information is updated when changes in the network occur. Larger networks require the dynamic routing method because there are usually many addresses and constant changes. These changes require updates to routing tables across all routers in the network, to prevent connectivity loss.

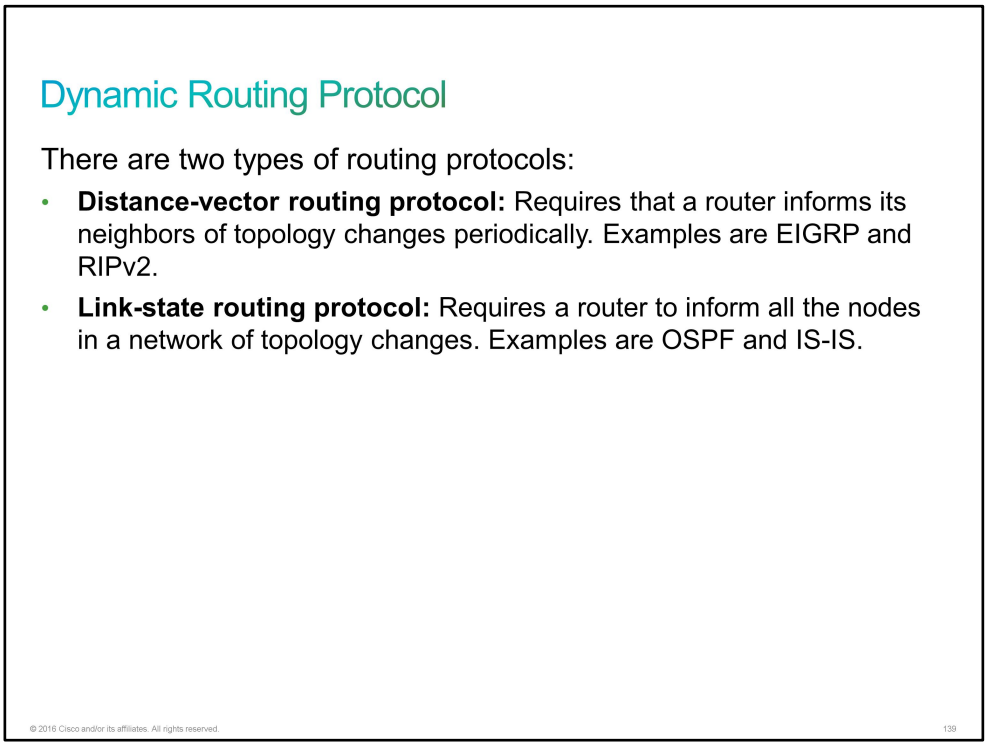


The figure shows the output of the **show ip route** command, which is used to display the contents of the routing table in a router. The first part of the output explains the codes, presenting the letters and the associated sources of the entries in the routing table.

- **The letter C:** Reserved for directly connected networks; labels the first and sixth entries.
- **The letter L:** Reserved for local routes and indicating local interfaces within connected networks; labels the second and seventh entries.
- **The letter S:** Reserved for static routes; labels the fifth and eighth entries.
- **The asterisk (*):** Indicates a default route. In this example command output, the default route is a static route.
- **The letter R:** Reserved for the RIP routing protocol; labels the third entry.
- **The letter O:** Reserved for the OSPF routing protocol; labels the fourth entry.
- **The letter D:** Reserved for EIGRP; labels the fifth entry. The letter D stands for DUAL, which is the update algorithm that EIGRP uses.

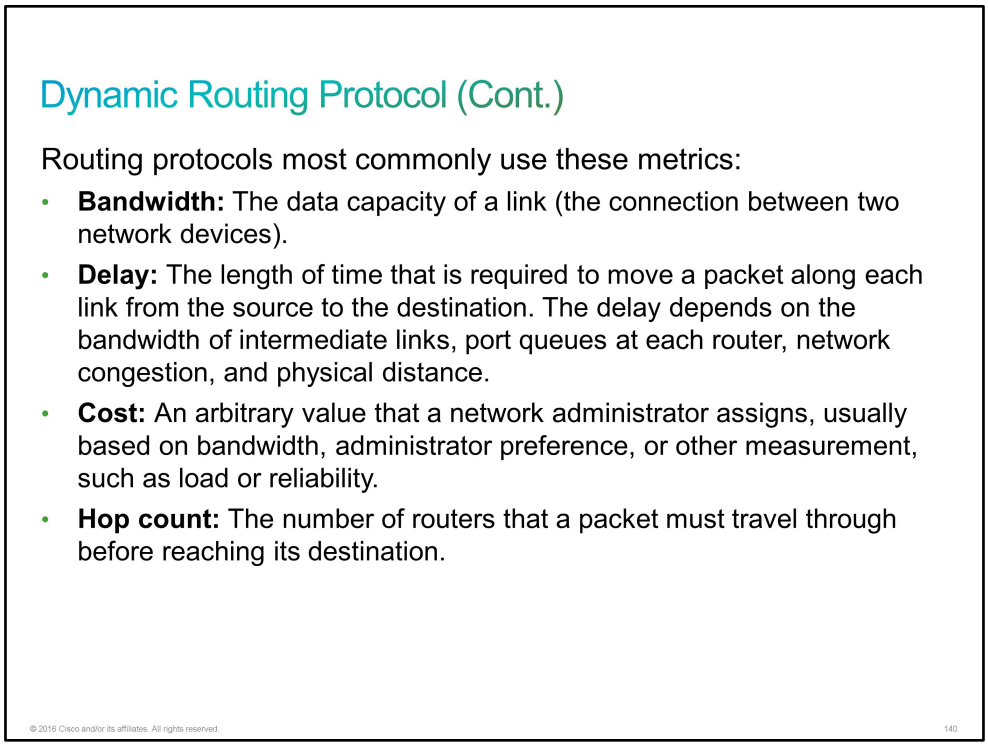
Dynamic Routing Protocol

A routing protocol is a set of processes, algorithms, and messages by which routers dynamically share their routing information. Examples of routing protocols include OSPF, EIGRP, RIPv2, and IS-IS.



Routers that are running routing protocols exchange routing update messages to keep their routing tables updated. When a router that is running a routing protocol becomes aware of changes to the network, it passes the information on to other routers that are running the same routing protocol. When a router receives information about new or changed routes, it updates its own routing table and, in turn, passes the information to other routers. In this way, all routers maintain accurate routing tables that are updated dynamically and can learn about routes to remote networks that are many hops away.

Routing protocols not only enable routers to learn about remote networks and to quickly adapt whenever there is a change in the topology; they also enable routers to choose the best path to destination networks. Although there may be multiple paths to a given destination, a routing table holds only one path to any given destination. A routing table holds only one entry for every network.

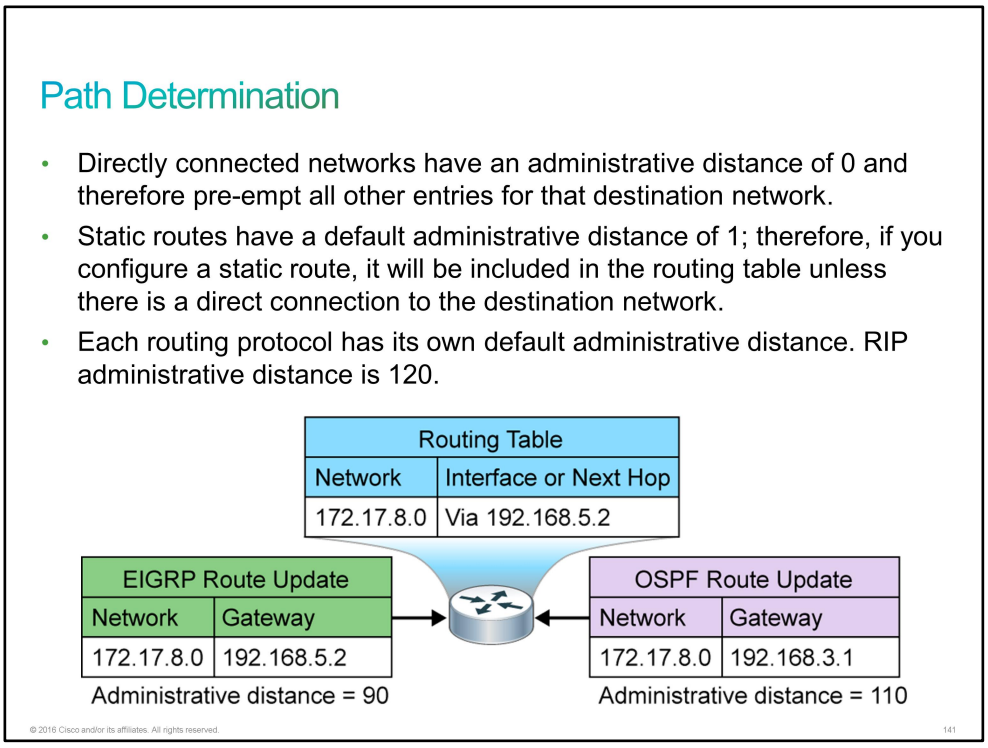


To determine the best path to any given destination, routing protocols use a number that is called a metric. The metrics that the various routing protocols use differ, depending on the design of the routing algorithm that is used. The routing algorithm that the protocol uses generates a metric for each path through the network. Metrics can be based on either a single characteristic or on several characteristics of a path. Sophisticated routing protocols can base route selection on multiple metrics, combining them into a single metric. Typically, the lower the metric value, the better the path.

Path Determination

Routing tables can be populated from three types of sources: directly connected networks, static routes, and routing protocols. The router must be able to evaluate the routing information from all the sources and select the best route to populate the routing table.

Routers use a feature called administrative distance to select the best path when they learn two or more different routes to the same destination. Administrative distance defines the reliability of the route source—the smaller the administrative distance value, the more trusted the source. Each source type has a default administrative distance. For example:



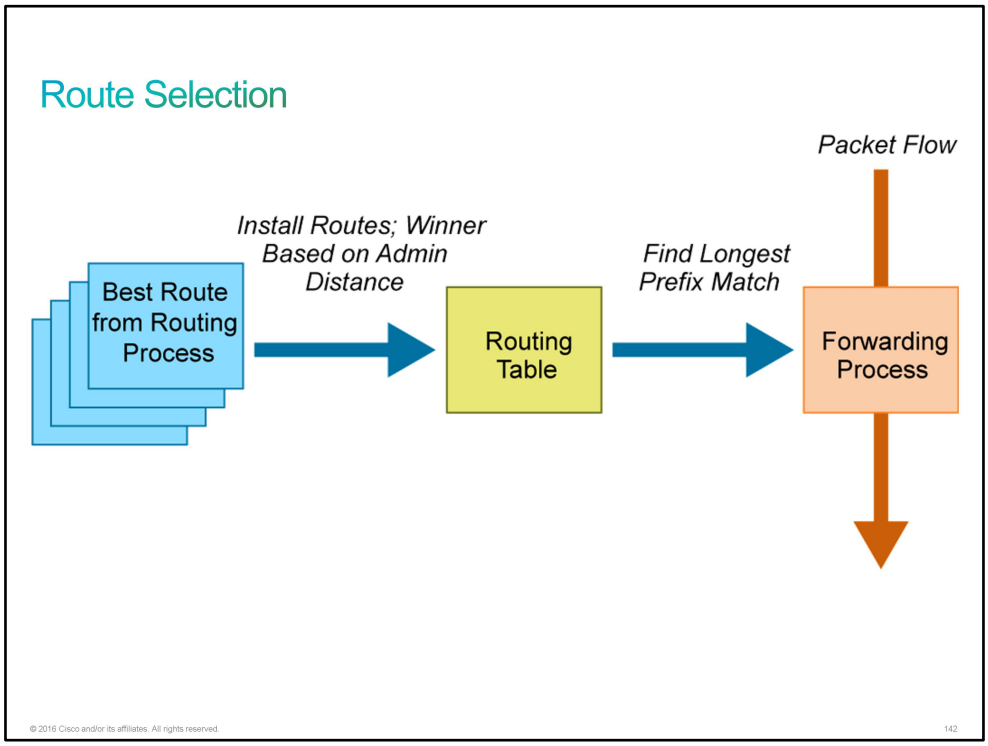
In the figure, the router has received two routing update messages—one from OSPF and one from EIGRP. The metric that EIGRP uses has determined that the best path to network 172.17.8.0 is via 192.168.5.2, but the metric that OSPF uses has determined that the best path to 172.17.8.0 is via 192.168.3.1. The router has used the administrative distance feature to determine which route to install in its routing table. Because the administrative distance for OSPF is 110 and the administrative distance for EIGRP is 90, the router has chosen the EIGRP route and adds only the EIGRP route to its routing table.

Because each entry in a routing table may specify a subnetwork, one destination address may match more than one routing table entry. The most specific of the matching table entries is called the longest prefix match. It is called this way because it is also the entry where the largest number of leading address bits of the destination address matches those addresses in the table entry.

For example, consider a routing table with these entries: 10.1.1.0/24 and 10.1.0.0/16. When the address 10.1.1.1 needs to be looked up, both entries match. In this case, the longest prefix of the candidate routes is 10.1.1.0/24.

Route Selection

Making a forwarding decision actually consists of three sets of processes: the routing protocols, the routing table, and the actual process that makes the forwarding decision and switches packets.



Three processes are involved in building and maintaining the routing table in a Cisco router:

- Various routing processes, which actually run a routing protocol, such as RIPv2, EIGRP, IS-IS, and OSPF. The best route from the routing process has a potential to be installed into the routing table.
- The routing table itself, which accepts information from the routing processes and also replies to requests for information from the forwarding process.
- The forwarding process, which requests information from the routing table to make a packet forwarding decision.

The longest prefix match always wins among the routes that are actually installed in the routing table, whereas the routing protocol with the lowest administrative distance always wins when installing routes into the routing table.

Challenge

1. Which statement best describes the role of a router?

   A. Routers are responsible only for reaching hosts that are in the local network.
   B. Routers use a MAC table to route between networks.
   C. Routers are required to reach hosts that are not in the local network.
   D. Routers use an ARP table to route between networks.

2. What do you call the router component that holds critical electronic components of the system?

   A. CPU
   B. motherboard
   C. memory
   D. ports

3. In which part of the router memory is the running configuration stored?

   A. RAM
   B. NVRAM
   C. ROM
   D. flash

4. Which two types of ports are available on a router? (Choose two.)

   A. network ports
   B. console ports
   C. AUX ports
   D. debug ports
   E. management ports
   F. monitoring ports

5. In a routing table, what is an optional entry that is used when no explicit path to a destination is found?

   A. static route
   B. directly connected route
   C. OSPF route
   D. default route

6. What does the letter D that is associated with the routing table entry present?

   A. route is learned by OSPF
   B. route is learned by EIGRP
   C. route is static
   D. route is learned by DMVPN

7. What is the administrative distance for OSPF?

   A. 110
   B. 115
   C. 90 for internal route and 170 for external route
   D. 170 for internal route and 90 for external route

Answer Key

Challenge

1. C
2. B
3. A
4. A, E
5. D
6. B
7. A

Lesson 5: Configuring a Cisco Router

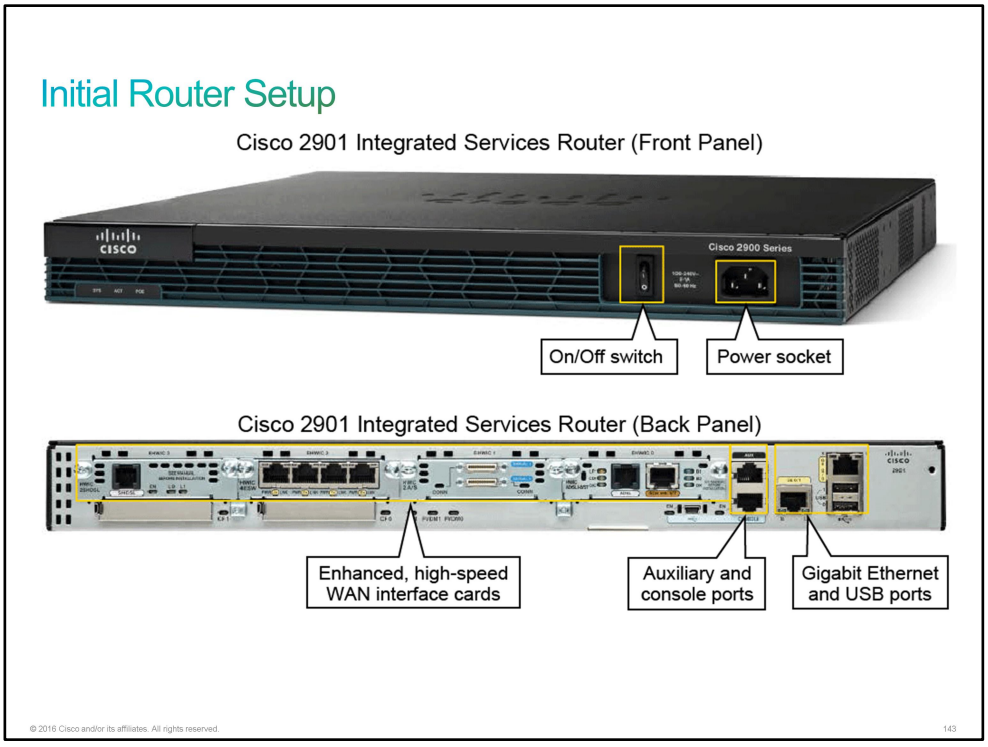
Introduction

Your boss sends you to a customer to install a new router. The router is already physically set, but you will need to configure it. You need to understand the initial configuration steps to properly configure the router. Also, you will configure and verify an interface on the router and use Cisco Discovery Protocol to draw the network topology.

Initial Router Setup

Cisco provides several different types of router hardware, including some routers that do only routing, while other routers offer additional functions. In fact, Cisco has a series of routers that is called ISR, with the name emphasizing the fact that many functions are integrated into a single device.

The following figure shows Cisco 2901 ISR with some of the more important features highlighted.



The startup of a Cisco router requires verifying the physical installation, powering up the router, and viewing the Cisco IOS Software output on the console. To start router operations, the router completes the following tasks:

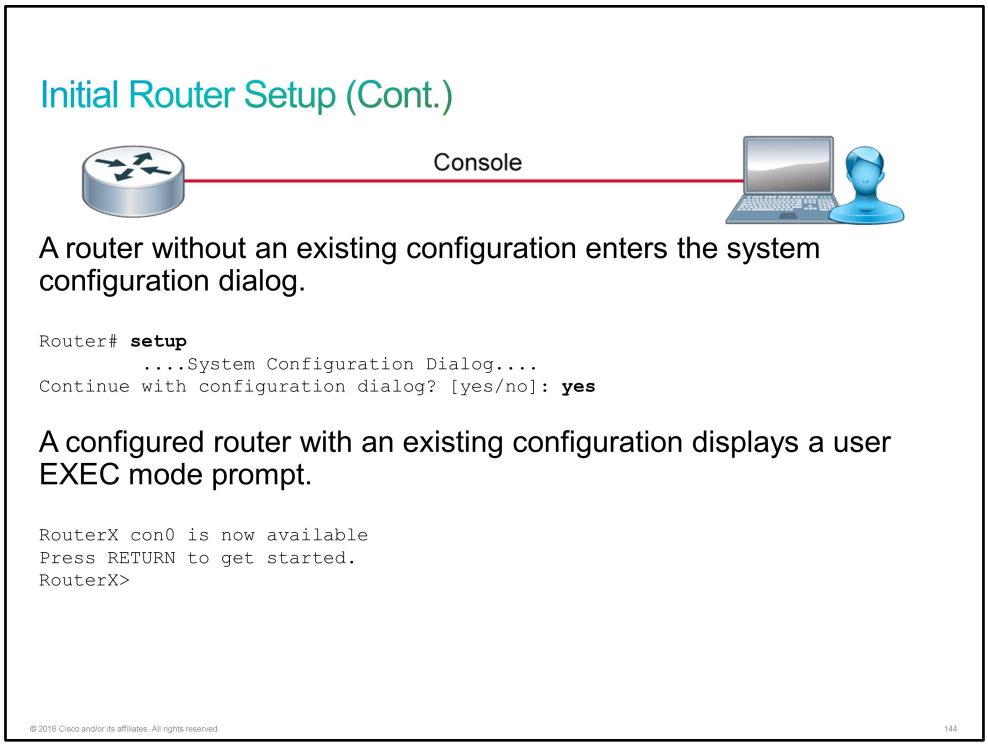
1. Runs the POST to test the hardware
2. Finds and loads the Cisco IOS Software that the router uses for its operating system
3. Finds and applies the configuration statements about router-specific attributes, protocol functions, and interface addresses

When a Cisco router powers on, it performs a POST. During the POST, the router executes diagnostics to verify the basic operation of the CPU, memory, and interface circuitry.

After verifying the hardware functions, the router proceeds with software initialization. During software initialization, the router finds and loads the Cisco IOS image. After the router loads the Cisco IOS image, it finds and loads the configuration file, if one exists.

Note	Before you start the router, verify the power and cooling requirements, cabling, and console connection. Then push the power switch to "On" and observe both the boot sequence and the Cisco IOS Software output on the console.

After a router completes the POST and loads a Cisco IOS image, it looks for a device configuration file in its NVRAM. If the router does not find one, it executes a question-driven, initial configuration routine that is called "setup." Setup is a prompt-driven program that allows a minimal device configuration. If the router has a startup configuration file in NVRAM, the user EXEC mode prompt appears.



When starting a new Cisco router, there is no configuration file. So the operating system executes the question-driven, initial configuration routine, which is referred to as the *initial configuration dialog* or *setup mode*.

The setup mode is not intended for entering complex protocol features in the router but rather for bringing up a minimal configuration. You do not have to use the setup mode; you can use other configuration modes to configure the router.

The primary purpose of the setup mode is to rapidly bring up a minimal-feature configuration for any router that cannot find its configuration from some other source. In addition to being able to run the setup mode when the router boots, you may also initiate it by entering the **setup** privileged EXEC mode command.

To skip the system configuration dialog and configure the router manually, answer the first question in the system configuration dialog with **no** or press **Ctrl-C**.

To verify the router status, use the **show version** command:

```
R1# show version
Cisco IOS Software, Linux Software (I86BI_LINUX-ADVENTERPRISEK9-M), Version 15.2(4)M3,
DEVELOPMENT TEST SOFTWARE
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Tue 26-Feb-13 19:06 by prod_rel_team

ROM: Bootstrap program is Linux

R1 uptime is 0 minutes
System returned to ROM by reload at 0
System restarted at 03:00:23 PST Wed Oct 7 2015
System image file is "unix:/iou_root/images/IOL/i86bi_linux-adventerprisek9-ms.152-
4.M3"
<... output omitted ...>
```

To verify the running configuration of the router, use the **show running-config** command:

```
R1# show running-config
Building configuration...

Current configuration : 2919 bytes
!
! No configuration change since last restart
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
<... output omitted ...>
```


Configuring Router Interfaces

One of the main functions of a router is to forward packets from one network device to another. For the router to perform this task, you must define the characteristics of the interfaces through which the router receives and sends the packets.

There are two general types of physical interfaces on Cisco routers: *Ethernet interfaces* and *serial interfaces*.

- **Ethernet interfaces:** The term Ethernet interface refers to any type of Ethernet interface. For example, some Cisco routers have an Ethernet interface that is capable of only 10 Mbps, so to configure this type of interface, you would use the **interface ethernet** *slot*/*interface number* configuration command. However, other routers have interfaces that are capable of operating up to 100 Mbps. These interfaces are referred to as Fast Ethernet ports. You use the **interface fastethernet** *slot/interface number* command to configure these types of ports. Similarly, the interfaces that are capable of Gigabit Ethernet speeds are referenced with the **interface gigabitethernet** *slot/interface number* command.
- **Serial interfaces:** Serial interfaces are the second major type of physical interfaces on Cisco routers. To support point-to-point leased lines and Frame Relay access-link standards, Cisco routers use serial interfaces. You can then choose which data link layer protocol to use, such as HDLC or PPP for leased lines or Frame Relay for Frame Relay connections, and configure the router to use the correct data link layer protocol. You use the **interface serial** *slot*/*interface number* command when configuring these types of interfaces.

Note	It is appropriate to mention the *loopback interface* here. A loopback interface is a virtual interface that resides on a router. It is not connected to any other device. Loopback interfaces are very useful because they will never go "down," unless the entire router goes down. This helps in managing routers because there will always be at least one active interface on the routers—the loopback interface. To create a loopback interface, all you need to do is enter the configuration mode for the interface. Optionally, you may add an IP address.

```
Router(config)# interface loopback 0
Router(config-if)# ip address 10.0.0.1 255.255.255.255
```

An IP address with all mask bits set to 1 is called the host IP address. The host IP address indicates that only one IP address is used in the subnet and is often used to address loopback interfaces.

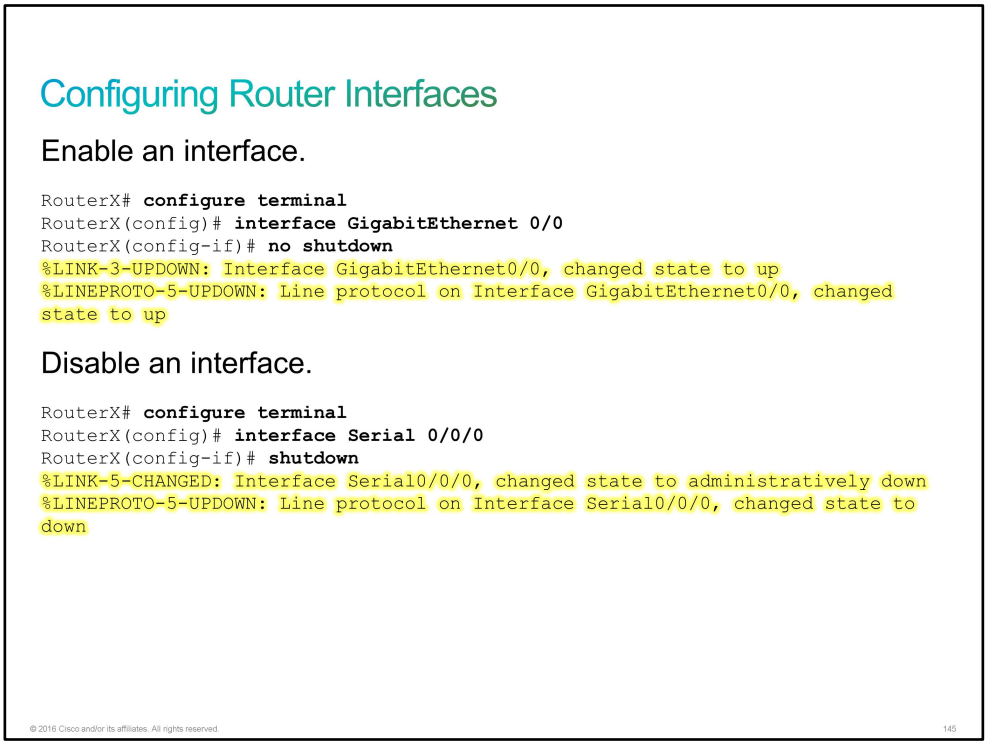
Routers use numbers to distinguish between the different interfaces of the same type. On routers, the interface numbers might be a single number, or two numbers that are separated by a slash, or three numbers that are separated by slashes. For example, all three of the following configuration commands are correct on at least one Cisco router model:

```
interface ethernet 0
interface fastethernet 0/1
interface serial 1/0/1
```

Note	The router interface characteristics include, but are not limited to, interface description, the IP address of the interface, the data link encapsulation method, the media type, the bandwidth, and the clock rate. You can enable many features on a per-interface basis.

When you first configure an interface, except in the setup mode, you must administratively enable the interface before the router can use it to transmit and receive packets. Use the **no shutdown** command to allow Cisco IOS Software to use the interface.

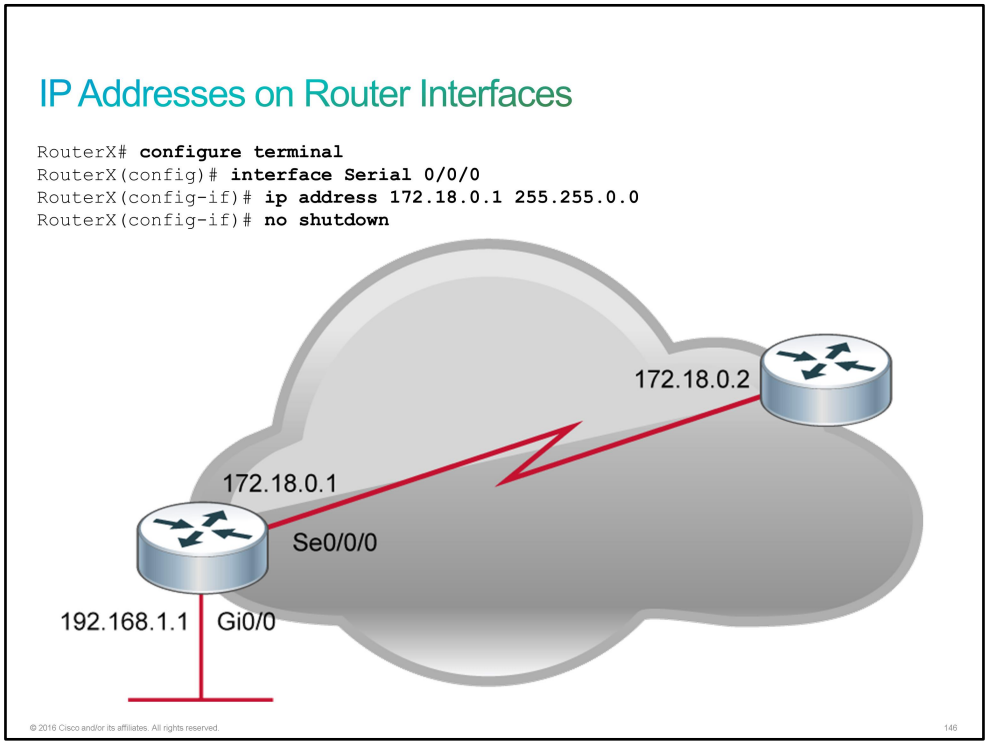
You may want to disable an interface to perform hardware maintenance on a specific interface or a segment of a network. You may also want to disable an interface if a problem exists on a specific segment of the network, and you must isolate this segment from the rest of the network. The **shutdown** command administratively turns off an interface. To restart the interface, use the **no shutdown** command.



IP Addresses on Router Interfaces

You need physical street addresses to identify the locations of specific homes and companies so that mail can reach those real-world locations efficiently. In the same way, each interface on a Cisco router must have its own IP address to uniquely identify it on the network. If no IP address is configured, even if the interface is in the "up/up" state, the router will not attempt to send and receive IP packets on the interface. To attain proper operation, for every interface that a router should use for forwarding IP packets, the router needs an IP address.

The configuration of an IP address on an interface is relatively simple. To configure the address and mask, simply use the **ip address** *ip-address mask* interface subcommand. The following example shows the configuration of an IP address on the serial interface of a router.



The specific steps to configure an interface on a Cisco router are as follows:

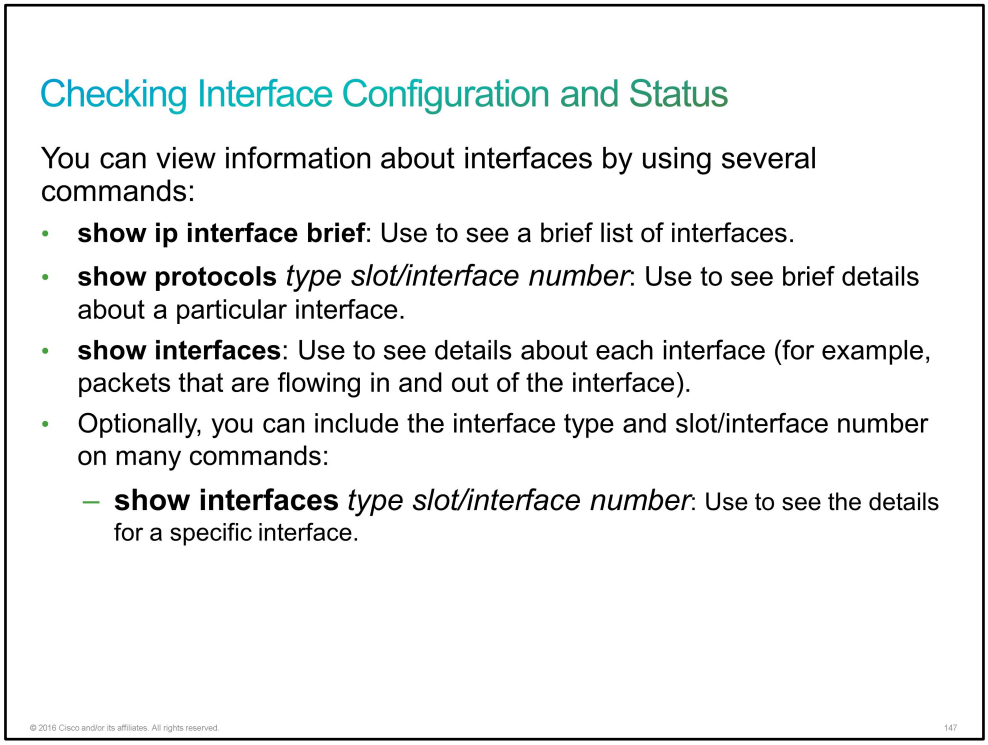
Configuration of an IP Address on the Serial Interface of a Router

Step	Action	Results and Notes
1	Enter the global configuration mode using the **configure terminal** command: Router# **configure terminal**	Displays a new prompt: Router(config)#
2	Identify the specific interface that requires an IP address by using the **interface** *type module/slot/port* command: Router(config)# **interface Serial 0/0/0**	Displays a new prompt; for example: Router(config-if)#

Step	Action	Results and Notes
3	Set the IP address and subnet mask for the interface by using the **ip address** *ip-address mask* command: Router(config-if)# **ip address 172.18.0.1 255.255.0.0**	Configures the IP address and subnet mask for the selected interface
4	Enable the interface to change the state from "administratively down" to "up" by using the **no shutdown** command: Router(config-if)# **no shutdown**	Enables the current interface
5	Exit the configuration mode of the interface by using the **exit** command: Router(config-if)# **exit**	Displays the global configuration mode prompt: Router(config)#

Checking Interface Configuration and Status

When you have completed the router interface configuration, you can verify the configuration by using various **show** commands.



The following examples show sample outputs from the presented commands.

```
RouterY# show ip  interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0        10.1.1.1          YES unset    up                      up
FastEthernet0/1        unassigned      YES unset   administratively    down down
Serial0/0/0                unassigned      YES unset   administratively    down down
Serial0/0/1                unassigned      YES unset   up                      up
Serial0/1/0                unassigned      YES unset   up                      up
Serial0/1/1                unassigned      YES unset   administratively    down down
```

The following table shows the output fields and their meanings.

Output Field	Description
Interface	Type of interface.
IP Address	IP address that is assigned to the interface.
OK?	"Yes" means that the IP address is valid. "No" means that the IP address is not valid.
Method	Describes how the IP address was obtained or configured.

Output Field	Description
Status	Shows the status of the interface.
Protocol	Shows the operational status of the routing protocol on this interface.

```
RouterX# show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is f866.f231.7250 (bia f866.f231.7250)
  Description: Link to ISP
  Internet address is 192.168.2.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full Duplex, 100Mbps, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:53, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
<... output omitted ...>
```

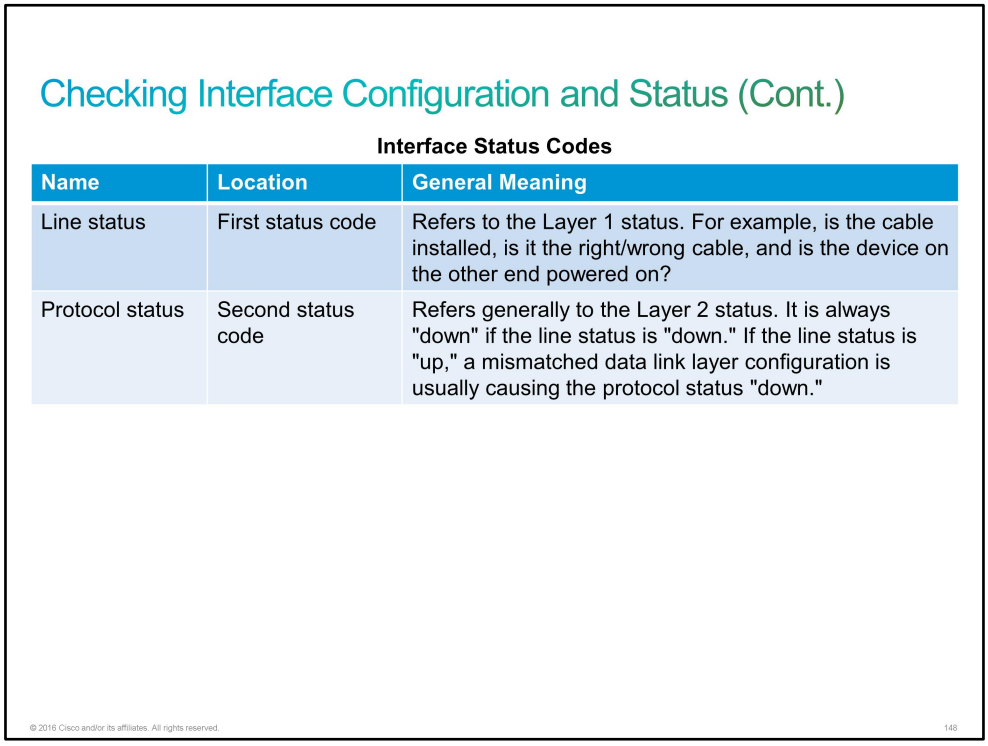
The following table shows some of the output fields for a Gigabit Ethernet interface and their meanings.

Output	Description
GigabitEthernet...is {up | down | administratively down}	Indicates whether the interface hardware is currently active, down, or if an administrator has taken it down.
Line protocol is {up | down}	Indicates whether the software processes that manage the line protocol consider the interface usable (that is, whether keepalives are successful). If the interface misses three consecutive keepalives, the line protocol is marked as down.
Hardware	Displays the hardware type and MAC address.
Description	Displays the configured interface description.
Internet address	Displays the IP address followed by the prefix length (subnet mask).
MTU	Displays the MTU of the interface.
BW	Shows the bandwidth of the interface in kilobits per second. The bandwidth parameter is used to compute routing protocol metrics and other calculations.
DLY	Shows the delay of the interface in microseconds.
Rely	Displays the reliability of the interface as a fraction of 255 (255/255 is 100-percent reliability), which is calculated as an exponential average over 5 minutes.

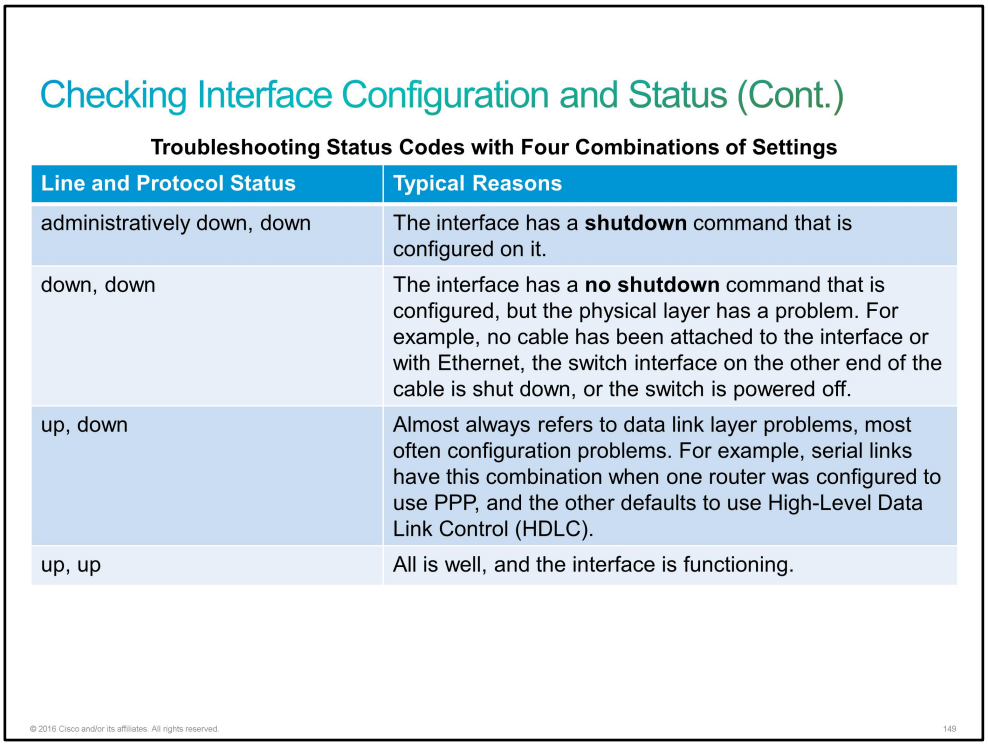
Output	Description
Load	Displays the load on the interface as a fraction of 255 (255/255 is completely saturated), which is calculated as an exponential average over 5 minutes.
Encapsulation	Shows the encapsulation method that is assigned to an interface.
5 minute input rate, 5 minute output rate	Shows the average number of bits and packets that the interface transmitted per second in the last 5 minutes.

Note By truncating the words, you can significantly shorten the commands that refer to router interfaces. For example, you can use **sh int fa0/0** instead of **show interfaces Fastethernet0/0**.

Each of the command outputs that the example shows lists two interface status codes. For a router to use an interface, the two interface status codes on the interface must be in the "up" state. The first status code refers essentially to whether the Layer 1 is working, and the second status code mainly (but not always) refers to whether the data link layer protocol is working. The following table summarizes these two status codes.



Four combinations of settings exist for the status codes when troubleshooting a network. The following table lists the four combinations, along with an explanation of the typical reasons of why an interface would be in this state. As you review the list, note that if the line status (the first status code) is not "up," the second will always be "down" because the data link layer functions cannot work if the physical layer has a problem.



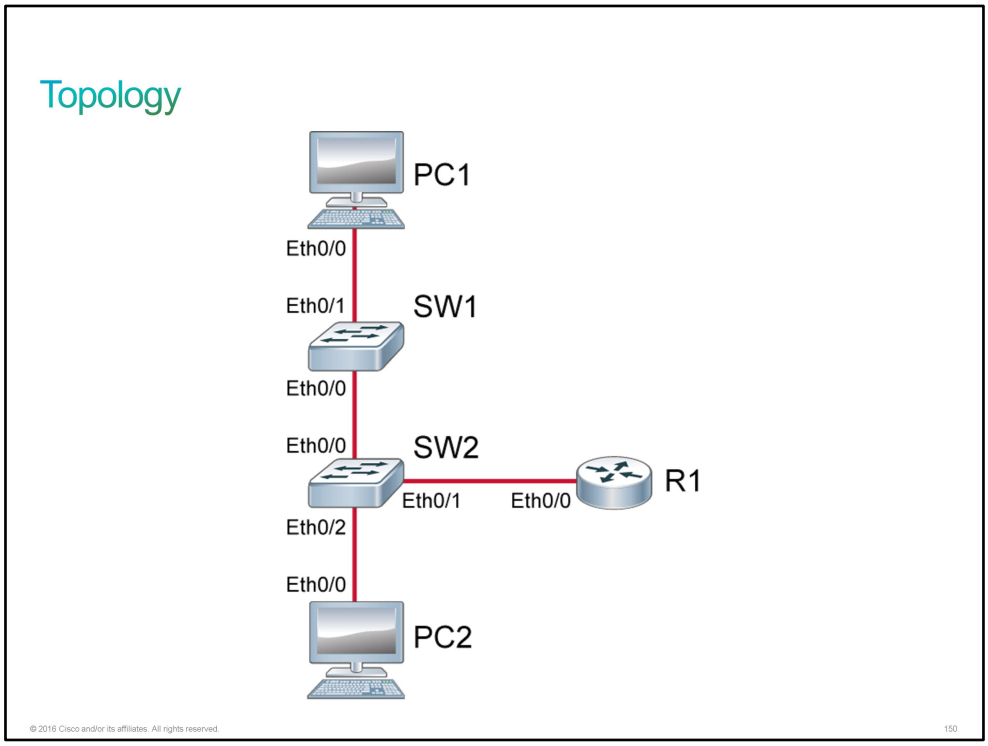
Note	Note that the **show protocols** command is not available in all versions of Cisco IOS Software.

Discovery 6: Start with Cisco Router Configuration

Introduction

This discovery lab will guide you through the configuration of an interface on a Cisco IOS router. The lab is prepared with the devices that are represented in the topology diagram and in the connectivity table. In general, the devices are fully configured. An exception is the interface Ethernet0/0 on R1. You will configure that interface now.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1

Device	Characteristic	Value
PC2	Hostname	PC2
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Not configured
R1	Ethernet0/0 IP address	Not configured
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Configure an IP Address on the Router Interfaces

Activity

Step 1 Access the console of R1 and enter the global configuration mode.

On R1, enter the following command:


```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#
```

Step 2 Enter the interface configuration mode for Ethernet0/0, configure 10.10.1.1/24 as its IP address, add an interface description, and then enable the interface.

On R1, enter the following commands:

```
R1(config)# interface Ethernet 0/0
R1(config-if)# ip address 10.10.1.1 255.255.255.0
R1(config-if)# description Link to SW2
R1(config-if)# no shutdown
```

Examine the IP routing table on R1. You should see the IP address (L - local) and IP subnet (C - connected) that you have just configured on the Ethernet0/0.

```
R1(config-if)# do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
C        10.10.3.0/24 is directly connected, Loopback0
L        10.10.3.1/32 is directly connected, Loopback0
```

Because you are in the interface configuration mode, you will use the **show** command with the **do** option.

Step 3 Use the **do** command to execute an EXEC mode **ping** command. Attempt to ping PC1 (10.10.1.10). The attempt should succeed.

On R1, enter the following command:

```
R1(config-if)# do ping 10.10.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.10, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1(config-if)#
```

It is not uncommon for the first one or two ICMP echo requests to time out. It is usually due to delays that are associated with updating the ARP cache with the MAC address of the local peer.

Step 4 Leave the global configuration mode.

On R1, enter the following command:

```
R1(config-if)# end
R1#
```

Task 2: Verify Interface Configuration and Status

Activity

Step 1 On R1, display the running configuration for the Ethernet0/0 interfaces.

On R1, enter the following command:

```
R1# show running-config interface Ethernet 0/0
Building configuration...

Current configuration : 90 bytes
!
interface Ethernet0/0
 description Link to SW2
 ip address 10.10.1.1 255.255.255.0
end
```

Step 2 On R1, display a brief summary of the IP information and the statuses of all interfaces.

On R1, enter the following command:

```
R1# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES manual up                    up
Ethernet0/1                unassigned      YES NVRAM  administratively down
down
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                unassigned      YES NVRAM  administratively down
down
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  unassigned      YES NVRAM  administratively down
down
Serial1/2                  unassigned      YES NVRAM  administratively down
down
Serial1/3                  unassigned      YES NVRAM  administratively down
down
Loopback0                  10.10.3.1       YES NVRAM  up                    up
```

Step 3 On the R1 router, display the status and statistics of the Ethernet0/0 interface.

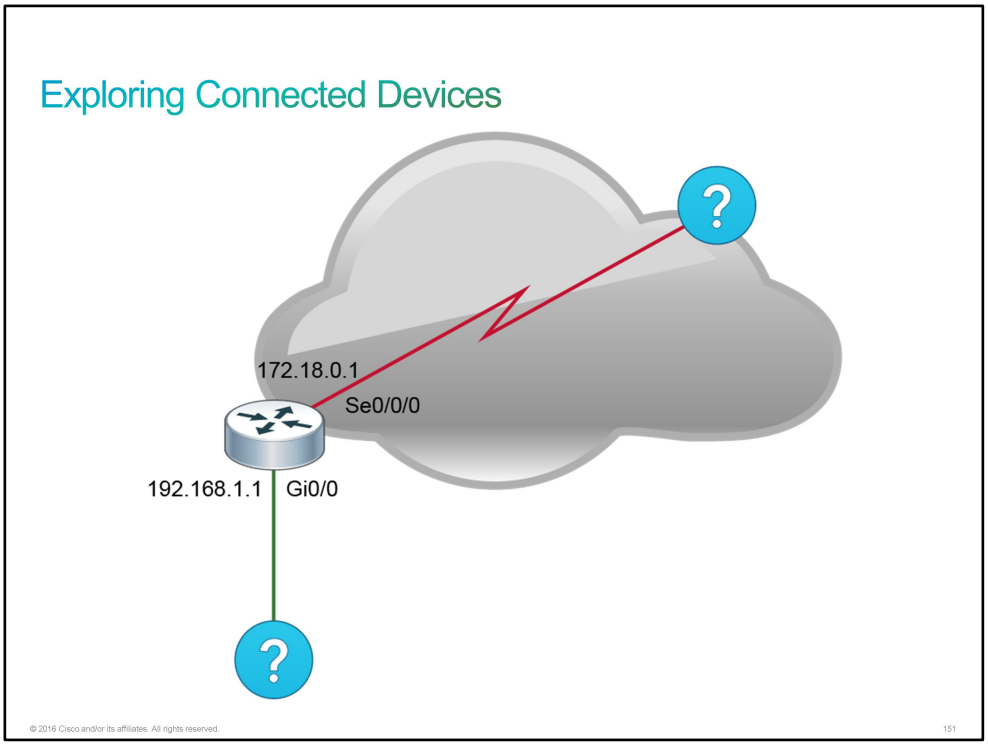
On R1, enter the following command:


```
R1# show interfaces Ethernet 0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.1800 (bia aabb.cc00.1800)
  Description: Link to SW2
  Internet address is 10.10.1.1/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:06, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1116 packets input, 71557 bytes, 0 no buffer
     Received 947 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     218 packets output, 23872 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     3 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

This is the end of the discovery lab.

Exploring Connected Devices

Most network devices, by definition, do not work in isolation. A Cisco device frequently has other Cisco devices as neighbors on the network. If you are able to obtain information about those other devices, it can help you with any network design decisions, troubleshooting, and completing equipment changes.



If you do not have any documentation about the network topology or if the existing documentation is not up to date, you may find yourself in a position of needing to discover the neighboring devices of a router. You can sometimes do this procedure manually by inspecting the physical wiring if the devices are installed next to each other. When neighboring devices are in other buildings or cities, you must use a different method.

One possibility is to use a dynamic discovery protocol that gathers information about directly connected devices. Cisco devices support Cisco Discovery Protocol, which provides information about directly connected Cisco devices and their functions and capabilities.

Cisco Discovery Protocol is a Cisco proprietary protocol that discovers basic information about neighboring Cisco devices without needing to know the passwords for the neighboring devices. To discover information, routers and switches send Cisco Discovery Protocol messages out each of their interfaces. The messages essentially announce information about the device that sent the Cisco Discovery Protocol message. Devices that support Cisco Discovery Protocol learn information about other devices by listening for the advertisements that these devices send.

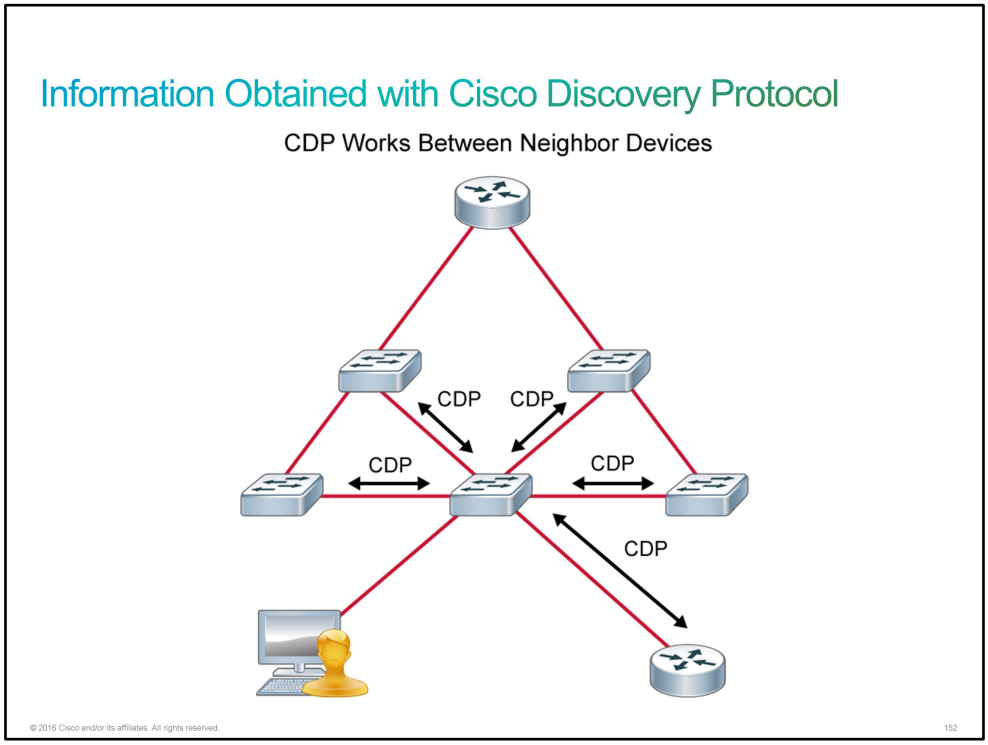
From a troubleshooting perspective, you can use Cisco Discovery Protocol to either confirm or fix the documentation that a network diagram shows or even discover the devices and interfaces that a network uses. Confirming that the network is actually cabled to match the network diagram is a good step to take before trying to predict the normal flow of data in a network.

On media that support multicasts at the data link layer, Cisco Discovery Protocol uses multicast frames; on other media, Cisco Discovery Protocol sends a copy of the Cisco Discovery Protocol update to any known data link addresses. So, any Cisco Discovery Protocol-supporting device that shares a physical medium with another Cisco Discovery Protocol-supporting device can learn about the other device.

Note	Another dynamic discovery protocol is LLDP, which is a standardized, vendor-independent discovery protocol that discovers neighboring devices from different vendors. The IEEE standardized this protocol as the 802.1AB standard. LLDP performs functions that are similar to Cisco Discovery Protocol.

Information Obtained with Cisco Discovery Protocol

The following figure displays an example of how Cisco Discovery Protocol (CDP) exchanges information with its directly connected neighbors. You can display the results of this information exchange on a console that is connected to a network device that is configured to run Cisco Discovery Protocol on its interfaces.



Cisco Discovery Protocol provides the following information about each neighboring device:

- **Device identifiers:** For example, the configured host name of the switch
- **Address list:** Up to one network layer address for each protocol that is supported
- **Port identifier:** The name of the local port and remote port, in the form of an ASCII character string such as Ethernet0
- **Capabilities list:** Supported features—for example, the device acting as a source-route bridge and also as a router
- **Platform:** The hardware platform of the device; for example, Cisco 7200 Series Routers

Notice that the upper router in the previous figure is not connected directly to the console of the administrator. To obtain Cisco Discovery Protocol information about this upper router from the console of the administrator, network staff could use Telnet to connect to a switch that is connected directly to this target device.

Using Cisco Discovery Protocol

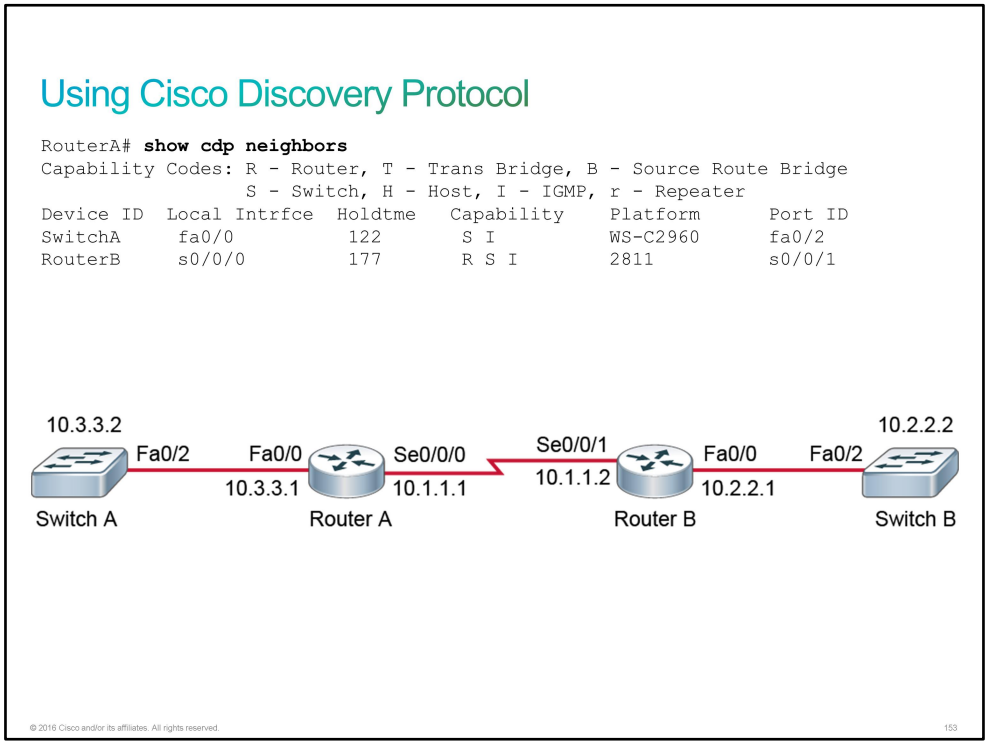
You can enable or disable Cisco Discovery Protocol on a router as a whole (global) or on a port-by-port (interface) basis. You can also view Cisco Discovery Protocol information with the **show cdp** command. Cisco Discovery Protocol has several keywords that enable access to different types of information and different levels of detail. The following example shows the different **show cdp** options.

```
RouterA# show cdp ?
  entry      Information for specific neighbor entry
  interface  CDP interface status and configuration
  neighbors  CDP neighbor entries
  traffic    CDP statistics
```

The Cisco Discovery Protocol functionality is enabled by default on all interfaces (except for Frame Relay multipoint subinterfaces), but you can disable this functionality at the device level. However, some interfaces, such as ATM interfaces, do not support Cisco Discovery Protocol. To prevent other Cisco Discovery Protocol-capable devices from accessing information about a specific device, you use the **no cdp run** global configuration command. To disable Cisco Discovery Protocol on an interface, use the **no cdp enable** command. To enable Cisco Discovery Protocol on an interface, use the **cdp enable** interface configuration command.

```
RouterA(config)# no cdp run
! Disable CDP Globally
RouterA(config)# interface serial0/0/0
RouterA(config-if)# no cdp enable
! Disable CDP on just this interface
```

The **show cdp neighbors** command displays information about Cisco Discovery Protocol neighbors. The following example shows the Cisco Discovery Protocol output for Router A.



For each Cisco Discovery Protocol neighbor, the interface displays the following information:

- Device ID
- Local interface
- Holdtime value, in seconds
- Device capability code
- Hardware platform
- Remote port ID

The holdtime value indicates how long the receiving device should hold the Cisco Discovery Protocol packet before discarding it.

The format of the **show cdp neighbors** output varies among different types of devices, but the available information is generally consistent across devices.

You can use the **show cdp neighbors** command on a Cisco Catalyst switch to display the Cisco Discovery Protocol updates that the switch receives on the local interfaces. Note that on a switch, the local interface is referred to as the local port.

If you add the **detail** argument to the **show cdp neighbors** command, the resulting output includes additional information, such as the network layer addresses of neighboring devices. The output from the **show cdp neighbors detail** command is identical to the one that the **show cdp entry *** command produces, as shown here.

```
Device ID: RouterB
Entry address(es):
  IP address: 10.1.1.2
Platform: Cisco 2811,  Capabilities: Router Switch IGMP
Interface: Serial0/0/0,  Port ID (outgoing port): Serial0/0/1
Holdtime : 155 sec
Version :
Cisco IOS Software, 2800 Software (C2800NM-ADVIPSERVICESK9-M), Version 12.4(12),
RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Fri 17-Nov-06 12:02 by prod_rel_team
```

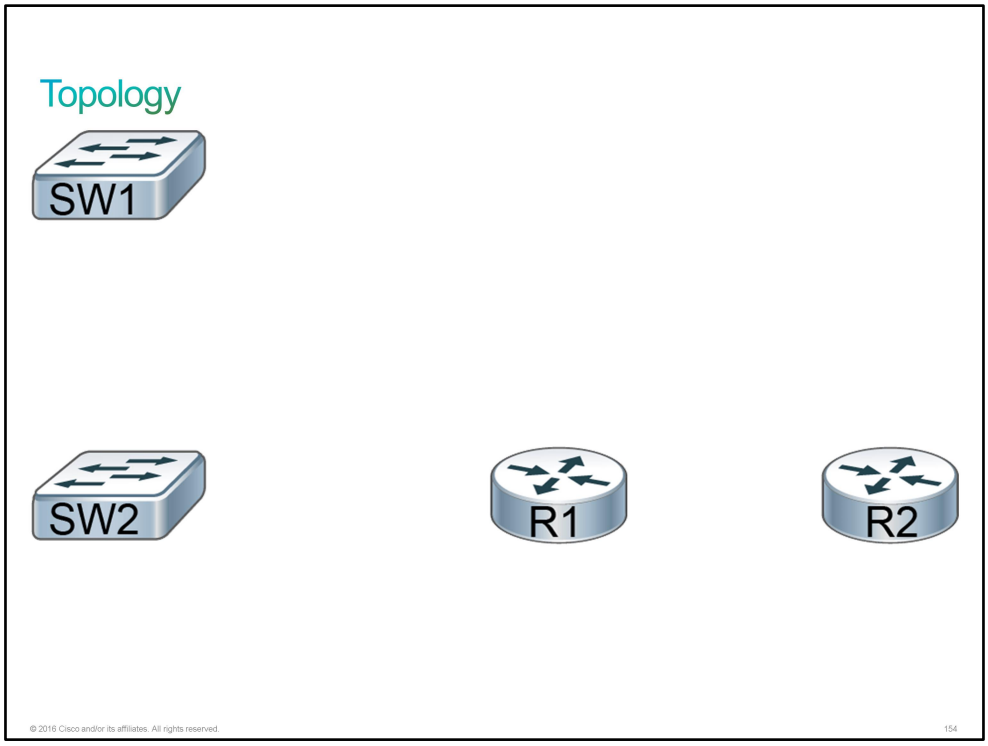
Note	Cisco Discovery Protocol is limited to gathering information about the directly connected Cisco neighbors. Other tools, such as Telnet, are available for gathering information about remote devices that are not directly connected.

Discovery 7: Configure Cisco Discovery Protocol

Introduction

During this discovery lab, you will use Cisco Discovery Protocol to map the connectivity within an unfamiliar network. There are four devices in the topology and you have access to their console ports, but you do not know how they are connected. Using Cisco Discovery Protocol commands, you will determine the actual topology.

Topology



Job Aid

There is no Job Aid available for this lab exercise because the objective of the lab is to map the connectivity within an unfamiliar network.

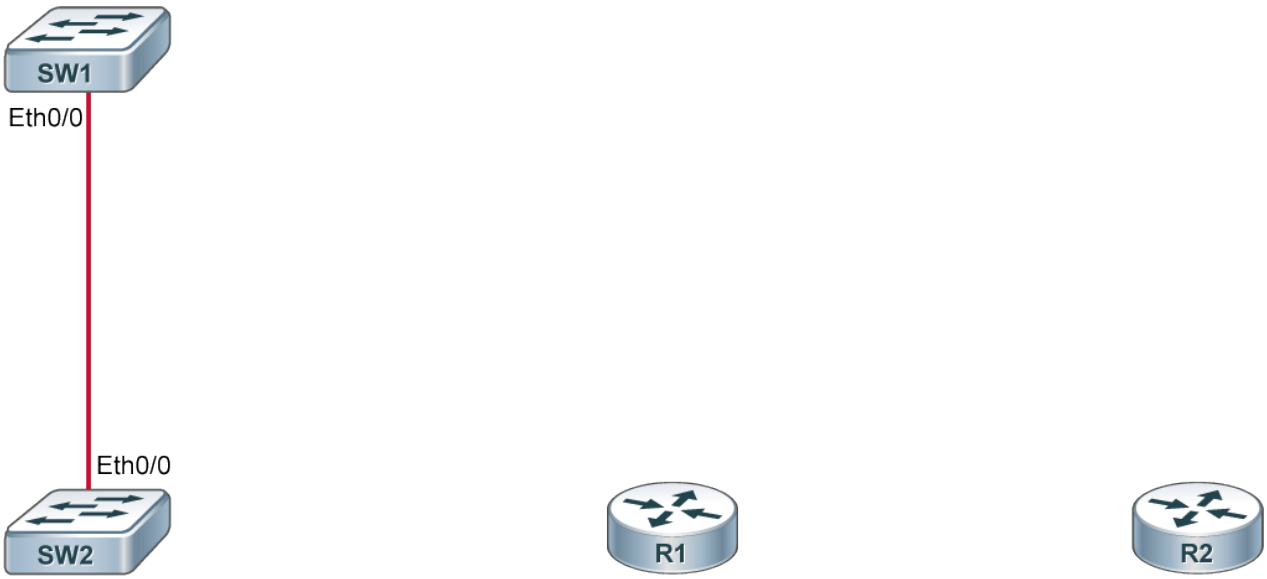
Task 1: Discover Neighbors Using Cisco Discovery Protocol

Activity

Step 1 Before accessing the console of SW1, wait 60 second for Cisco Discovery Protocol to populate its database. On SW1, use the **show cdp neighbor** command to determine the devices to which SW1 is connected. Note both the local port and the port on the remote device.

On SW1, enter the following command:

```
SW1# show cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW2              Eth 0/0           153              S I   Linux Uni Eth 0/0
```

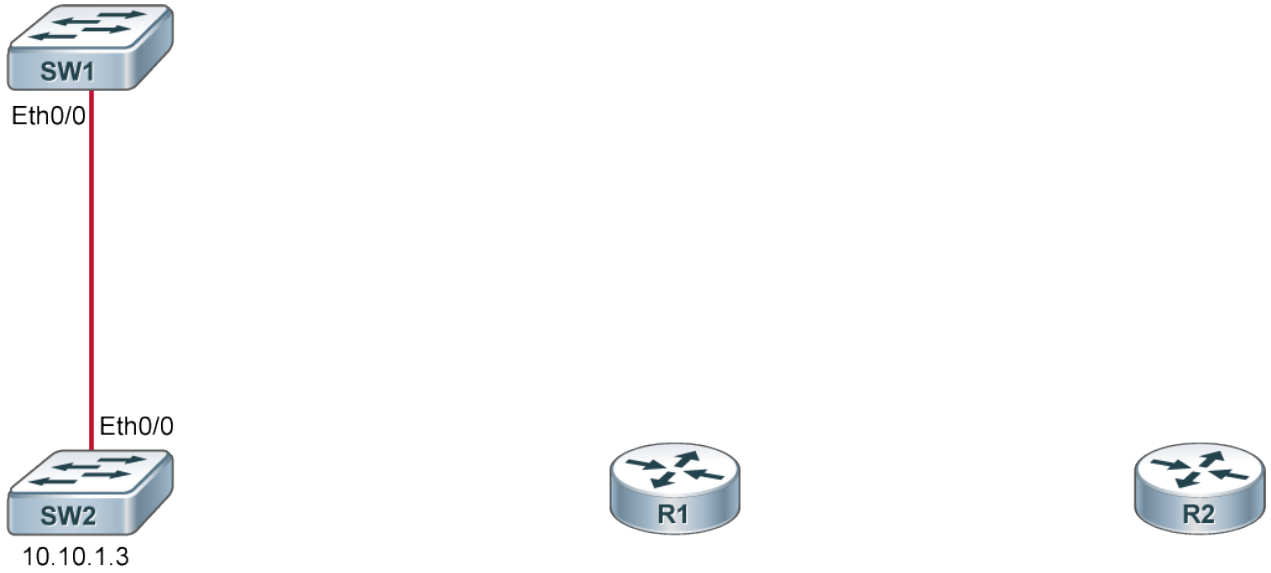


Step 2 Execute the **show cdp neighbor** command again, but this time using the **detail** argument. What is the IP address of SW2?

On SW1, enter the following command:

```
SW1# show cdp neighbor detail
-------------------------
Device ID: SW2
Entry address(es):
  IP address: 10.10.1.3
Platform: Linux Unix,  Capabilities: Switch IGMP
Interface: Ethernet0/0,  Port ID (outgoing port): Ethernet0/0
Holdtime : 147 sec

Version :
Cisco IOS Software, Solaris Software (I86BI_LINUXL2-ADVENTERPRISEK9-M),
Experimental Version 15.1(20130919:231344) [dstivers-sept19-2013pm-team_track
107]
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Thu 19-Sep-13 22:38 by dstivers

advertisement version: 2
VTP Management Domain: ''
Duplex: half
Management address(es):
  IP address: 10.10.1.3
```

The IP address of SW2 is 10.10.1.3.

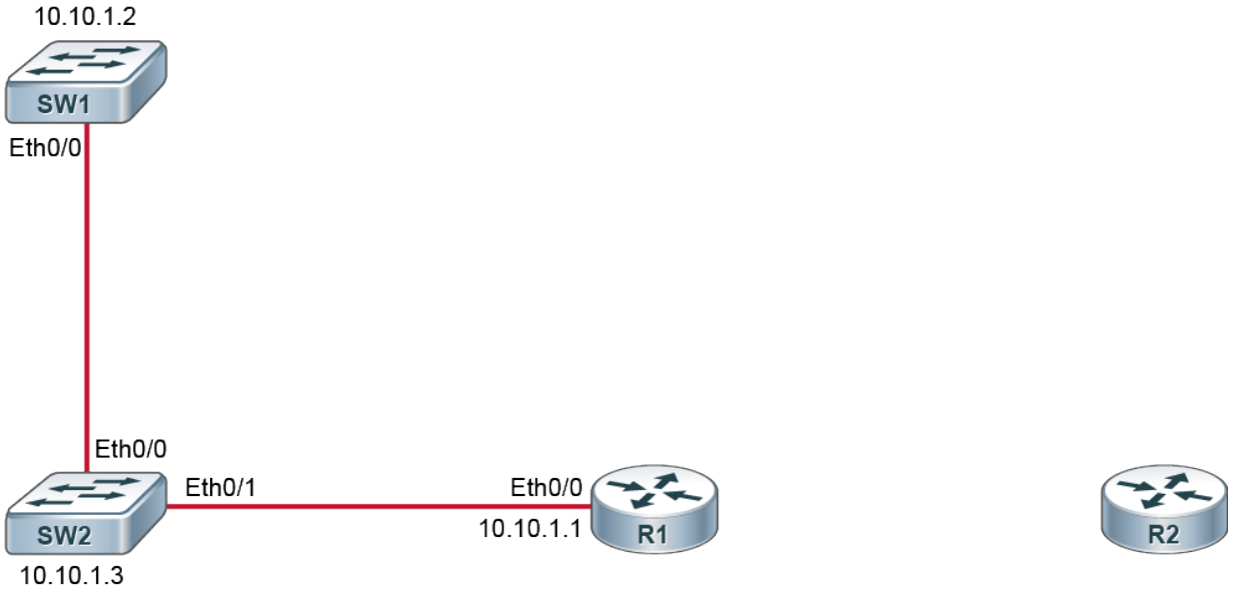


Step 3 Continue the topology inspection from SW2. You know that SW1 is one of the neighbors of SW2. What are the other neighbors of SW2?

On SW2, enter the following command:

```
SW2# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Eth 0/0           130              S I   Linux Uni Eth 0/0
R1               Eth 0/1           153               R    Linux Uni Eth 0/0
SW2# show cdp neighbors detail
-------------------------
Device ID: SW1
Entry address(es):
  IP address: 10.10.1.2
Platform: Linux Unix,  Capabilities: Switch IGMP
Interface: Ethernet0/0,  Port ID (outgoing port): Ethernet0/0
Holdtime : 124 sec

Version :
Cisco IOS Software, Solaris Software (I86BI_LINUXL2-ADVENTERPRISEK9-M),
Experimental Version 15.1(20130919:231344) [dstivers-sept19-2013pm-team_track
107]
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Thu 19-Sep-13 22:38 by dstivers

advertisement version: 2
VTP Management Domain: ''
Duplex: half
Management address(es):
  IP address: 10.10.1.2

-------------------------
Device ID: R1
Entry address(es):
  IP address: 10.10.1.1
Platform: Linux Unix,  Capabilities: Router
Interface: Ethernet0/1,  Port ID (outgoing port): Ethernet0/0
Holdtime : 147 sec

Version :
Cisco IOS Software, Linux Software (I86BI_LINUX-ADVENTERPRISEK9-M), Version
15.2(4)M3, DEVELOPMENT TEST SOFTWARE
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Tue 26-Feb-13 19:06 by prod_rel_team

advertisement version: 2
Duplex: half
Management address(es):
```

The IP address of SW1 is 10.10.1.2 and the IP address of R1 is 10.10.1.1.

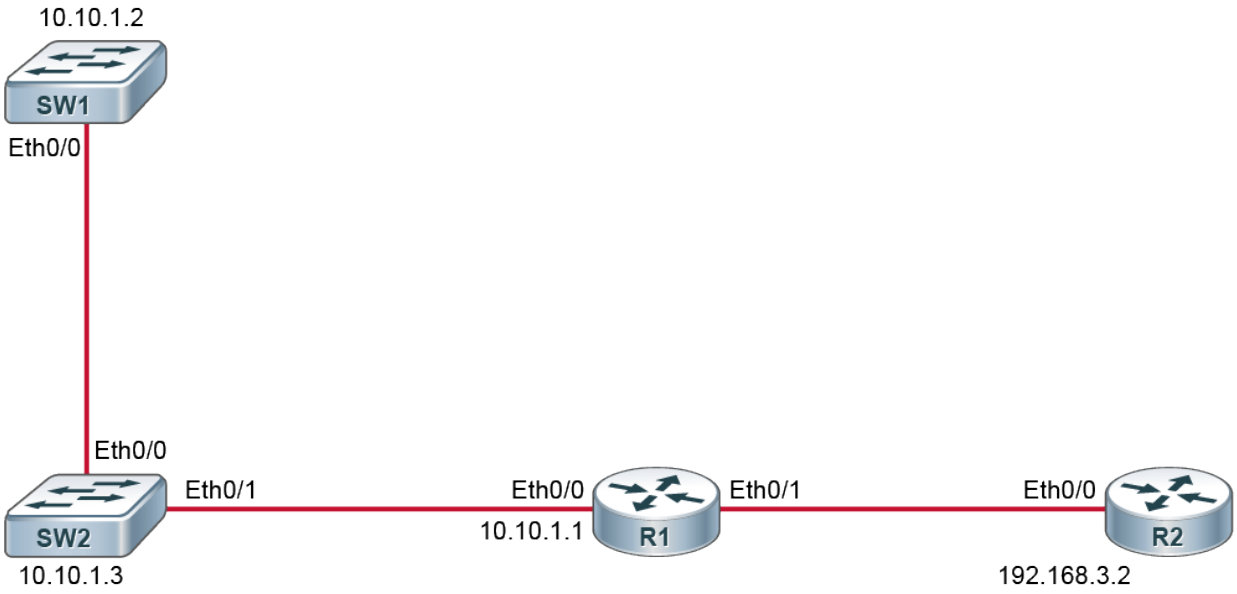


Step 4 Continue the topology inspection from R1. What are the other neighbors of R1?

On R1, enter the following commands:

```
R1# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW2              Eth 0/0            164             S I   Linux Uni Eth 0/1
R2               Eth 0/1            173              R    Linux Uni Eth 0/0
R1# show cdp neighbors  detail
-------------------------
Device ID: SW2
Entry address(es):
  IP address: 10.10.1.3
Platform: Linux Unix,  Capabilities: Switch IGMP
Interface: Ethernet0/0,  Port ID (outgoing port): Ethernet0/1
Holdtime : 161 sec

Version :
Cisco IOS Software, Solaris Software (I86BI_LINUXL2-ADVENTERPRISEK9-M),
Experimental Version 15.1(20130919:231344) [dstivers-sept19-2013pm-team_track
107]
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Thu 19-Sep-13 22:38 by dstivers

advertisement version: 2
VTP Management Domain: ''
Native VLAN: 1
Duplex: half

-------------------------
Device ID: R2
Entry address(es):
  IP address: 192.168.3.2
Platform: Linux Unix,  Capabilities: Router
Interface: Ethernet0/1,  Port ID (outgoing port): Ethernet0/0
Holdtime : 170 sec

Version :
Cisco IOS Software, Linux Software (I86BI_LINUX-ADVENTERPRISEK9-M), Version
15.2(4)M3, DEVELOPMENT TEST SOFTWARE
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Tue 26-Feb-13 19:06 by prod_rel_team

advertisement version: 2
Duplex: half
```

The IP address of R2 is 192.168.3.2.

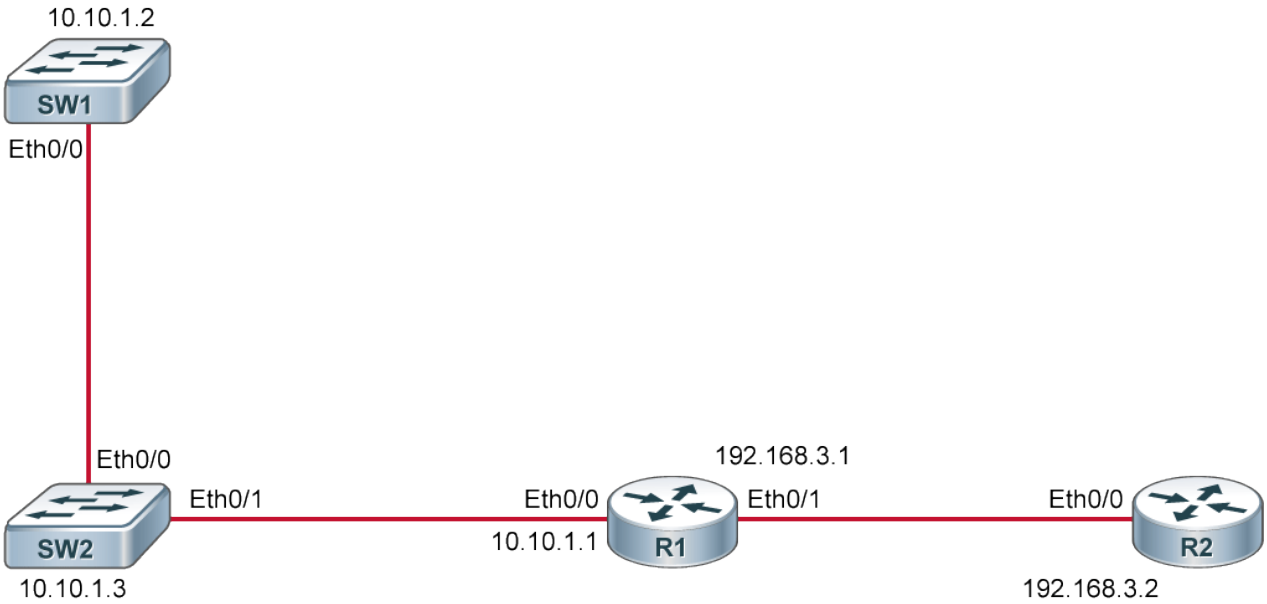


Step 5 Continue the topology inspection from R2.

On R2, enter the following commands:

```
R2# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R1               Eth 0/0            176             R    Linux Uni Eth 0/1
R2# show cdp neighbors detail
-------------------------
Device ID: R1
Entry address(es):
  IP address: 192.168.3.1
Platform: Linux Unix,  Capabilities: Router
Interface: Ethernet0/0,  Port ID (outgoing port): Ethernet0/1
Holdtime : 174 sec

Version :
Cisco IOS Software, Linux Software (I86BI_LINUX-ADVENTERPRISEK9-M), Version
15.2(4)M3, DEVELOPMENT TEST SOFTWARE
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Tue 26-Feb-13 19:06 by prod_rel_team

advertisement version: 2
Duplex: half
```

R2 has no additional neighbors.



This is the end of the discovery lab.

Configuring LLDP

To permit the discovery of non-Cisco devices, the switch also supports LLDP, which is a vendor-neutral device discovery protocol that IEEE 802.1AB standard defines. LLDP allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data link layer, which allows two systems that are running different network layer protocols to learn about each other.

LLDP is a one-way protocol that transmits information about the capabilities and the status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

LLDP supports a set of attributes that it uses to discover other devices. These attributes contain TLV descriptions. LLDP devices can use TLVs to send and receive information to other devices on the network. Using this protocol, devices can advertise details such as configuration information, device capabilities, and device identity.

LLDP advertises the following TLVs by default:

- DCBXP
- Management address
- Port description
- Port VLAN
- System capabilities
- System description
- System name

Configuring LLDP

LLDP has the following configuration guidelines and limitations:

- Must be enabled on the device before you can enable or disable it on any interface
- Is supported only on physical interfaces
- Can discover up to one device per port
- Can discover Linux servers

Configuring LLDP (Cont.)

Enable or disable LLDP globally.

```
[no] lldp run
```

Enable or disable LLDP on an interface.

```
[no] lldp transmit
[no] lldp receive
```


After you globally enable LLDP, it is enabled on all supported interfaces by default. The **lldp transmit** command enables the transmission of LLDP packets on an interface. The **lldp receive** command enables the reception of LLDP packets on an interface.

Challenge

1. Which of the following does the router first perform upon being booted up?

   A. Finds and loads IOS
   B. Finds and applies the configuration statements
   C. Runs POST to test the hardware
   D. Displays command line prompt

2. If POST completes during the startup of a Cisco Device, and there is no configuration file, what does the router do?

   A. It reboots
   B. It initiates initial configuration through 'setup'
   C. It creates a configuration file with default settings and boots that.
   D. It shutdowns the router

3. **show ip interface brief** command displays the packets that are flowing in and out of the interface. True or False.?

   A. True
   B. False

4. Match the correct description about the line and protocol status on the right with conditions listed on the left.

Ethernet0/0 is down, line protocol is down	Interface is shutdown condition
Ethernet0/0 is administratively down, line protocol is down	Interface is in no shutdown condition, but cable is not connected to ethernet interface
Serial0/0/0 is up, line protocol is down	Cable is connected to the interface and placed in no shutdown condition, but encapsulation mismatch encountered on the serial links
Serial 0/0/0 is up, line protocol is up	Interface is functioning well with no issues

5. Which command would you use to enable LLDP globally on a router?

   A. **lldp transmit**
   B. **lldp run**
   C. **lldp receive**
   D. **cdp run**
   E. **cdp enable**

6. CDP can determine the platform version of the neighboring devices. True or False?

   A. True
   B. False

7. Which command displays the IP address of a neighboring device?

   A. **show cdp neighbors**
   B. **show ip interface brief**
   C. **show interfaces**
   D. **show cdp neighbors detail**

Answer Key

Challenge

1. C
2. B
3. B
4.

Ethernet0/0 is administratively down, line protocol is down	Interface is shutdown condition
Ethernet0/0 is down, line protocol is down	Interface is in no shutdown condition, but cable is not connected to ethernet interface
Serial0/0/0 is up, line protocol is down	Cable is connected to the interface and placed in no shutdown condition, but encapsulation mismatch encountered on the serial links
Serial 0/0/0 is up, line protocol is up	Interface is functioning well with no issues

5. B
6. A
7. D

Lesson 6: Exploring the Packet Delivery Process

Introduction

Your boss sends you to your customer to debug problems with undelivered IP packets. You will need to illustrate the role of the Layer 2 address and Layer 3 address in the packet delivery process. You will investigate the role of ARP. At this point, you will need to understand all individual pieces of the packet delivery process.

Address Resolution Protocol

Because a frame must contain a MAC address, there must be a way to resolve an IP address to a MAC address. For example, if you issue the **ping 10.1.1.3** command, the MAC address of 10.1.1.3 must be included in the MAC destination field of the frame. To determine the MAC address of 10.1.1.3, a process is performed by a Layer 2 protocol called ARP.

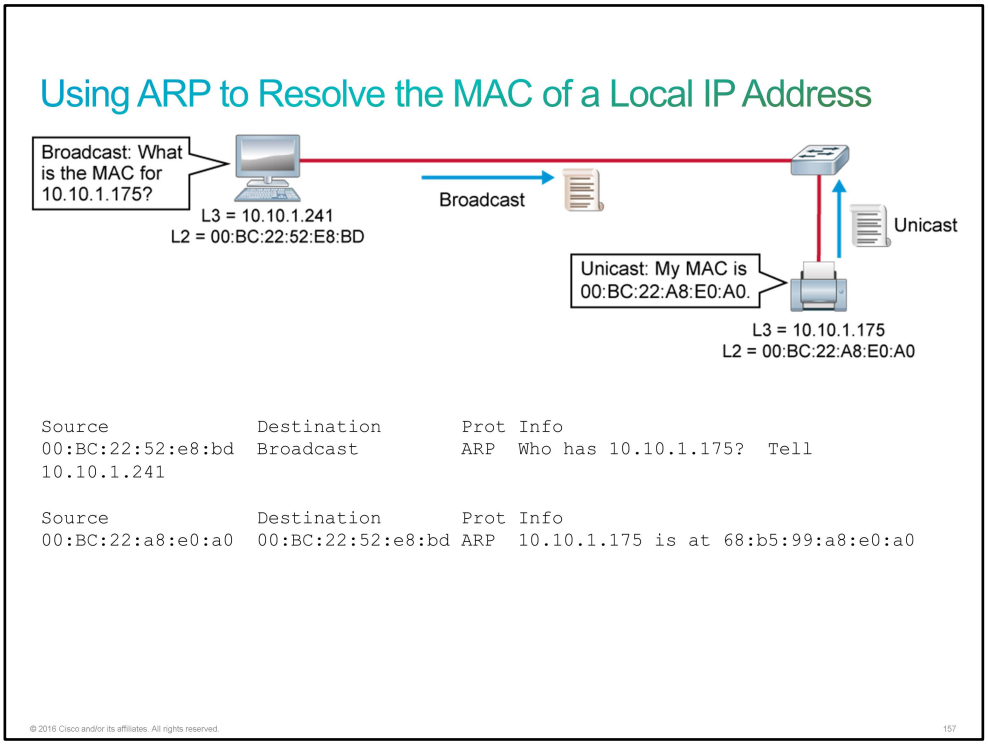
ARP provides two essential services:

- **Address resolution:** Mapping IP addresses to MAC addresses on a network
- **Caching:** Locally storing MAC addresses that are learned via ARP

The term *address resolution* refers to the process of binding the IP address of a remote device to its MAC address. ARP sends a broadcast message to all devices on the local network. This message includes its own IP address and the destination IP address. The message is basically asking the device on which the destination IP address resides to respond with its MAC address. The address resolution procedure is completed when the originator receives the reply packet, which contains the required MAC address, from the target and updates the table containing all the current bindings.

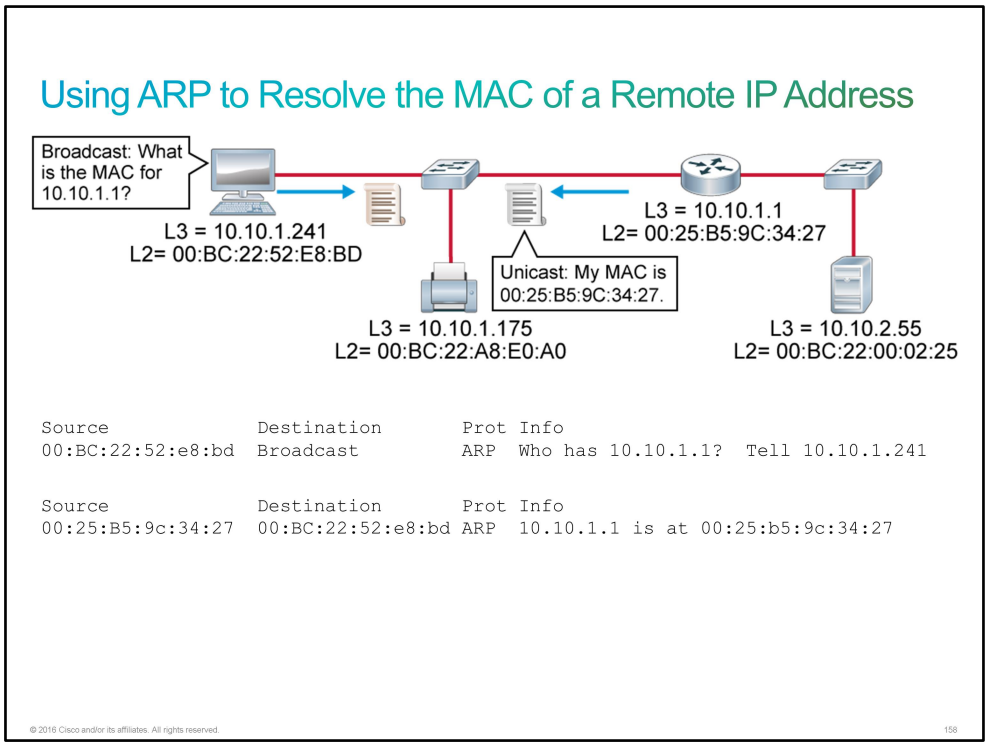
Using ARP to Resolve the MAC of a Local IP Address

Because ARP is a Layer 2 protocol, its scope is limited to the local LAN. An IP host can tell if the IP host that it wants to communicate with is on the same network by comparing the destination IP address against its configured subnet mask. For example, IP host 10.10.1.241/24 is on the 10.10.1.0 network. If the IP host that it wants to communicate with is 10.10.1.175, it knows that this IP host is also on the local 10.10.1.0 network, and can request an ARP for its MAC address directly.



Using ARP to Resolve the MAC of a Remote IP Address

When the IP host 10.10.1.241 wants to communicate with the destination IP host 10.10.2.55, it compares this IP address against its subnet mask and discovers that the host is on a different IP network (10.10.2.0). You know that when a PC wants to send data to a device that is on another network, it sends the data to the default gateway. So the destination MAC address in the frame needs to be the MAC address of the default gateway. In this situation, the IP source must request an ARP for its default gateway. The default gateway is the IP address of the router interface on the local subnet. In the example, IP host 10.10.1.241 sends an ARP broadcast for the MAC address of 10.10.1.1.



Understanding the ARP Cache

Each IP device on a network segment maintains a table in memory—the ARP table. The purpose of this table is to cache recent IP addresses and MAC address bindings. When a host wants to transmit data to another host on the same network, it searches the ARP table to see if there is an entry. If there is an entry, the host uses it. If there is no entry, the IP host sends an ARP broadcast requesting resolution.

Note	By caching recent bindings, ARP broadcasts can be avoided for any mappings in the cache. Without the ARP cache, each IP host would have to send an ARP broadcast each time that it wanted to communicate with another IP host.

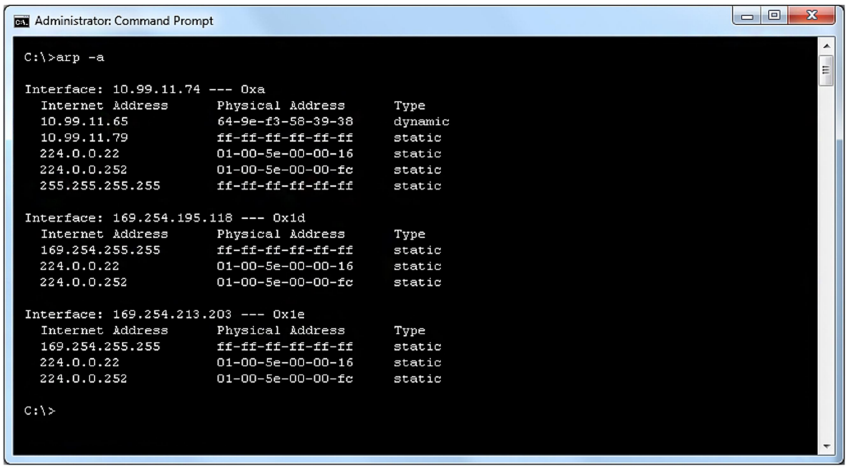
Each entry, or row, of the ARP table has a pair of values—an IP address and a MAC address. The relationship between the two values is a map, which simply means that you can locate an IP address in the table and discover the corresponding MAC address. The ARP table caches the mapping for the devices on the local LAN.

The device creates and maintains the ARP table dynamically. It adds and changes address relationships as they are used on the local host. The entries in an ARP table usually expire after 300 seconds, which is the default value. This short timeout ensures that the table does not contain information for systems that may be switched off or that have been moved. When the local host wants to transmit data again, the entry in the ARP table is regenerated through the ARP process.

If no device responds to the ARP request, the packet is dropped because a frame cannot be created without the destination MAC address.

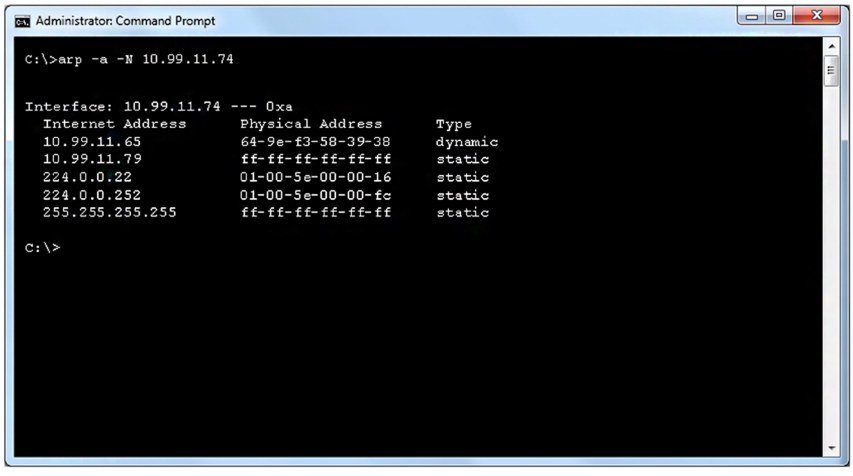
Understanding the ARP Cache

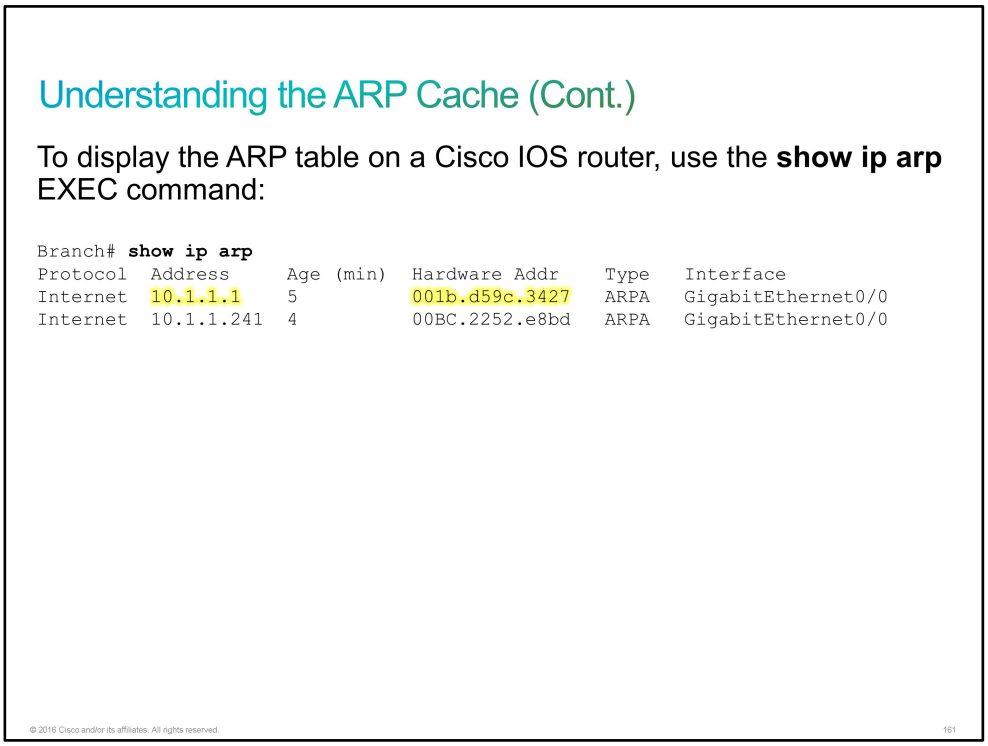
The **arp -a** command displays the current ARP table for all interfaces on a PC using the Microsoft Windows operating system.



Understanding the ARP Cache (Cont.)

To limit the output of the **arp** command to a single interface, use the **arp -a -N** *ip_address* command.





The proper syntax to display the ARP table is **show ip arp** [*ip-address*] [*host-name*] [*mac-address*] [*interface type number*].

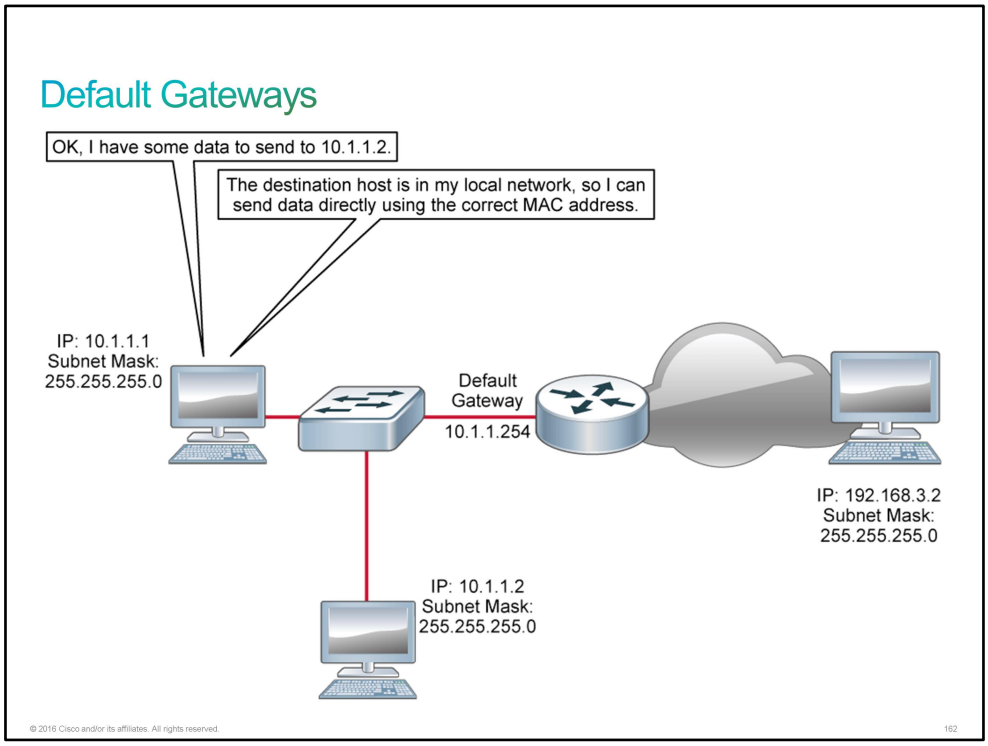
Syntax Description

Parameter	Description
ip-address	(Optional) Displays ARP entries matching this IP address
host-name	(Optional) Hostname
mac-address	(Optional) 48-bit MAC address
interface type number	(Optional) Displays ARP entries that are learned via this interface type and number

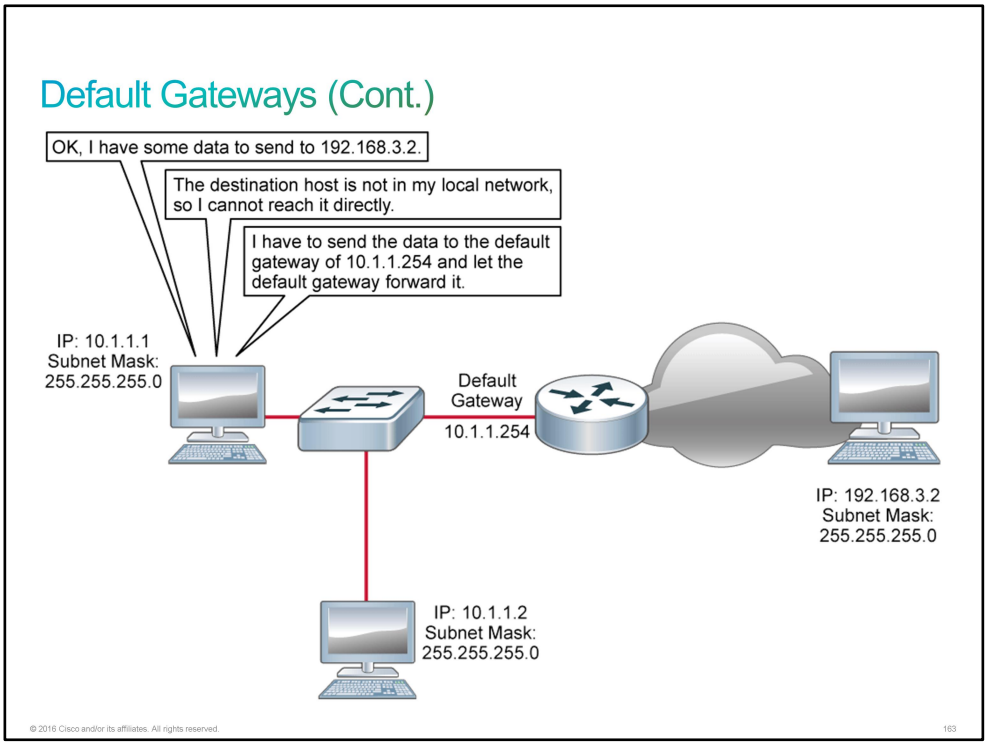
Address Resolution Protocol

Default Gateways

The source host is able to communicate directly with the destination host only if the two hosts are on the same network. If the two hosts are on different networks, the sending host must send the data to the default gateway, which will forward the data to the destination.



Before an end system can send a packet to its destination, it must first determine if the destination address is in the local network. The subnet mask defines the network part of the IP address. The end system compares the network portion of the local network address with the destination network address of the packet to be sent. If the network portion of the local network address is the same as the destination network address, the end system can deliver packets directly. If the network portion of the local network address is not the same as the destination network address, the packets must be forwarded to some other network.



The default gateway is needed to send a packet out of the local network. If the network portion of the destination address of the packet is different from the network of the originating host, the packet has to be routed outside of the original network. To do this, the packet is sent to the default gateway. This default gateway is a router interface that is connected to the local network. The default gateway interface has a network layer address that matches the network address of the hosts. The hosts are configured to recognize that address as the default gateway.

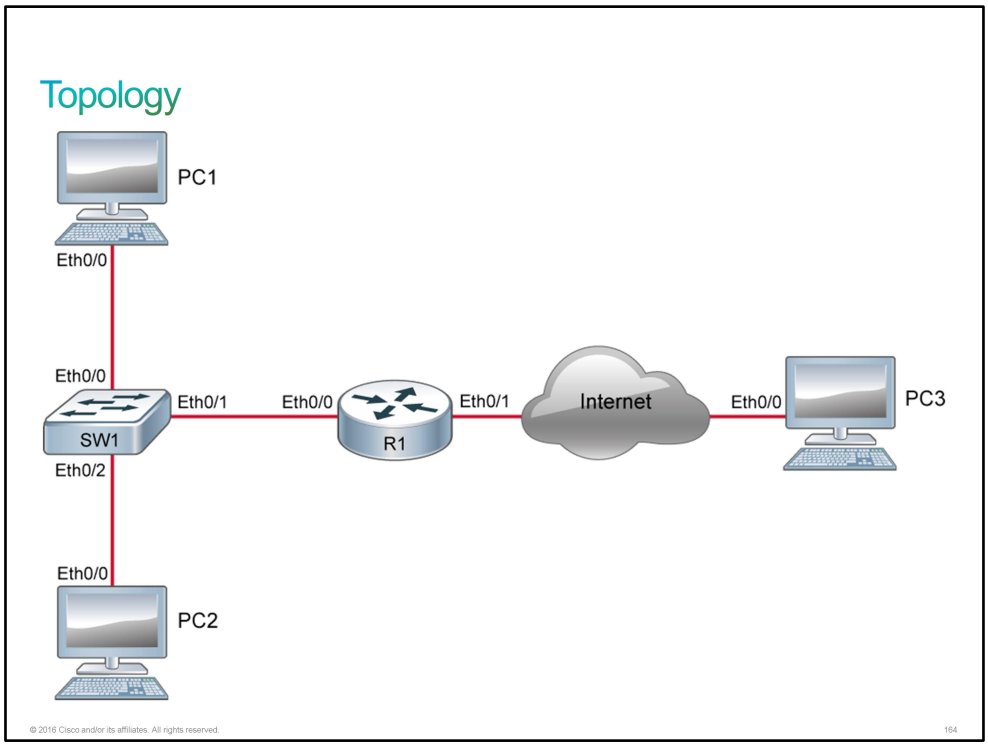
On a Windows computer, the Internet Protocol (TCP/IP) Properties tools are used to enter the default gateway IP address. The host IP address and the default gateway address must have the same network portion of their respective addresses.

Discovery 8: Configure Default Gateway

Introduction

This discovery lab will help you explore how ARP maps IP addresses to MAC addresses and how default gateways allow access to hosts on remote subnets. The lab is prepared with the devices represented in the topology diagram with the IP addresses as depicted in the table. Note that PC1, PC2, PC3, SW1, and R1 are fully configured.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	Not configured
PC2	Hostname	PC2

Device	Characteristic	Value
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
PC3	Hostname	PC3
PC3	IP address	192.168.3.2/24
PC3	Default gateway	192.168.3.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to PC1
SW1	Ethernet0/1 description	Link to R1
SW1	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/1 IP address	192.168.3.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Configure Default Gateway

Activity

Step 1 All networked devices with IP addresses maintain an ARP cache. Via the ARP process, devices learn the MAC address of other hosts on their local subnet for which they need to communicate. Access the console of PC1 and execute the **show arp** command.

PC1 should have an entry for itself (10.10.1.10).

```
PC1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.10             -   aabb.cc00.2200  ARPA   Ethernet0/0
```


If there was activity within the discovery before executing the **show arp** command, you may find that there are other entries in the table.

The command **show arp** does not work on PCs. It is used here because the actual device that is used to simulate a PC is a router.

MAC addresses in your output may be different.

Step 2 To initiate communication between PC1 and other devices on the subnet, which will initiate the ARP process to learn the appropriate MAC addresses, use the **ping** command. Ping PC2 (10.10.1.20), R1 (10.10.1.1) and SW1 (10.10.1.2).

Sometimes the first ping times out due to the delay that the ARP process caused.

```
PC1# ping 10.10.1.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.20, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
PC1# ping 10.10.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/201/1003 ms
PC1# ping 10.10.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1004 ms
```

Step 3 Examine the ARP cache on PC1 again.

The ARP cache is now populated with all four hosts that have IP addresses on the 10.10.1.0/24 subnet.

```
PC1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1               2   aabb.cc00.2100  ARPA   Ethernet0/0
Internet  10.10.1.2               0   aabb.cc80.2a00  ARPA   Ethernet0/0
Internet  10.10.1.10              -   aabb.cc00.2200  ARPA   Ethernet0/0
Internet  10.10.1.20              2   aabb.cc00.2800  ARPA   Ethernet0/0
```

Step 4 From PC1, ping 192.168.3.2, which is a PC on a different subnet.

From PC1, ping PC3:

```
PC1# ping 192.168.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

View the ARP cache on PC1.

```
PC1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1              1   aabb.cc00.2100  ARPA   Ethernet0/0
Internet  10.10.1.2              0   aabb.cc80.2a00  ARPA   Ethernet0/0
Internet  10.10.1.10             -   aabb.cc00.2200  ARPA   Ethernet0/0
Internet  10.10.1.20             1   aabb.cc00.2800  ARPA   Ethernet0/0
Internet  192.168.3.2            0   aabb.cc00.2100  ARPA   Ethernet0/0
```

There is an ARP cache entry for 192.168.3.2. The MAC address for 192.168.3.2 and 10.10.1.1 are identical. This behavior is the result of the Proxy ARP feature which is enabled on IOS routers by default. PC1 does not have a default gateway that is configured, so it attempts to ARP for all addresses. R1 saw the ARP request for a remote address which was available in its routing table, and sent an ARP reply with its own MAC address. PC1 can then forward traffic that is destined to 192.168.3.2 to R1's MAC address and R1 will forward as necessary. While Proxy ARP can be helpful as a last resort, properly configuring a default gateway is a better practice.

Step 5 Verify that PC1 does not have a default route in its routing table.

On PC1, enter the following command:

```
PC1# show ip route
Default gateway is not set

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

Step 6 Configure R1 as the default gateway for PC1.

On PC1, enter the following command:

```
PC1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)# ip default-gateway 10.10.1.1
PC1(config)# end
PC1#
```

Verify the routing table on PC1 again:

```
PC1# show ip route
Default gateway is 10.10.1.1

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

Step 7 Remove the entry for 192.168.3.2 from the ARP cache of PC1 and verify that the entry has been removed.

On PC1, enter the following commands:


```
PC1# clear ip arp 192.168.3.2
PC1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1               6   aabb.cc00.2100  ARPA   Ethernet0/0
Internet  10.10.1.2               6   aabb.cc80.2a00  ARPA   Ethernet0/0
Internet  10.10.1.10              -   aabb.cc00.2200  ARPA   Ethernet0/0
Internet  10.10.1.20              6   aabb.cc00.2800  ARPA   Ethernet0/0
```

Step 8 Ping 192.168.3.2 and verify that there is no entry for 192.168.3.2 in the ARP cache of PC1.

On PC1, enter the following commands:

```
PC1# ping 192.168.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
PC1# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1               8   aabb.cc00.2100  ARPA   Ethernet0/0
Internet  10.10.1.2               8   aabb.cc80.2a00  ARPA   Ethernet0/0
Internet  10.10.1.10              -   aabb.cc00.2200  ARPA   Ethernet0/0
Internet  10.10.1.20              8   aabb.cc00.2800  ARPA   Ethernet0/0
```

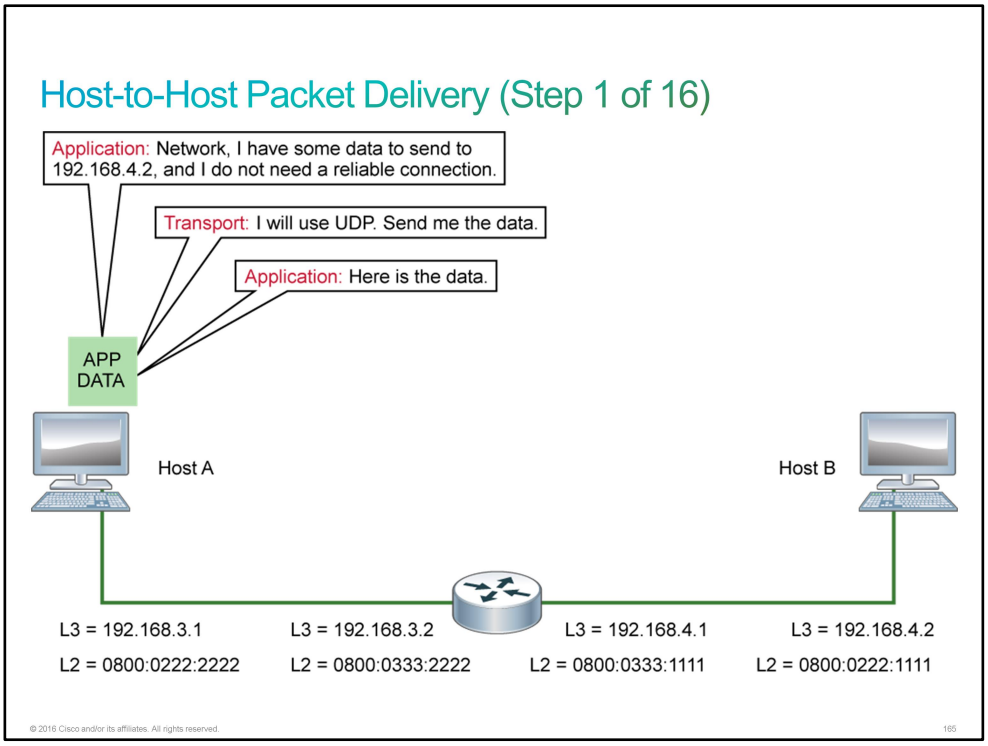
For all addresses outside of the 10.10.1.0/24 subnet, PC1 will use the destination MAC address of 10.10.1.1 (R1, its default gateway). R1 will then forward the packet appropriately due to its routing table.

This is the end of the discovery lab.

Host-to-Host Packet Delivery

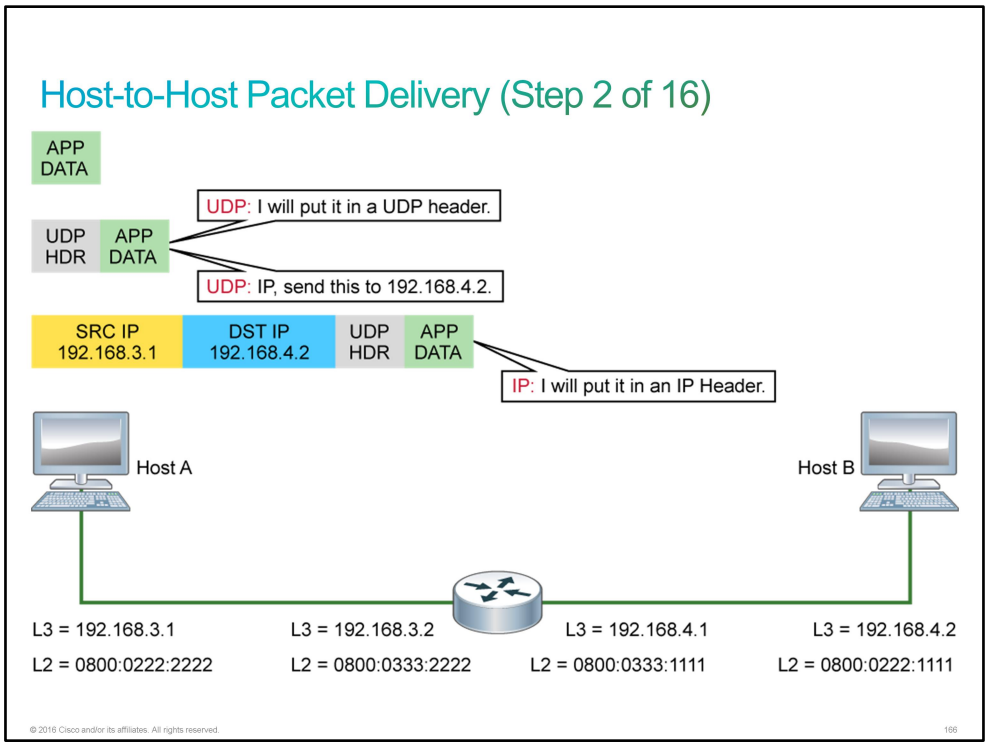
Host-to-host packet delivery consists of an interesting series of processes. In this multipart example, you will discover what happens "behind the scenes" when an IP host communicates with another IP host. The IP host 192.168.3.1 needs to send arbitrary application data to the IP host 192.168.4.2, which is located on another subnet.

Host-to-Host Packet Delivery (Step 1 of 16)



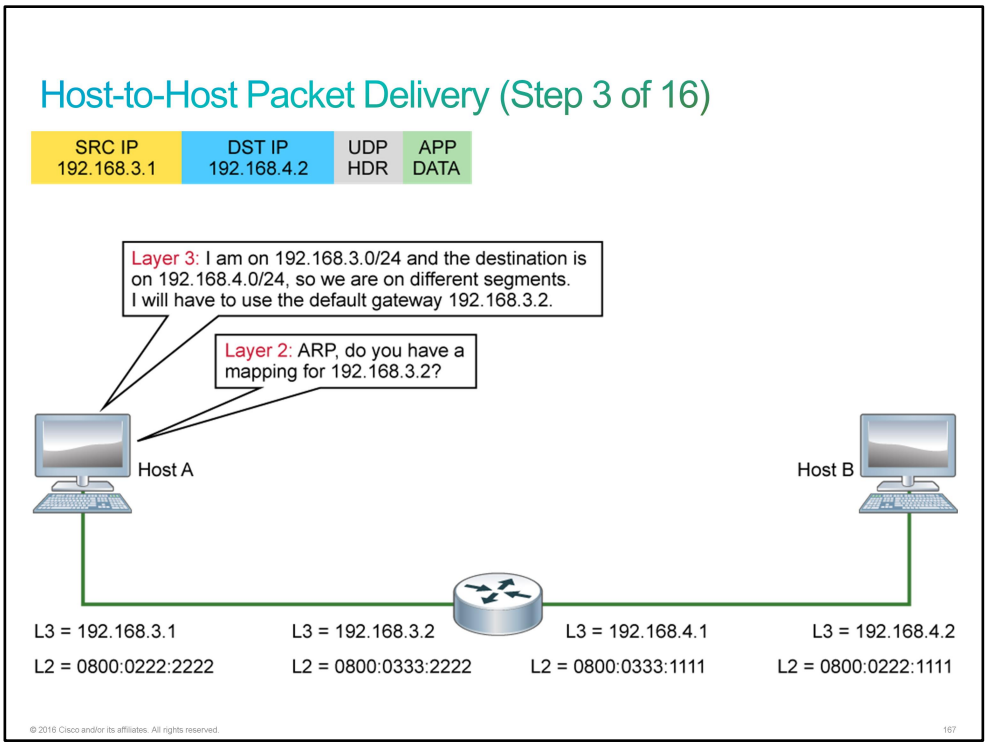
In this example, the host 192.168.3.1 has data that it wants to send to the host 192.168.4.2. The application does not need a reliable connection, so it uses UDP. Because it is not necessary to set up a session, the application can start sending data.

Host-to-Host Packet Delivery (Step 2 of 16)



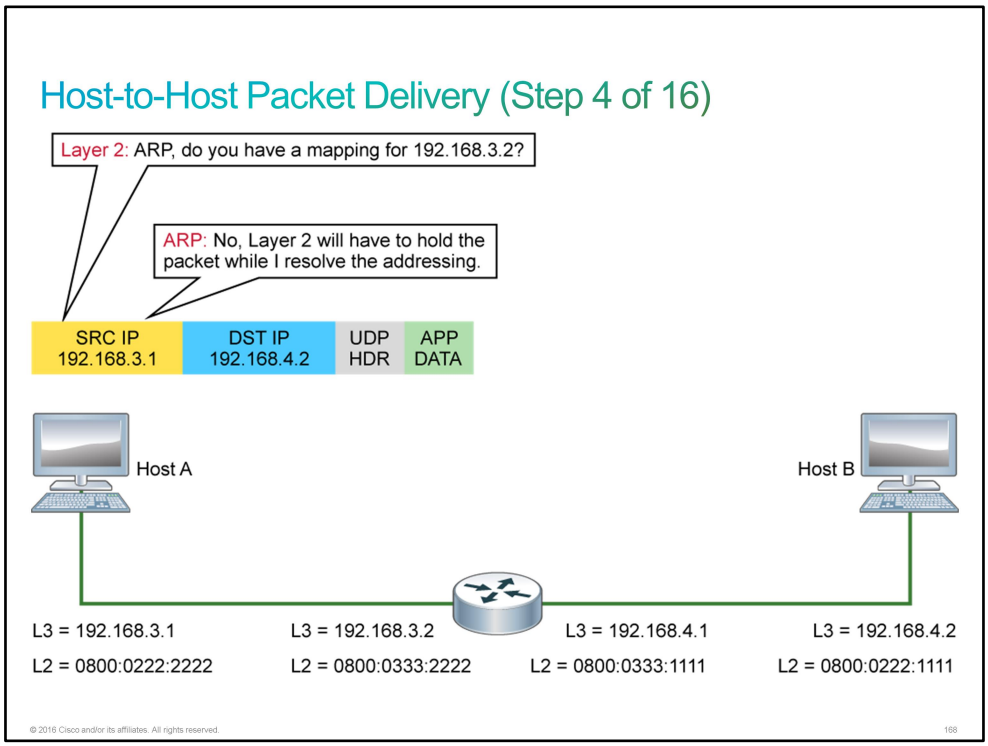
UDP prepends a UDP header (UDP HDR) and passes the PDU to the IP (Layer 3) with an instruction to send the PDU to 192.168.4.2. IP encapsulates the PDU in a Layer 3 packet, setting the source IP address (SRC IP) of the packet to 192.168.3.1, while the destination IP address (DST IP) is set to 192.168.4.2.

Host-to-Host Packet Delivery (Step 3 of 16)



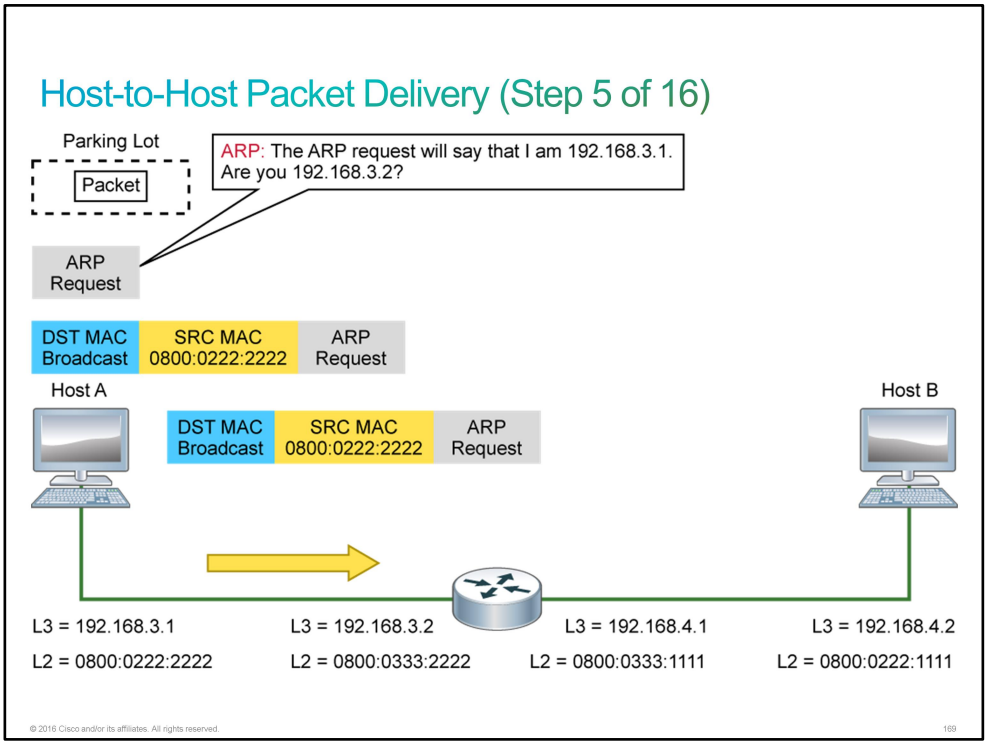
When Host A analyzes the destination address, it finds that the destination address is on a different network. The host sends any packet that is not destined for the local IP network to the default gateway. The default gateway is the address of the local router, which must be configured on hosts (PCs, servers, and so on). IP encapsulates the PDU in a Layer 3 packet and passes it to Layer 2 with instructions to forward it to the default gateway. Host A must place the packet in its parking lot until it can obtain the needed information that is related to the default network.

Host-to-Host Packet Delivery (Step 4 of 16)



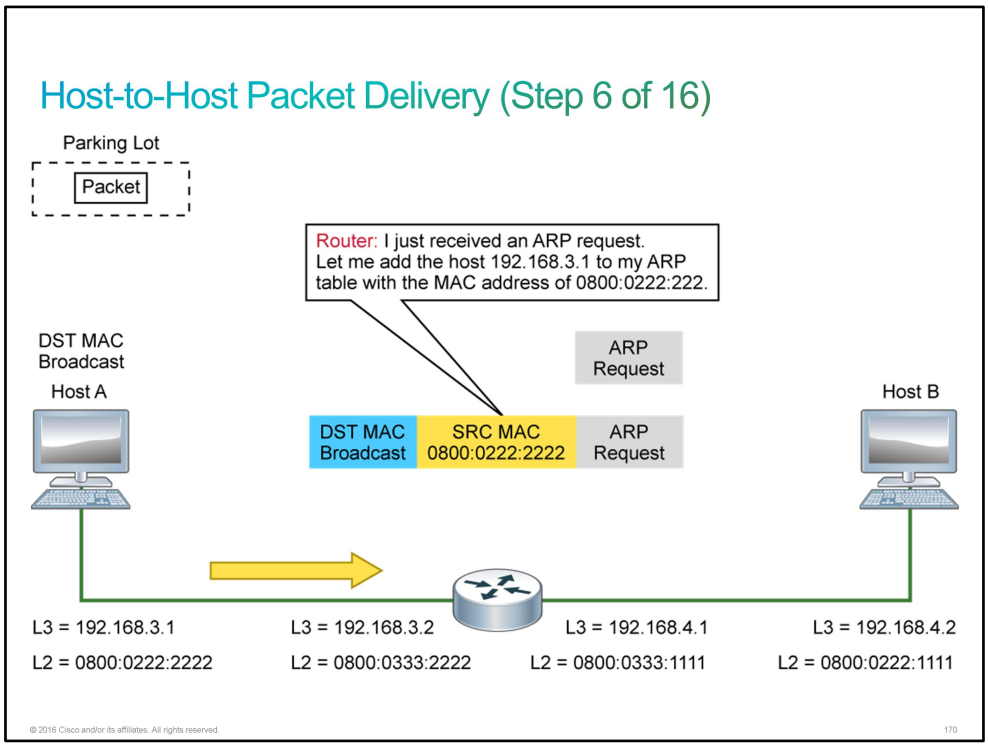
To deliver the packet, the host needs the Layer 2 information of the next-hop device. The ARP table in the host does not have an entry and must resolve the Layer 2 address (MAC address) of the default gateway. The default gateway is the next hop for the packet. The packet waits while the host resolves the Layer 2 information.

Host-to-Host Packet Delivery (Step 5 of 16)



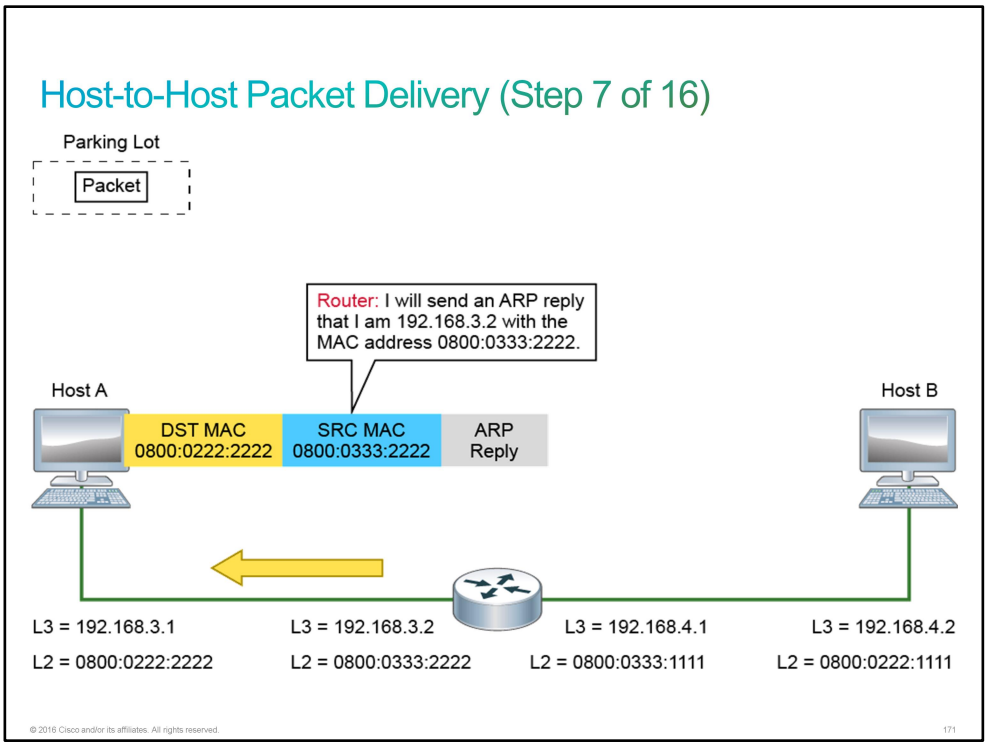
Because the host does not have a mapping of Layer 2 and Layer 3 addresses for the default gateway, the host uses the standard ARP process to obtain the mapping. The host sends an ARP request to the router.

Host-to-Host Packet Delivery (Step 6 of 16)



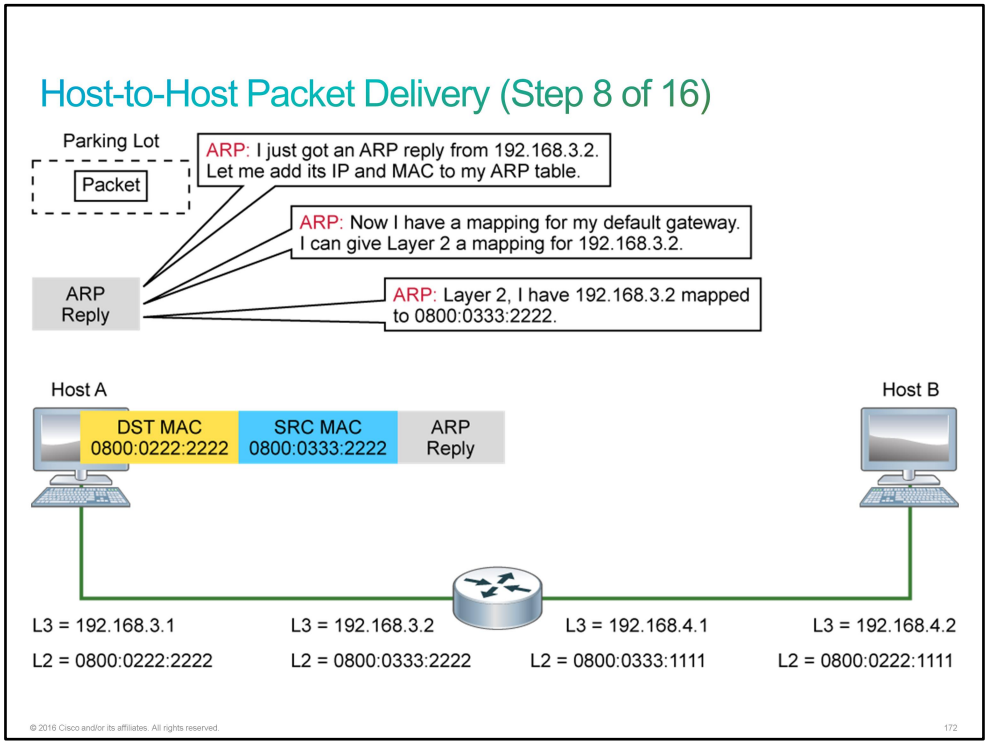
The user has programmed the IP address of 192.168.3.2 as the default gateway. The host 192.168.3.1 sends out the ARP request, and the router receives it. The ARP request contains information about the host, and the router adds the information to its ARP table.

Host-to-Host Packet Delivery (Step 7 of 16)



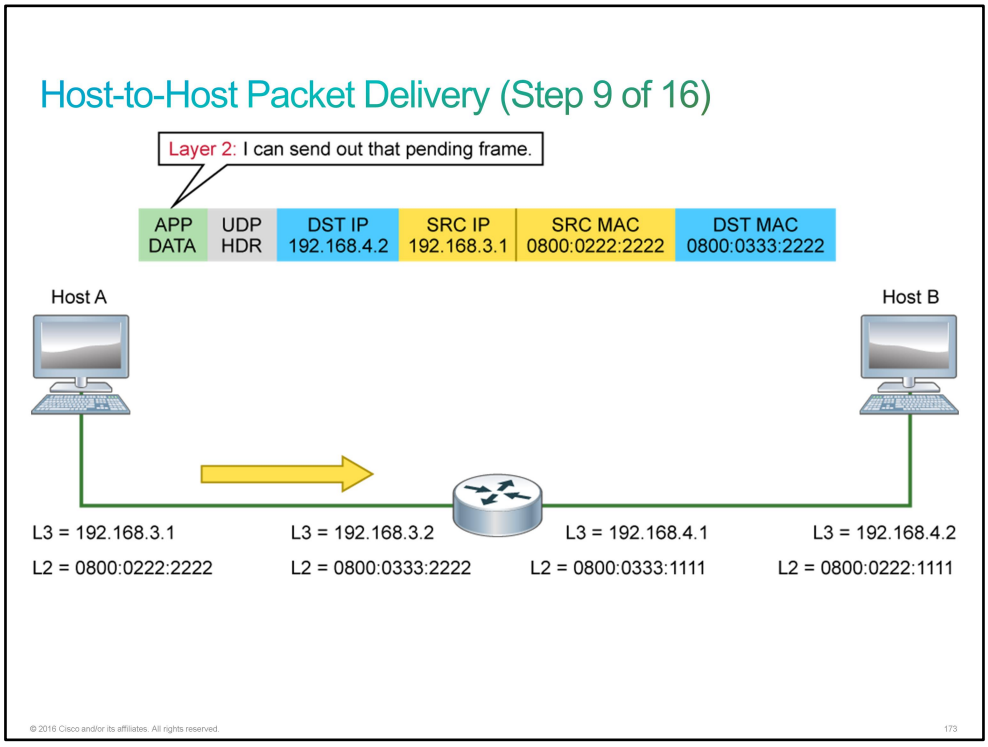
The router processes the ARP request like any other host and sends the ARP reply with its own information.

Host-to-Host Packet Delivery (Step 8 of 16)



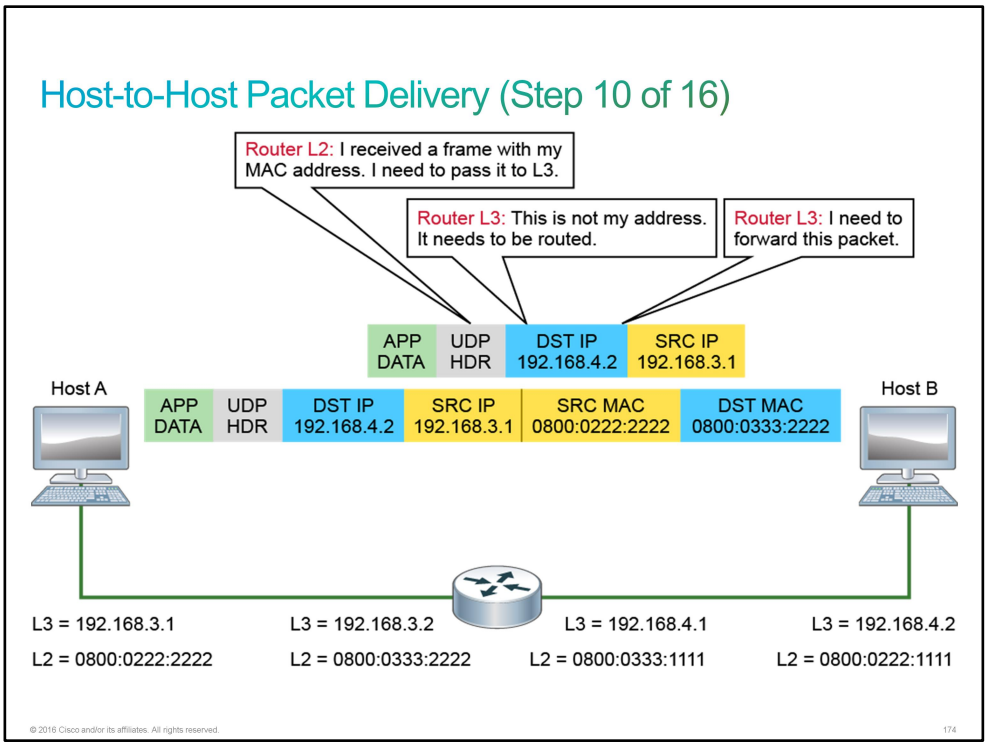
The host receives an ARP reply to the ARP request and enters the information to its local ARP table.

Host-to-Host Packet Delivery (Step 9 of 16)



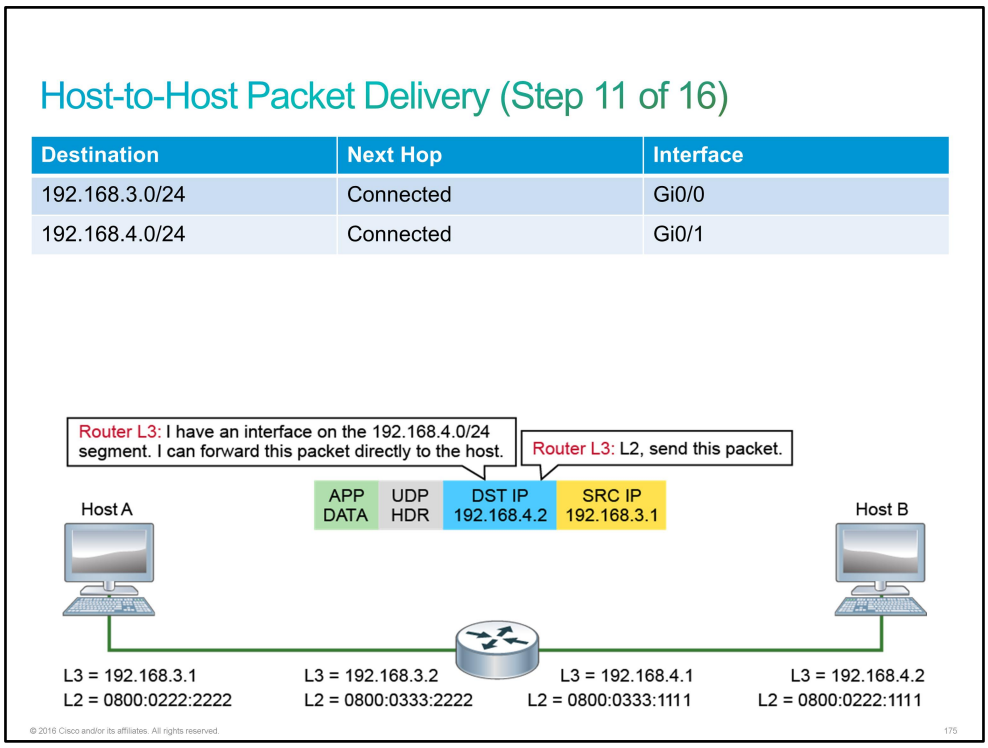
Now the Layer 2 frame with the application data can be sent to the default gateway. The pending frame is sent with the local host IP address and MAC address as the source. However, the destination IP address is that of the remote host, but the destination MAC address is that of the default gateway.

Host-to-Host Packet Delivery (Step 10 of 16)



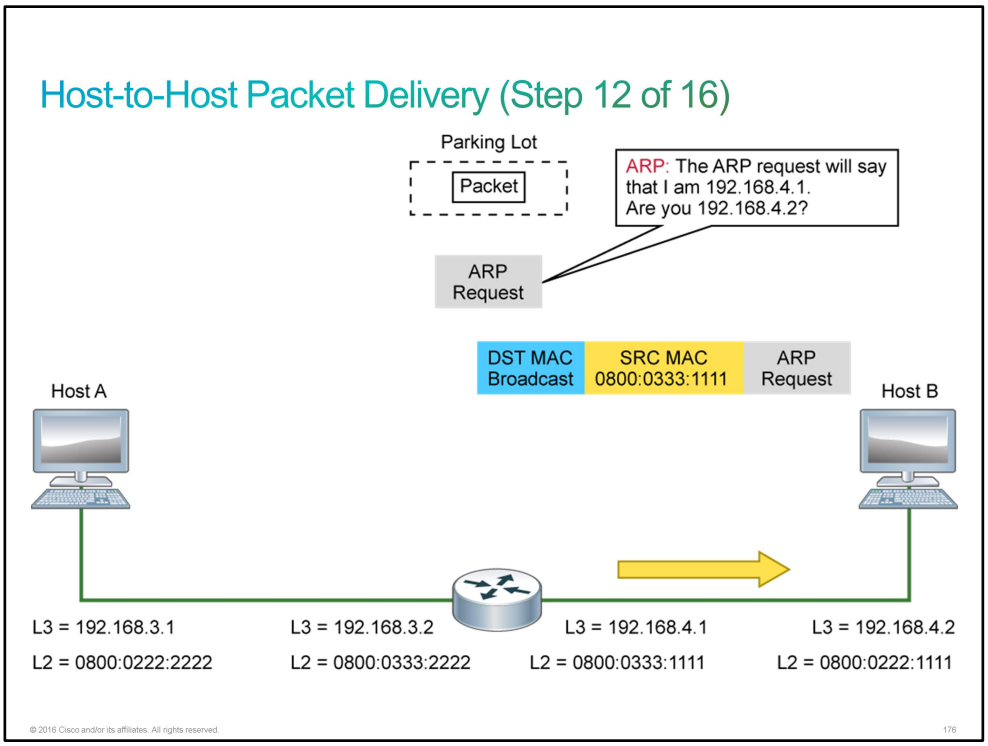
When the router receives the frame, the router recognizes its MAC address and processes the frame. At Layer 3, the router sees that the destination IP address is not its address. A host Layer 3 device would discard the frame. However, because this device is a router, it passes all packets that are for unknown destinations to the routing process. The routing process determines where to send the packet.

Host-to-Host Packet Delivery (Step 11 of 16)



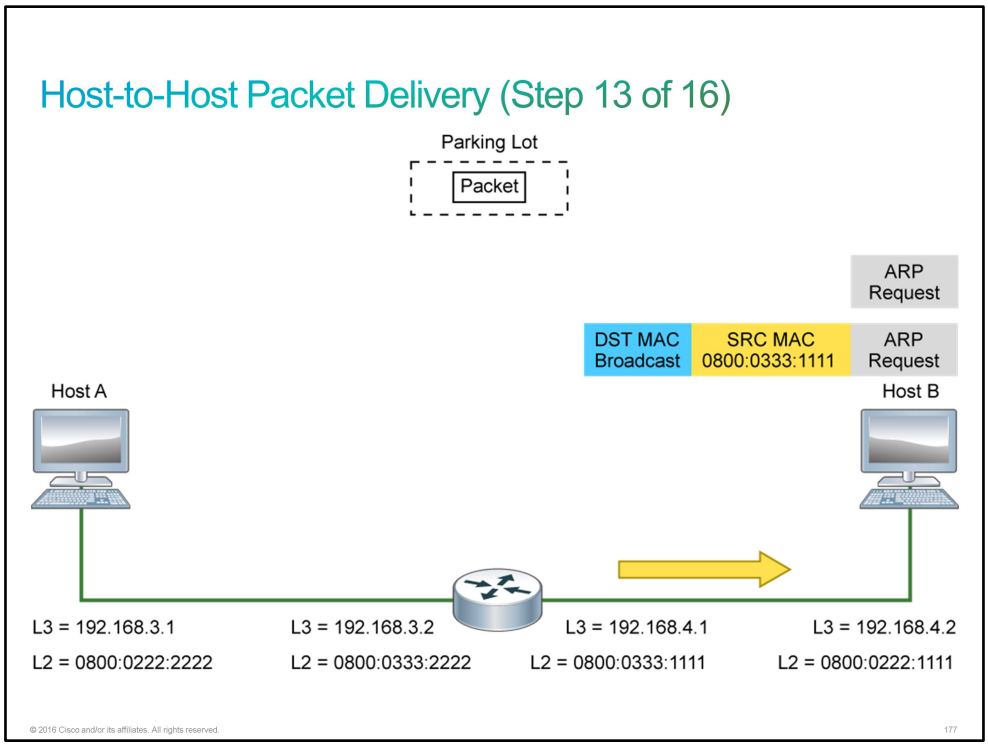
The routing process looks up the destination IP address in its routing table. In this example, the destination segment is directly connected. Therefore, the routing process can pass the packet directly to Layer 2 for the appropriate interface.

Host-to-Host Packet Delivery (Step 12 of 16)



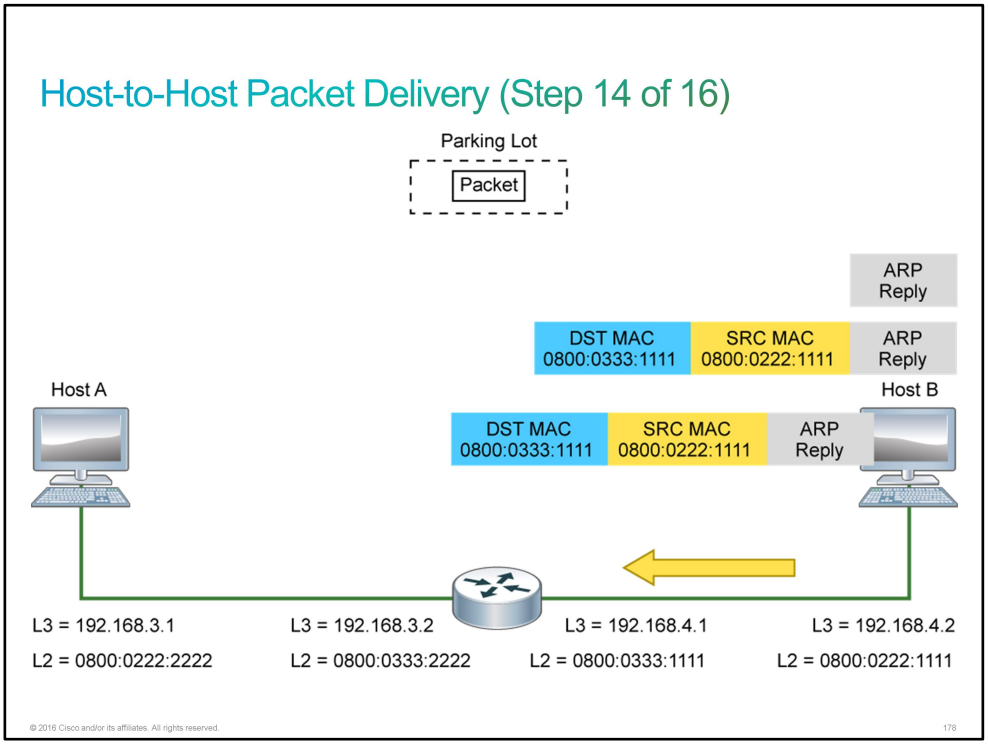
Layer 2 uses the ARP process to obtain the mapping for the IP address and the MAC address. The router asks for the Layer 2 information in the same way as the hosts. An ARP request for the destination Layer 3 address is sent to the link.

Host-to-Host Packet Delivery (Step 13 of 16)



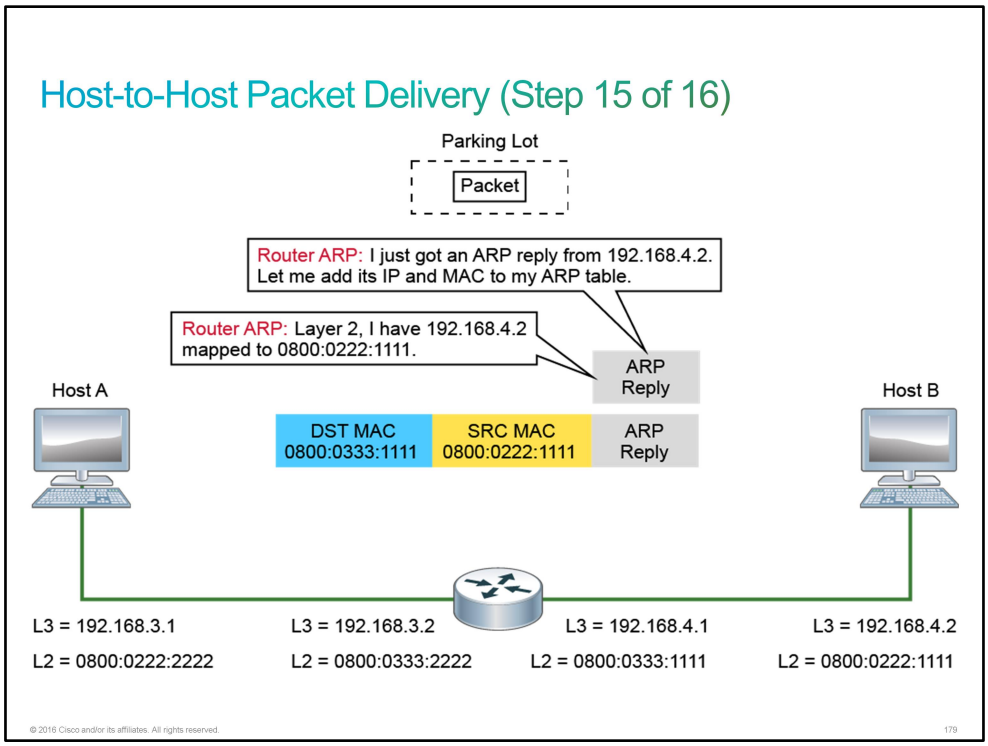
The destination receives and processes the ARP request.

Host-to-Host Packet Delivery (Step 14 of 16)



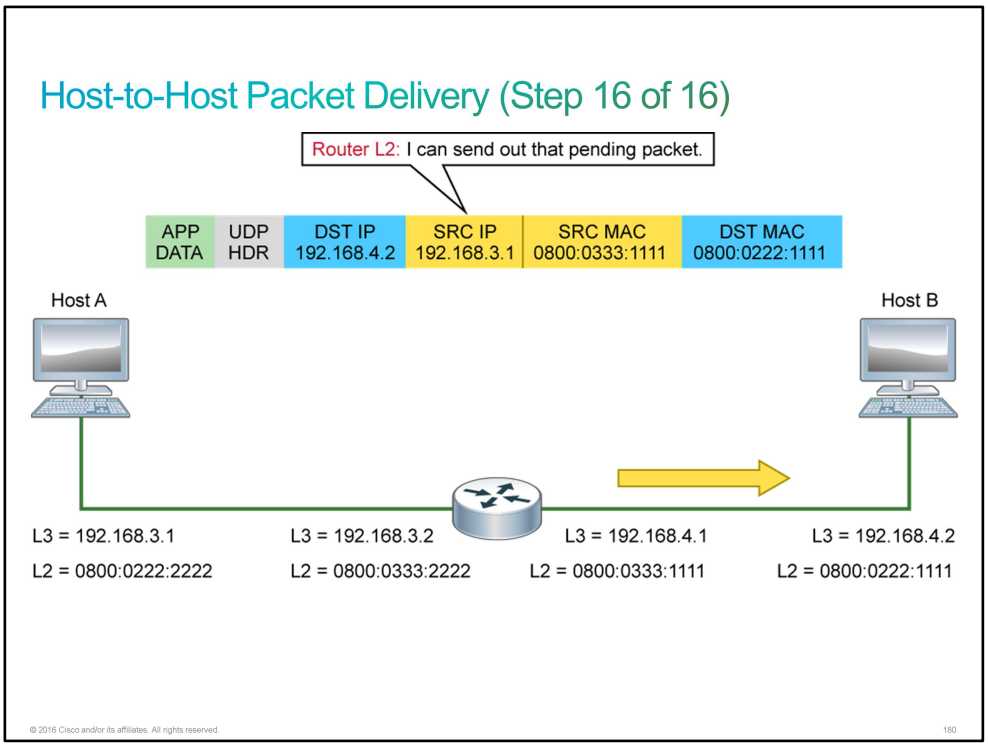
The host receives the frame that contains the ARP request and passes the request to the ARP process. The ARP process takes the information about the router from the ARP request and places the information in its local ARP table. The ARP process generates the ARP reply and sends it back to the router.

Host-to-Host Packet Delivery (Step 15 of 16)



The router receives the ARP reply and takes the information that is required for forwarding the packet to the next hop. The router populates its local ARP table and starts the packet-forwarding process.

Host-to-Host Packet Delivery (Step 16 of 16)



The frame is forwarded to the destination. Note that the router changed the Layer 2 address as needed, but it will not change the Layer 3 address.

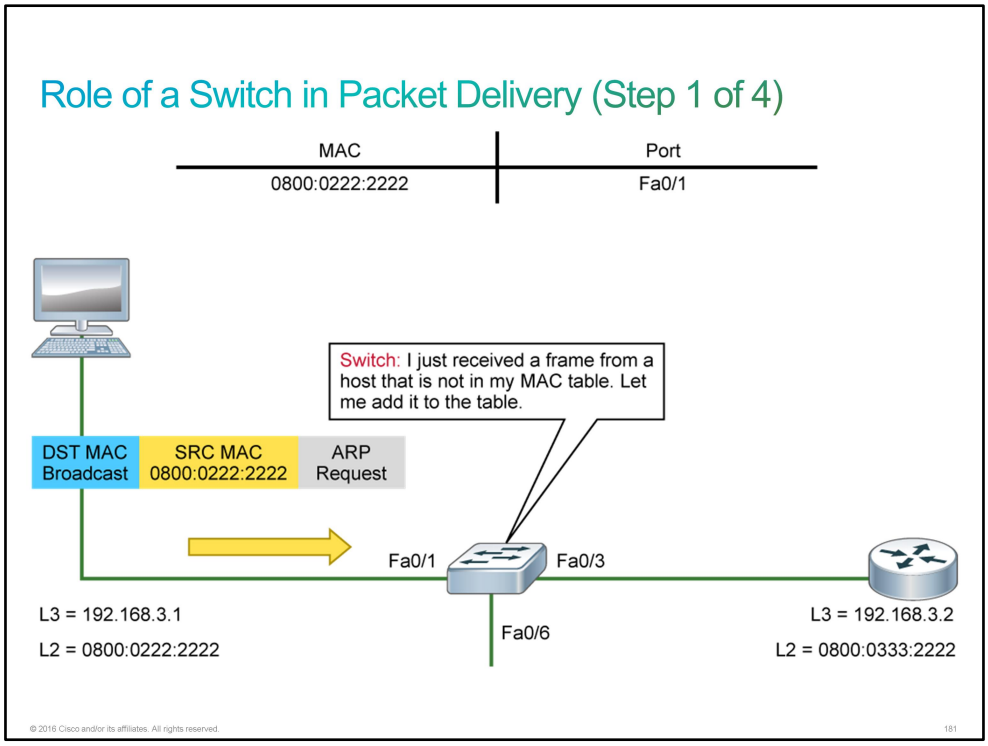
Note	The router changes source and destination MAC addresses, while source and destination IP addresses remain the same.

Note	Remember that a switch does not change the frame in any way. When the switch receives the frame, it needs to forward it out the proper port according to the MAC address table.

Role of a Switch in Packet Delivery

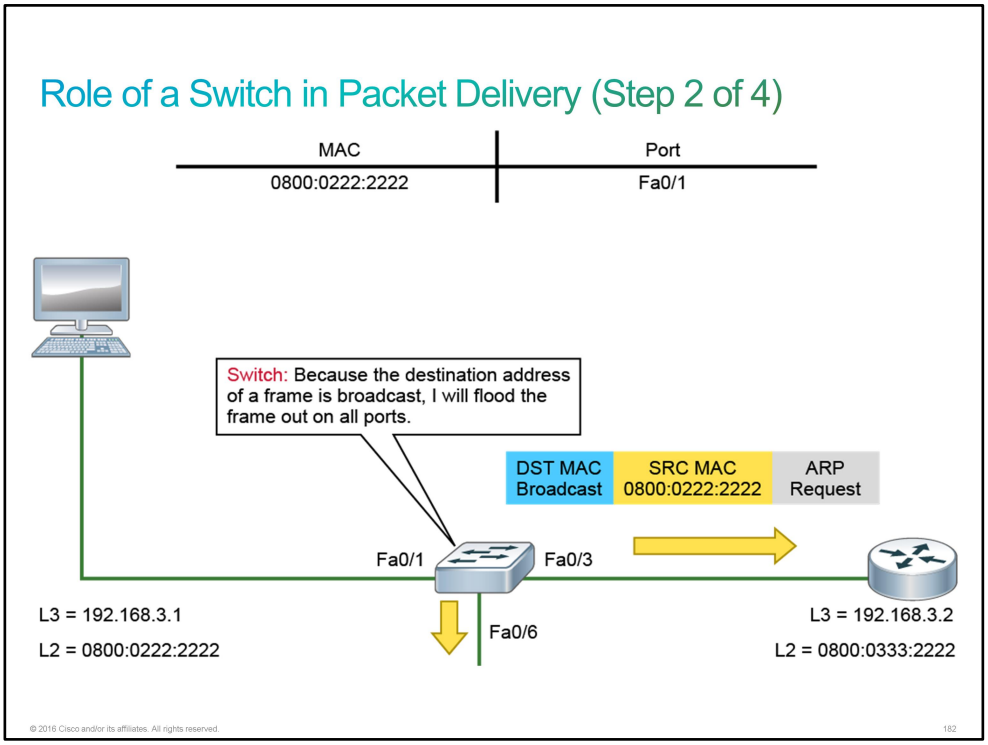
This example focuses on the role of a switch in the host-to-host packet delivery process. An application on host PC A wishes to send data to a distant network. Before an IP packet can be forwarded to the default gateway, its MAC address needs to be obtained. ARP on PC A creates an ARP request and sends it out. Before the ARP request reaches other devices on a network, the switch receives it.

Role of a Switch in Packet Delivery (Step 1 of 4)



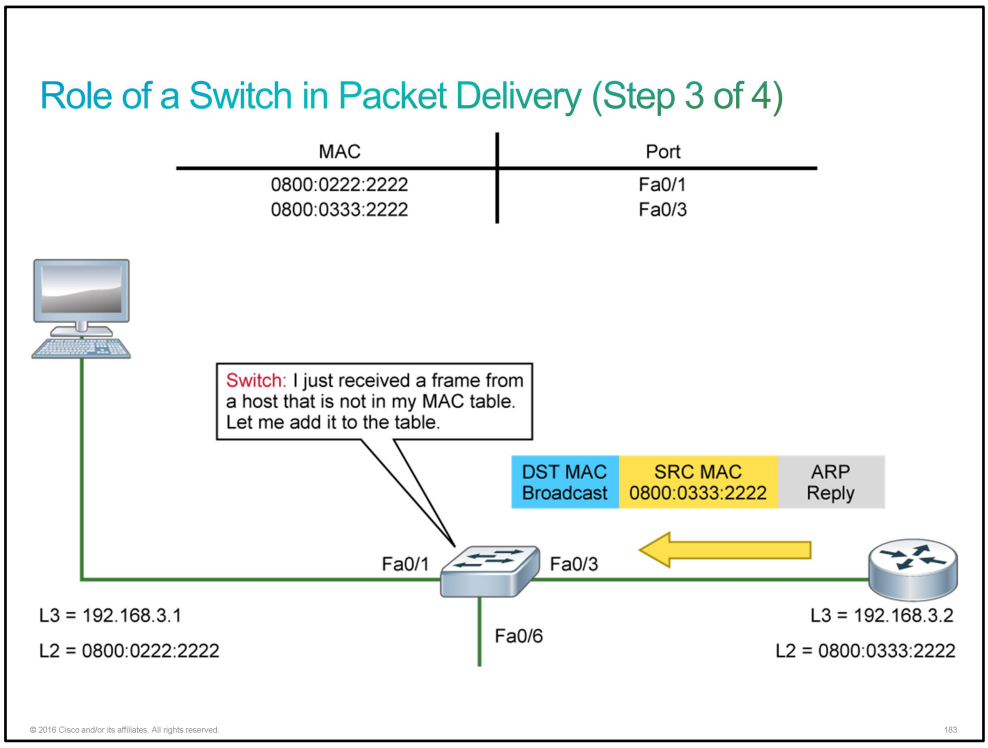
When the switch receives the frame, it needs to forward it out on the proper port. However, in this example, the source and destination MAC addresses are not in the MAC address table of the switch. The switch can learn the port mapping for the source host from the source MAC address in the frame, so the switch will add it to the table (0800:0222:2222 = port FastEthernet0/1).

Role of Switch in Packet Delivery (Step 2 of 4)



Because the destination address of the frame is a broadcast, the switch has to flood the packet out to all the ports. The only exception is the port on which the switch received the broadcast frame.

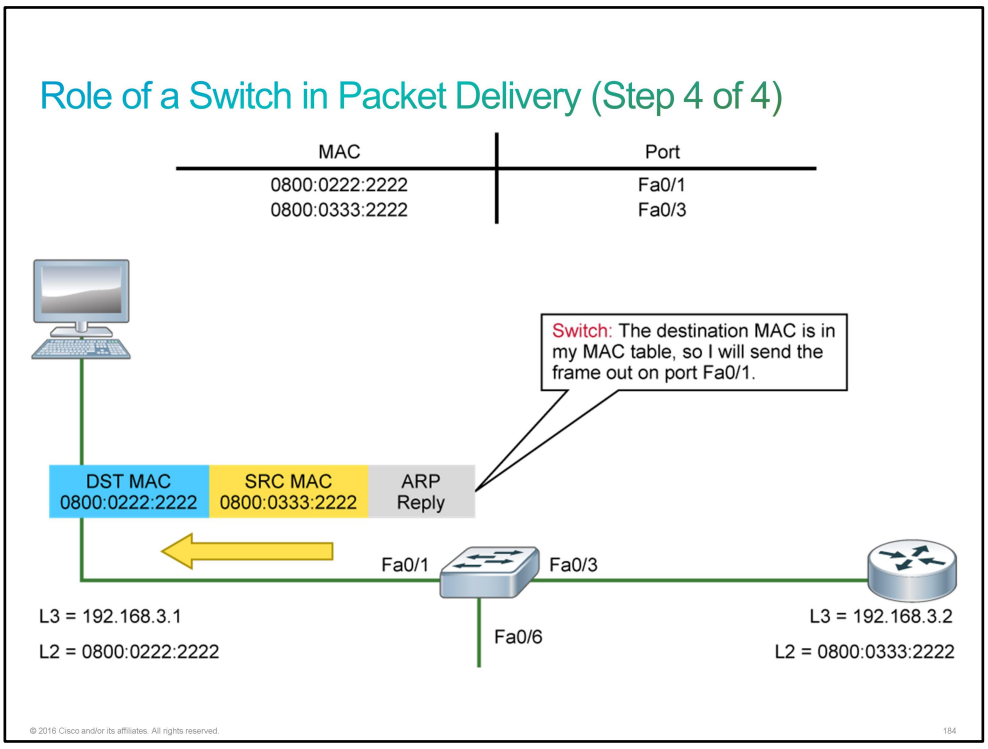
Role of Switch in Packet Delivery (Step 3 of 4)



The router replies to the ARP request and sends an ARP reply packet back to the sender as a unicast frame.

The switch learns the port mapping for the new source host from the source MAC address in the frame. The switch adds it to the MAC address table (0800:0333:2222 = port FastEthernet0/3).

Role of Switch in Packet Delivery (Step 4 of 4)



The destination address of the frame is found in the MAC address table, so the switch can forward out the frame on port FastEthernet0/1. If the destination address is not found in the MAC address table, the switch would need to flood out the frame on all ports.

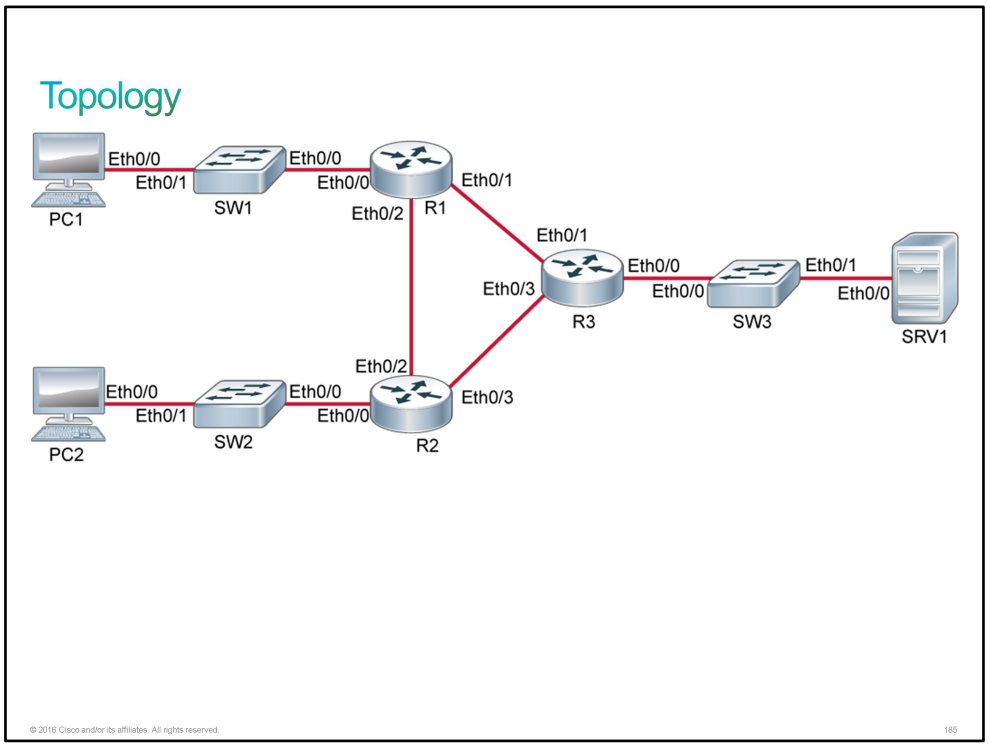
All frames pass through the switch unchanged. When the switch builds the MAC address table, it sends all unicast frames directly to a destination host based on the destination MAC address and data that are stored in the MAC address table.

Discovery 9: Exploration of Packet Forwarding

Introduction

This discovery lab will guide you through the exploration of packet forwarding. The lab is prepared with the devices as represented in the topology diagram. The devices are fully configured, including static routing on the routers.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2
PC2	IP address	10.10.2.20/24

Device	Characteristic	Value
PC2	Default gateway	10.10.2.1
SRV1	Hostname	SRV1
SRV1	IP address	10.10.3.30/24
SRV1	Default gateway	10.10.3.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.2.4/24
SW2	Default gateway	10.10.2.1
SW2	Ethernet0/0 description	Link to R2
SW2	Ethernet0/1 Description	Link to PC2
SW3	Hostname	SW3
SW3	VLAN 1 IP address	10.10.3.4/24
SW3	Default gateway	10.10.3.1
SW3	Ethernet0/0 description	Link to R3
SW3	Ethernet0/1 description	Link to SRV1
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/1 description	Link to R3
R1	Ethernet0/1 IP address	10.1.1.2/30
R1	Ethernet0/2 description	Link to R2

Device	Characteristic	Value
R1	Ethernet0/2 IP address	10.1.1.10/30
R2	Hostname	R2
R2	Ethernet0/0 description	Link to SW2
R2	Ethernet0/0 IP address	10.10.2.1/24
R2	Ethernet0/2 description	Link to R1
R2	Ethernet0/2 IP address	10.1.1.9/30
R2	Ethernet0/3 description	Link to R3
R2	Ethernet0/3 IP address	10.1.1.6/30
R3	Hostname	R3
R3	Ethernet0/0 description	Link to SW3
R3	Ethernet0/0 IP address	10.10.3.1/24
R3	Ethernet0/1 description	Link to R1
R3	Ethernet0/1 IP address	10.1.1.1/30
R3	Ethernet0/3 description	Link to R2
R3	Ethernet0/3 IP address	10.1.1.5/30

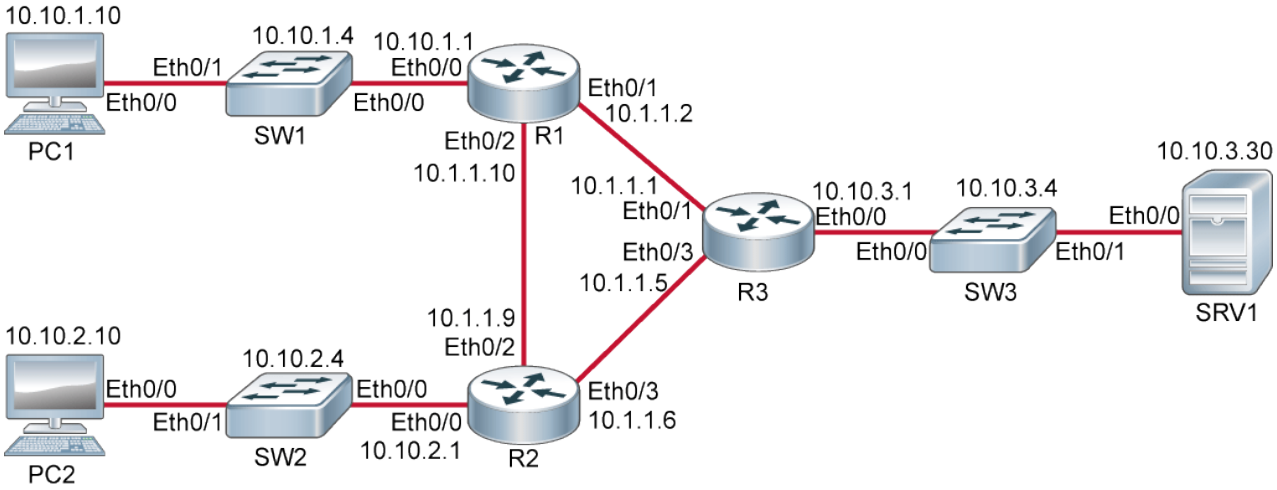
PCs and SRV in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Exploration of Packet Forwarding

Activity

Step 1 Consult the topology diagram. This discovery will focus on the forwarding of packets from PC1 to SRV1.

The devices in the path between these two hosts are SW1, R1, R3 and SW3.



Step 2 Access the console of PC1 and verify connectivity to SRV1 using the **ping** and **traceroute** commands.

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
PC1# traceroute 10.10.3.30
Type escape sequence to abort.
Tracing the route to 10.10.3.30
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 0 msec 0 msec
  2 10.1.1.1 1 msec 0 msec 1 msec
  3 10.10.3.30 0 msec *  1 msec
```

The **traceroute** output shows 10.10.1.1 and 10.1.1.1 being in the forwarding path to SRV1. These addresses belong to Ethernet0/0 on R1 and Ethernet0/1 on R3. The interfaces Ethernet0/1 on R1 and Ethernet0/0 on R3 are also involved in the forwarding process, as are the switches SW1 and SW3.

Step 3 One at a time, access the consoles of PC1, R1, R3, and SRV1 and use the **show interfaces** command to inventory the IP addresses and MAC addresses on the interfaces that are involved in the forwarding process.

The information that you need is in the output of the **show interfaces** command, but to focus explicitly on the data that you are interested in it would be useful to send the output through the include filter and only display lines that contain the string **address**.


```
PC1# show interfaces Ethernet0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.1500 (bia aabb.cc00.1500)
  Internet address is 10.10.1.10/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:07, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1470 packets input, 93664 bytes, 0 no buffer
     Received 1229 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     311 packets output, 34770 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     2 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

This example illustrated the full command syntax of the **show interfaces** command. The examples that follow will utilize command abbreviation.

The string that is passed to the include filter cannot be abbreviated, per se. That is, Cisco IOS cannot determine that when you use the string **add** that you intend for it to be an abbreviation of **address**. But the only appearance of the string **add** in the command's output is as a substring of **address**, therefore it is an acceptable string to use for this purpose.

```
R1# sh int e0/0 | inc add
  Hardware is AmdP2, address is aabb.cc00.1200 (bia aabb.cc00.1200)
  Internet address is 10.10.1.1/24
R1# sh int e0/1 | inc add
  Hardware is AmdP2, address is aabb.cc00.1210 (bia aabb.cc00.1210)
  Internet address is 10.1.1.2/30
```

The main purpose of MAC addresses is to differentiate between hosts on a multiple access network. With point to point serial links, there are exactly two systems on the network. Packets that one sends the other one receives, and vice versa. Because of this reciprocity, there is no need for MAC addresses to differentiate hosts on this type of network.

```
R3# sh int e0/0 | inc add
  Hardware is AmdP2, address is aabb.cc00.1400 (bia aabb.cc00.1400)
  Internet address is 10.10.3.1/24
R3# sh int e0/1 | inc add
  Hardware is AmdP2, address is aabb.cc00.1410 (bia aabb.cc00.1410)
  Internet address is 10.1.1.1/30


SRV1# sh int e0/0 | inc add
  Hardware is AmdP2, address is aabb.cc00.2e00 (bia aabb.cc00.2e00)
  Internet address is 10.10.3.30/24
```

MAC addresses in your output may be different.

Step 4 The output of the **show interfaces** commands can be compiled for reference into a table.

The table would appear as follows:

Device	Interface	MAC Address	IP Address
PC1	Ethernet0/0	aabb.cc00.1500	10.10.1.10
R1	Ethernet0/0	aabb.cc00.1200	10.10.1.1
R1	Ethernet0/1	aabb.cc00.1210	10.10.1.2
R3	Ethernet0/1	aabb.cc00.1410	10.10.1.10
R3	Ethernet0/0	aabb.cc00.1400	10.10.3.1
SRV1	Ethernet0/0	aabb.cc00.2e00	10.10.3.30

Step 5 When PC1 generates an IP packet for SRV1, it will encapsulate the data with an IP header specifying 10.10.3.30 as the destination IP address and 10.10.1.10 as the source IP address. It will then encapsulate the IP packet with an Ethernet header specifying aabb.cc00.1200 (the Ethernet0/0 MAC address in R1) as the destination MAC address and aabb.cc00.1500 (its own MAC address) as the source. PC1 obtains the MAC address of R1 from its ARP cache. Access the console of PC1 and display its ARP cache.

The entry for 10.10.1.1 was populated in the ARP table when you performed the **ping** operation at the beginning of this discovery.

```
PC1# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1             45   aabb.cc00.1200  ARPA   Ethernet0/0
Internet  10.10.1.10             -   aabb.cc00.1500  ARPA   Ethernet0/0
```

Step 6 Execute the following sequence of commands to observe the behavior of the ARP process. You will enable debugging of ARP packets and you will use **show** commands to provide visibility into the process. You will shut down the Ethernet0/0 interface of PC1, which will clear the ARP cache entries that are associated with the interface. You will then re-enable the interface and initiate connectivity, both actions stimulating ARP activity. The informational notes that are imbedded in the directions further explain the operations.

On PC1, enable debugging of ARP packets:

```
PC1# debug arp
ARP packet debugging is on
```

Be very careful when using **debug** commands in production environments. Depending on the circumstances, they can have a catastrophic effect on router performance. Until you have experience with **debug** commands, it is best to consult a senior engineer within your organization on their use.


```
PC1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)# interface Ethernet 0/0
PC1(config-if)# do show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1              51   aabb.cc00.1200  ARPA   Ethernet0/0
Internet  10.10.1.10              -   aabb.cc00.1500  ARPA   Ethernet0/0
```

The **do** command allows access to EXEC mode commands from within the configuration mode. The **show ip arp** command verifies that the two entries are still in the ARP cache.

```
PC1(config-if)# shutdown
PC1(config-if)#
*Oct  9 12:40:03.589: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to
administratively down
*Oct  9 12:40:04.589: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/0, changed state to down
PC1(config-if)# do show ip arp
```

This time there is no output from the **show ip arp** command. The ARP cache on R1 is currently empty. The entries that are associated with Ethernet0/0 were cleared when the interface was shut down.

```
PC1(config-if)# no shutdown
PC1(config-if)#
*Oct  9 12:41:24.437: IP ARP: sent rep src 10.10.1.10 aabb.cc00.1500,
                 dst 10.10.1.10 ffff.ffff.ffff Ethernet0/0
*Oct  9 12:41:24.437: IP ARP: sent rep src 10.10.1.10 aabb.cc00.1500,
                 dst 10.10.1.10 ffff.ffff.ffff Ethernet0/0
```

The two messages above are debug messages. They both indicate that R1 sent ARP packets. The destination IP address is 10.10.1.10. R1 is sending this ARP broadcast asking any hosts that have the IP address 10.10.1.10 to respond back with an ARP reply. Cisco IOS sends this ARP broadcast automatically when interfaces are brought online. It is an attempt to recognize when there are duplicate IP addresses on the network. If any responses are received, syslog messages would be generated to alert the network administrator about the situation. No replies were received, which is normal.

```
*Oct  9 12:41:26.434: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to
up
*Oct  9 12:41:27.434: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/0, changed state to up
```

These two messages are the normal syslog messages, which are generated when interfaces change their state.

```
PC1(config-if)# end
PC1#
```

You just left configuration mode. The rest of this exploration will be completed from privileged EXEC.

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

For this ping operation from PC1 to SRV1 to complete, PC1 must send packets to R1 for forwarding. PC1 needs to know the MAC address of R1 to send the packets to R1.

```
*Oct  9 12:41:27.434: IP ARP: creating incomplete entry for IP address:
10.10.1.1 interface Ethernet0/0
```

This debug message indicates that PC1 recognizes that it needs the MAC address for 10.10.1.1 (R1, its default gateway). PC1 creates an entry in its ARP cache and starts the ARP process.

```
*Oct  9 12:41:27.434: IP ARP: sent req src 10.10.1.10 aabb.cc00.1500,
                 dst 10.10.1.1 0000.0000.0000 Ethernet0/0
```

This debug message indicates that PC1 sent an ARP request specifying 10.10.1.1 as the destination. This request is broadcast to all hosts within the broadcast domain. PC1 is requesting any system with the IP address 10.10.1.1 to respond with an ARP reply.

```
*Oct  9 12:41:27.435: IP ARP: rcvd rep src 10.10.1.1 aabb.cc00.1200, dst
10.10.1.10 Ethernet0/0
```

This debug message indicates that PC1 received an ARP reply from 10.10.1.1, indicating that its MAC address is aabb.cc00.1200.

Step 7 View the ARP cache on PC1 resulting from the exchange of ARP packets that you just witnessed.

The ARP cache of PC1 now has an entry mapping the IP address of R1 to the MAC address of R1.

```
PC1# show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.10.1.1               8   aabb.cc00.1200  ARPA   Ethernet0/0
Internet  10.10.1.10              -   aabb.cc00.1500  ARPA   Ethernet0/0
```

Step 8 The close inspection of the ARP process is complete. Turn off the debug operations.

Debug can be turned off on a per classification basis. That is, you could have used **undebug arp** to turn off the debug process that you started with the **debug arp** command.

```
PC1# undebug all
All possible debugging has been turned off
```

A common abbreviation that is used for **undebug all** is **u all**.

The lab environment does not support capturing packets on the interface links, but the results above support the following extrapolation which describes how a packet is forwarded from PC1 to SRV1:

- The IP header remains constant across the entire path, the IP header will specify 10.10.3.30 as the destination IP address and 10.10.1.10 as the source IP address.
- A unique Layer 2 header is used to traverse each network segment.
- PC1 and R1 learn each other's MAC addresses via ARP.
- R1 and R3 learn each other's MAC addresses via ARP.
- SRV1 and R3 learn each other's MAC addresses via ARP.
- PC1 will encapsulate the IP packet with an Ethernet header that specifies aabb.cc00.1200 (R1 Ethernet0/0) as the destination MAC address and aabb.cc00.1500 (PC1) as the source MAC address.
- PC1 will send this packet out its Ethernet0/0 interface, and R1 will receive it on its Ethernet0/0 interface.
- R1 will strip the Ethernet header and replace it with another Ethernet header that specifies aabb.cc00.1410 (R3 Ethernet 0/1) as the destination MAC address and aabb.cc00.1210 (R1 Ethernet0/1) as the source MAC address.
- R3 will strip the Ethernet header and replace it with another Ethernet header that specifies aabb.cc00.2e00 (SRV1) as the destination MAC address and aabb.cc00.1400 as the source MAC address.
- R3 will send this packet out its Ethernet0/0 interface, and SRV1 will receive it on its Ethernet0/0 interface.

Step 9 The previous steps did not depict how SW1 supports the forwarding of packets between PC1 and R1 and how SW3 supports the forwarding of packets between R3 and SRV1. Switches learn which ports connect to which MAC addresses based on examination of the source MAC address on incoming frames. When they know which ports lead to which MAC address, they can forward to those MAC addresses based on the destination MAC address in frames. Access the console of SW1 and view its MAC address table.

The MAC address of PC1 is associated with the port Ethernet0/1 and the MAC address of R1 is associated with the port Ethernet0/0.

```
SW1# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.1200    DYNAMIC     Et0/0
   1    aabb.cc00.1500    DYNAMIC     Et0/1
Total Mac Addresses for this criterion: 2
```

Step 10 Clear the MAC address table on SW1 and display it again to verify that it is empty.

The example shows an empty MAC address table, but when you attempt this step, you may see that the entries have already repopulated. If so, simply repeat the last two commands as quickly as possible (use the **Up Arrow** key for command recall) until you see the empty MAC address table.

```
SW1# clear mac address-table dynamic
SW1# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
```

Step 11 Wait at least 10 seconds after viewing the empty MAC address table before continuing. Display the MAC address table one more time.

The MAC address table has again been repopulated with the MAC addresses of PC1 and R1.

```
SW1# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    aabb.cc00.1200    DYNAMIC     Et0/0
   1    aabb.cc00.1500    DYNAMIC     Et0/1
Total Mac Addresses for this criterion: 2
```

Step 12 How did the table get repopulated? When the switch receives a packet of any kind, it examines the source MAC address to determine if it needs to add it to the MAC address table. By default, with Cisco IOS, Ethernet interfaces send packets to their own MAC address every 10 seconds as a keepalive mechanism. Verify this setting by accessing the console of PC1 and use the **show interface** command to view the status of Ethernet0/0.

The keepalive value is set to 10 seconds. Also make note of the number of packets output from the interface.


```
PC1# sh int e0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.1500 (bia aabb.cc00.1500)
  Internet address is 10.10.1.10/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     3583 packets input, 225244 bytes, 0 no buffer
     Received 3014 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     708 packets output, 76818 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     2 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Step 13 Wait 10 seconds and repeat the **show interface** command. Verify that the number of packets output has increased by at least 1.

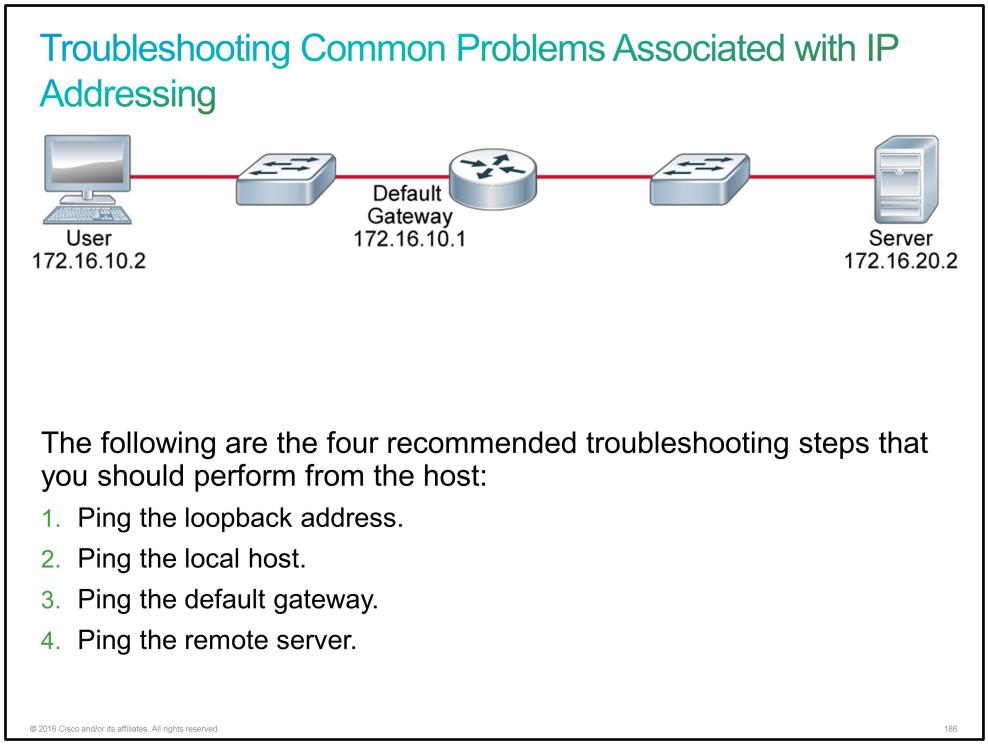
You now have some experience with the forwarding of packets between IP hosts, including the ARP process and the use of MAC addresses on Ethernet networks. You also investigated in how switches populate and use the MAC address tables. Feel free to continue exploring independently within the lab environment.

```
PC1# sh int e0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.1500 (bia aabb.cc00.1500)
  Internet address is 10.10.1.10/24
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     3624 packets input, 227780 bytes, 0 no buffer
     Received 3048 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     716 packets output, 77591 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     2 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

This is the end of the discovery lab.

Troubleshooting Common Problems Associated with IP Addressing

Troubleshooting IP addressing is an important skill and will prove valuable when resolving several network issues. Assume that a host cannot communicate to a server that is on a remote network.



1. Ping the Loopback Address

Access the command prompt and ping 127.0.0.1. This address is the diagnostic or loopback address. If you get a successful ping, your IP stack is considered to be initialized. If it fails, you have an IP stack failure and you need to reinstall TCP/IP on the host.

```
C:\> ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. Ping the Local Host

From the command prompt, ping the IP address of the local host. If the ping is successful, your NIC is functioning. If it fails, there is a problem with the NIC. If the ping is successful, it does not mean that a cable is plugged into the NIC, but only that the IP protocol stack on the host can communicate to the NIC (via the LAN driver).

```
C:\> ping 172.16.10.2
Pinging 172.16.10.2 with 32 bytes of data:
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Reply from 172.16.10.2: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3. Ping the Default Gateway

From the command prompt, ping the default gateway (router). If the ping works, it means that the NIC is plugged into the network and can communicate on the local network. If it fails, you have a local physical network problem that could be anywhere from the NIC to the router.

```
C:\> ping 172.16.10.1
Pinging 172.16.10.1 with 32 bytes of data:
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Reply from 172.16.10.1: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

4. Ping the Remote Server

If Steps 1 through 3 are successful, try to ping the remote server. If the ping works, then you know that you have IP communication between the local host and the remote server. You also know that the remote physical network is working.

```
C:\> ping 172.16.20.2
Pinging 172.16.20.2 with 32 bytes of data:
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Reply from 172.16.20.2: bytes=32 time<1ms TTL=128
Ping statistics for 172.16.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

If you still cannot communicate with the server after Steps 1 through 4 are successful, you probably have some type of name resolution problem and need to check your DNS settings. But if the ping to the remote server fails, then you know that you have some type of remote physical network problem and need to go to the server and work on Steps 1 through 3 until you find the issue.

Important Troubleshooting Commands

As a network engineer, your primary goal is to make sure that your network equipment is always operating properly. This section will cover the most important commands that you will find helpful and perhaps even mandatory throughout your networking career and specifically during network troubleshooting situations.

The commands, which are truly invaluable, as are follows:

- **ping—ping** uses the IP ICMP echo request and echo reply messages to test reachability to a remote system. In its simplest form, **ping** simply confirms that an IP packet is capable of getting to and getting back from a destination IP address. This tool generally returns two pieces of information: whether the source can reach the destination (and, by inference, vice versa) and the RTT (typically in milliseconds). If **ping** fails or returns an unusual RTT, you can use the **traceroute** command to help narrow down the problem. You can also vary the size of the ICMP echo payload to test problems that are related to the MTU.
- **traceroute**—**traceroute** can return useful information about TCP/IP connectivity across your network. The **traceroute** utility sends out either ICMP echo request (Windows) or UDP (most implementations) messages with gradually increasing IP TTL values to probe the path by which a packet traverses the network. The first packet with the TTL set to 1 will be discarded by the first hop. Also, the first hop will send back an ICMP TTL exceeded message that is sourced from its IP address facing the source of the packet. When the machine running the **traceroute** receives the ICMP TTL exceeded message, it can determine the hop via the source IP address. This process continues until the request or message reaches the destination. The destination will return either an ICMP echo reply (Windows) or an ICMP port unreachable, indicating that the request or message has reached the destination. The Cisco implementation of **traceroute** sends out three packets at each TTL value, allowing **traceroute** to report routers that have multiple, equal-cost paths to the destination.
- **tracert—**This is the same command as **traceroute**, but it is a Microsoft Windows command and will not work on a Cisco router.
- **arp -a** —The device uses ARP to perform IP address resolution that is the linking of IP addresses to MAC addresses. ARP uses a broadcast to do this action by asking the host that has the given IP address to respond to the broadcast with its MAC address. The **arp -a** command displays IP-to-MAC-address mappings on a Windows PC.

```
C:\Windows\system32> arp -a
Interface: 10.1.10.100 --- 0xd
  Internet Address      Physical Address      Type
   10.1.10.1             54-75-d0-8e-9a-d8     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

- **show ip arp**—This is the same command as **arp -a,** but it displays the ARP table on a Cisco router. Like the **traceroute** and **tracert** commands, the two are not interchangeable through DOS and Cisco.
- **ipconfig /all—ipconfig**—This is a command-line utility that is available on all versions of Microsoft Windows starting with Windows NT. You run the **ipconfig** utility from the Windows command prompt. This utility allows you to get the IP address information of a Windows computer. The **ipconfig /all** option displays the IP address, network mask, and gateway for all physical and virtual network adapters. Also, it displays DNS and WINS settings for each adapter.

- **telnet**—You will find Telnet or SSH applications useful for connecting to remote devices. One way to obtain information about a remote network device is to connect to it using either the Telnet or SSH applications. Telnet and SSH are virtual terminal protocols that are part of the TCP/IP suite. The protocols allow connections and remote console sessions from one network device to one or more remote devices. To log on to a host that supports Telnet, use the **telnet** EXEC command:

```
RouterA# telnet host
```

(where *host* is an IP address or hostname of a remote system)

- **ssh**—Telnet is the most common method of accessing a network device. However, Telnet is an insecure way of accessing a network. SSH is a secure replacement for Telnet that gives the same type of access. Communication between the client and server is encrypted in both SSHv1 and SSHv2. Implement SSHv2 when possible because it uses a more enhanced security encryption algorithm. To start an encrypted session with a remote networking device, use the **ssh user** EXEC command:

```
RouterA# ssh ip address
```


Challenge

1. Which two statements about ARP are accurate? (Choose two.)

   A. A device can use ARP to assign MAC addresses to dynamic NICs.
   B. ARP entries are not permanent.
   C. Devices can exchange entire ARP tables.
   D. ARP tables are created and maintained dynamically.

2. Which two features does ARP provide? (Choose two.)

   A. mapping IP addresses to DLCIs on a network
   B. mapping IP addresses to MAC addresses on a network
   C. locally storing DLCIs that are learned via ARP
   D. mapping MAC addresses to DLCIs on a network
   E. locally storing MAC addresses that are learned via ARP

3. Which Cisco IOS command do you use to display the ARP table?

   A. **show arp table**
   B. **show -a**
   C. **show ip arp**
   D. **arp -a**

4. Which Microsoft Windows command do you use to display the ARP table?

   A. **show arp table**
   B. **show arp**
   C. **show ip arp**
   D. **arp -a**

5. Which of the following troubleshooting steps would you take if you wanted to check the connectivity to the local network?

   A. Ping the loopback address.
   B. Ping the local host.
   C. Ping the default gateway.
   D. Ping the remote server.

6. Which of the following troubleshooting steps would you take if you wanted to check the IP Stack on your own device?

   A. Ping the loopback address.
   B. Ping the local host.
   C. Ping the default gateway.
   D. Ping the remote server.

7. Which of the following troubleshooting steps would you take if you wanted to check the NIC on your own device?

   A. Ping the loopback address.
   B. Ping the local host.
   C. Ping the default gateway.
   D. Ping the remote server.

Answer Key

Challenge

1. B, D
2. B, E
3. C
4. D
5. C
6. A
7. B

Lesson 7: Enabling Static Routing

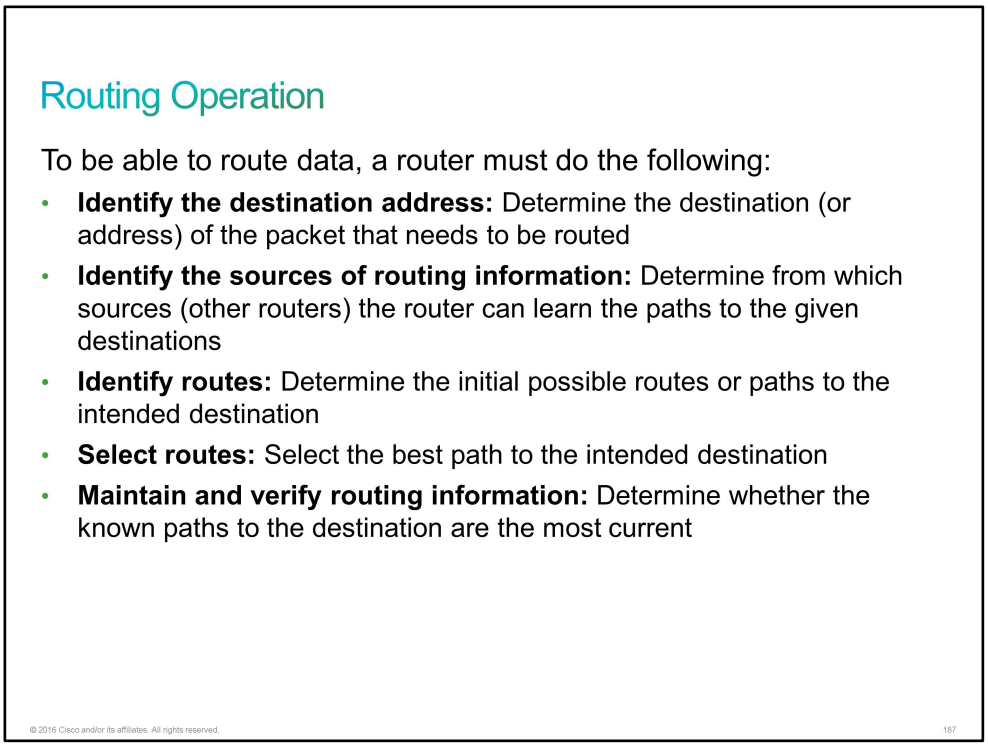
Introduction

Your boss sends you to your customer to enable basic static IP routing. You should be able to present to the customer the difference between static and dynamic routing. If there is a problem with several static routes and a static default route, you should be able to understand what the solution is. You should be able to configure and verify both static and default static routes.

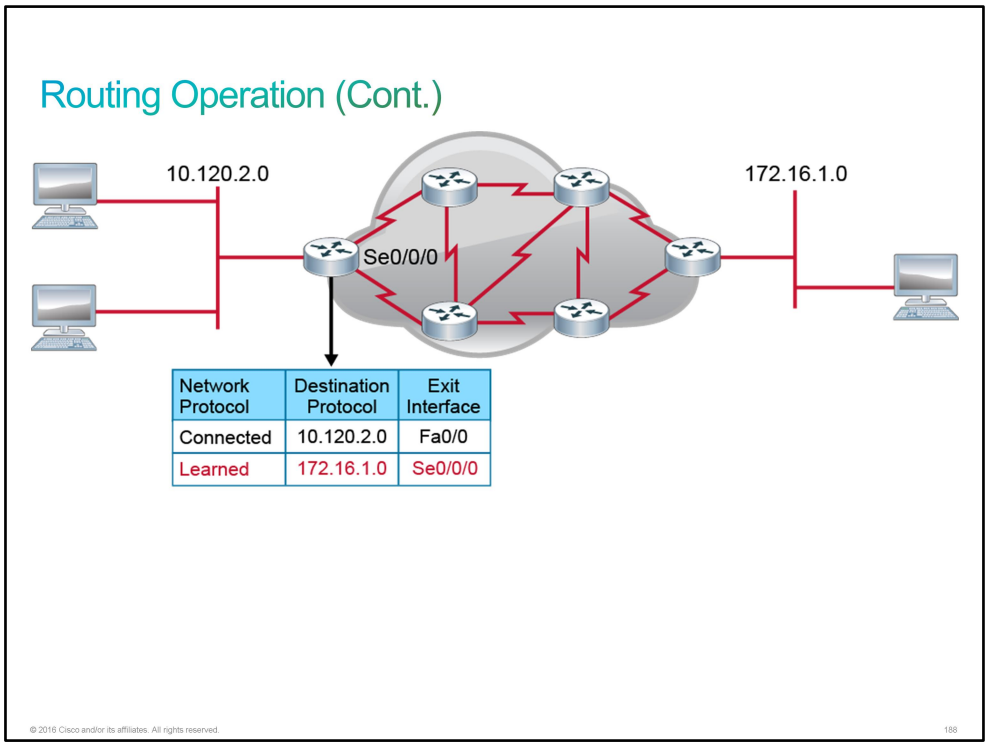
Routing Operation

Routing is the process of determining where to send data packets that are destined for addresses outside the local network. Routers gather and maintain routing information to enable the transmission and receipt of such data packets.

Conceptually, routing information takes the form of entries in a routing table, with one entry for each identified route. You can manually configure the entries in the routing table, or the router can use a routing protocol to create and maintain the routing table dynamically to accommodate network changes when they occur.



The routing information that a router obtains from other routers is placed in its routing table. The router relies on this table to tell it which interfaces to use when forwarding packets. The following figure shows that the router on the left uses interface Serial0/0/0 to get to the 172.16.1.0 subnet.



If the destination network is directly connected, that is, if there is an interface on the router that belongs to that network, the router already knows which interface to use when forwarding packets. If destination networks are not directly attached, the router must learn the best route to use when forwarding packets.

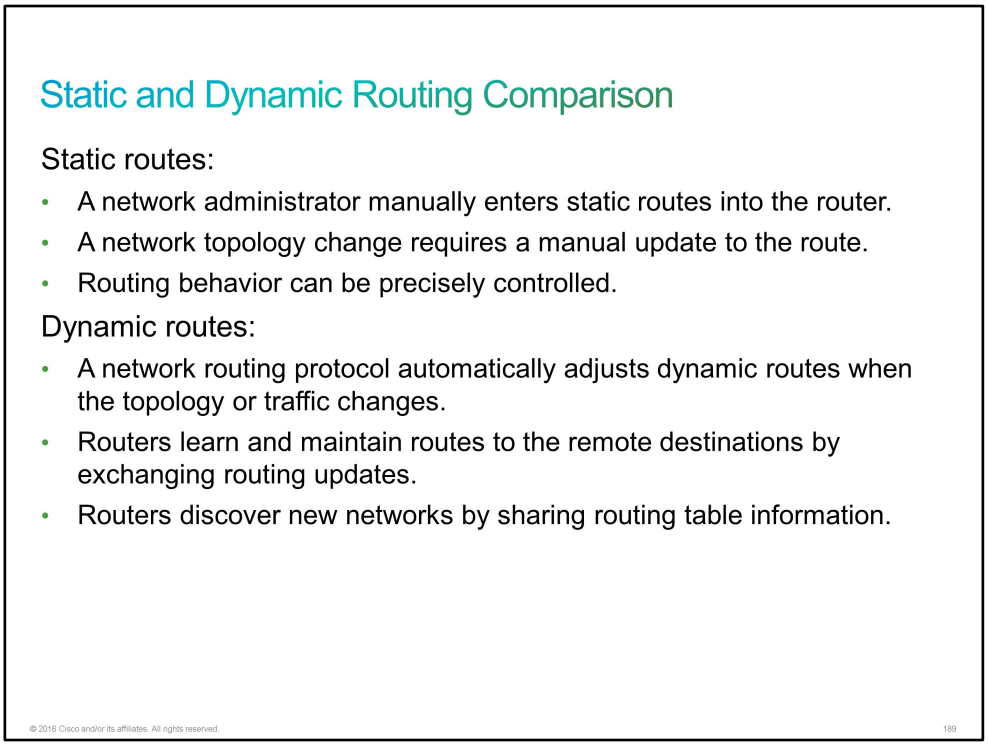
The destination information can be learned in two ways:

- You can enter routing information manually, also known as a static route.
- You can collect routing information through the dynamic routing process that runs in the routers.

Static and Dynamic Routing Comparison

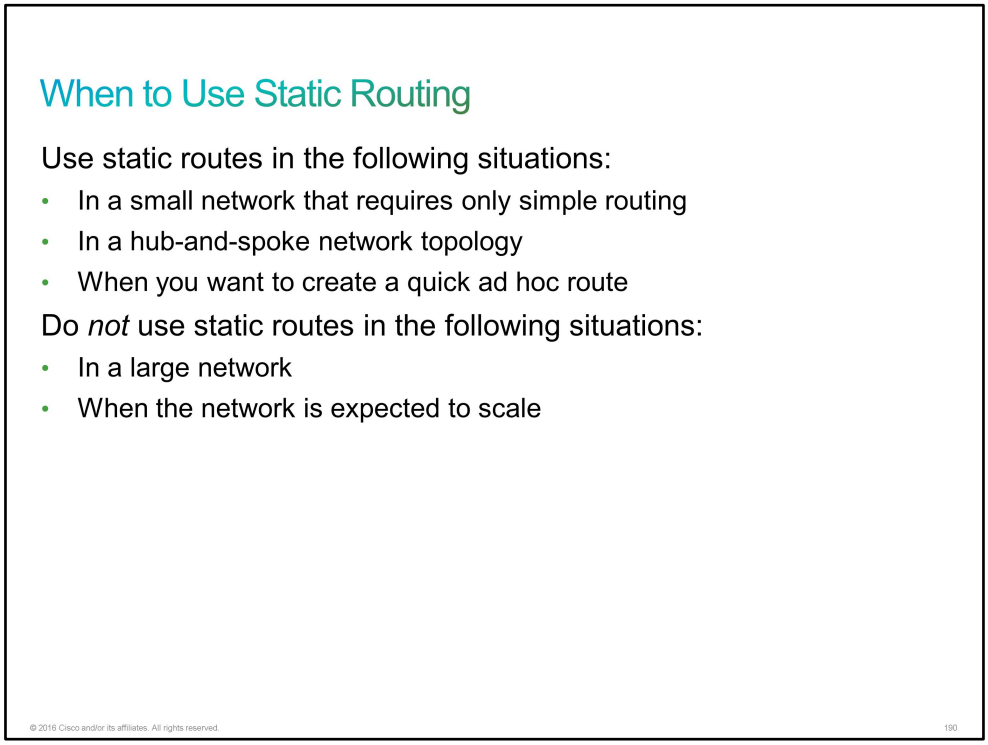
There are two ways that a router can learn where to forward packets to destination networks that are not directly connected.

- **Static routing:** The router learns routes when an administrator manually configures the static route. The administrator must manually update this static route entry whenever an internetwork topology change requires an update. Static routes are user-defined routes that specify the path that packets take when moving between a source and a destination. These administrator-defined routes allow very precise control over the routing behavior of the IP internetwork.
- **Dynamic routing:** The router dynamically learns routes after an administrator configures a routing protocol that helps determine routes. Unlike the situation with static routes, after the network administrator enables dynamic routing, the routing process automatically updates route knowledge whenever the device receives new topology information. The router learns and maintains routes to the remote destinations by exchanging routing updates with other routers in the internetwork.



When to Use Static Routing

Static routes are best suited for small networks, such as LANs, where routes rarely change. If routes change, you need to manually update your routes to reflect the new data transmission paths.



Some of the advantages of using static routes are as follows:

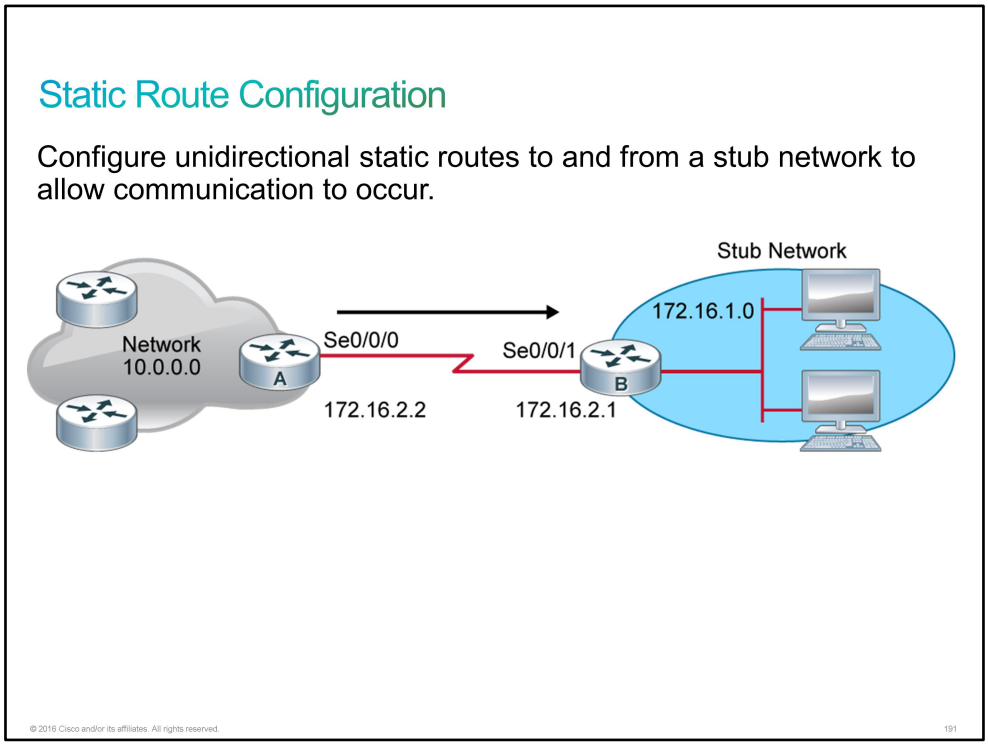
- **Conserving router resources**: Static routing does not consume network bandwidth and the CPU resources of the router. When you use a routing protocol, the traffic between routers adds some overhead as the routers exchange routing updates about remote networks. Depending on the size of the network, a router requires some CPU cycles to compute the best way to remote networks.
- **Simple to configure in a small network**: Static routes are commonly used in small networks that have few routers. Many small networks are designed as stub networks; for these types of networks, static routes are the most appropriate solution. Also, most of these networks are designed in a hub-and-spoke topology, where you can use default routes for branches that are pointing to the hub, which is the gateway to other networks.
- **Security**: Sometimes, you may want to define static routes to control the data transmission paths that are used by your data. This option may be useful in highly secure environments.

Some of the disadvantages of using static routes are as follows:

- **Scalability**: Static routing might be appropriate for networks that have fewer than four or five routers. Dynamic routing is more appropriate for large networks to reduce the probability of errors in routing configuration.
- **Accuracy**: If your network changes and you do not update the static routes, your router does not have accurate knowledge of your network. Not having accurate knowledge of your network can result in lost or delayed data transmissions.
- **High maintenance**: When the number of routers increases, the number of static routes also increases. In large networks, adding even one router with only one new network means that in addition to configuring the newly added router with static routes to other networks, you must configure all existing routers in the network with static routes to the new network.

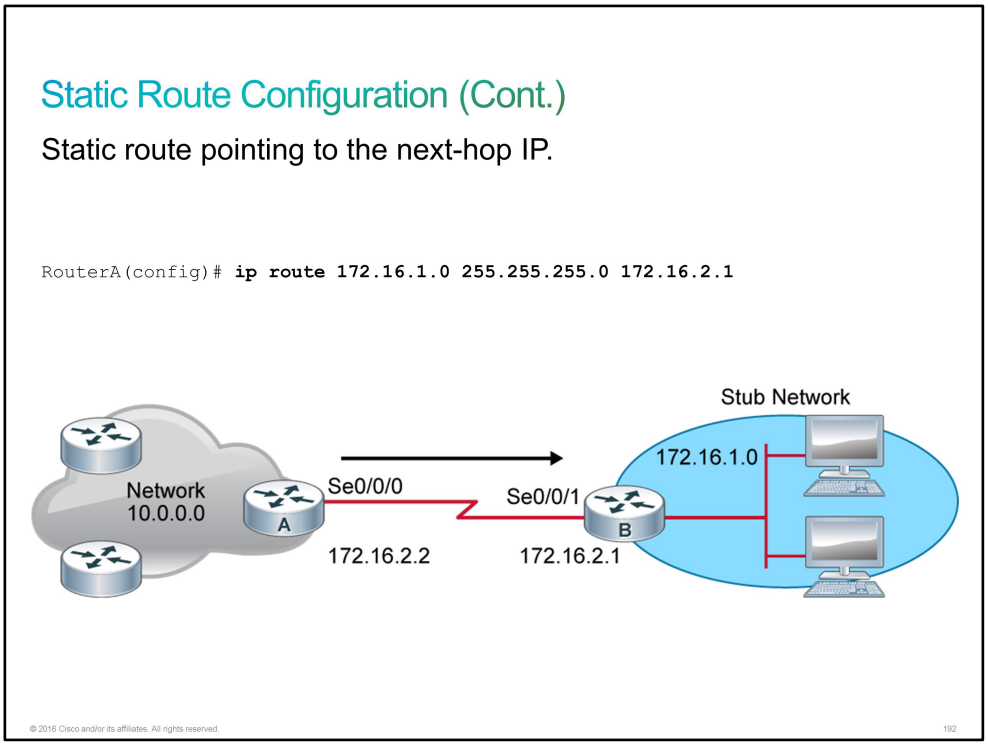
Static Route Configuration

Static routes are commonly used when you are routing from a network to a stub network (a network that is accessed by a single link). Static routes can also be useful for specifying a "gateway of last resort" to which all packets with an unknown destination address are sent.



Static route configuration steps:

- Define a path to an IP destination network (172.16.1.0 255.255.255.0).
- Use the IP address of the next-hop router (172.16.2.1).
- Or, use the outbound interface of the local router (Serial0/0/0).



In the figure, router A is configured with a static route to reach the 172.16.1.0 subnet via the next hop IP 172.16.2.1 using the **ip route** command.

Alternatively, you can configure the static route by pointing to the exit interface instead of using the next-hop IP address.

```
RouterA(config)# ip route 172.16.1.0 255.255.255.0 serial0/0/0
```

The table lists the **ip route** command parameters for this example.

Command Parameters	Description
ip route	Identifies the static route
172.16.1.0	IP address of a static route to the destination subnetwork
255.255.255.0	Indicates the subnet mask—there are 8 bits of subnetting in effect
172.16.2.1	IP address of the next-hop router in the path to the destination
Serial 0/0/0	Identifies the interface that will be used to reach the next-hop router

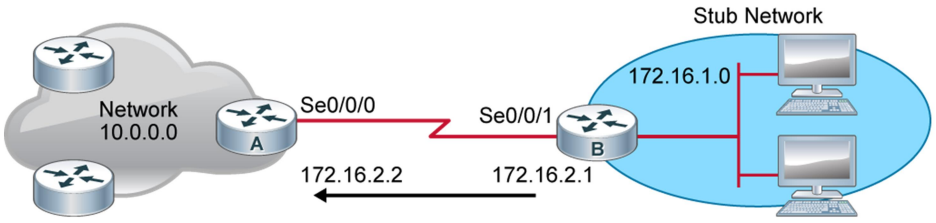
In the figure, you would also need to configure router B with a static or default route to reach the networks behind router A via the serial interface of router B.

Default Routes

Use a default route when the route from a source to a destination is not known or when it is not feasible for the router to maintain many routes in its routing table.

Default Routes

This route allows the stub network to reach all known networks beyond router A.



193

Default Routes (Cont.)

Default route pointing to the next-hop IP.

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

Default route pointing to the exit interface.

```
RouterB(config)# ip route 0.0.0.0 0.0.0.0 Serial0/0/1
```


194

A default static route is a route that matches all packets. Default static routes are used in these instances:

- When no other routes in the routing table match the destination IP address of the packet, or when a more specific match does not exist. A common use for a default static route is to connect the edge router of a company to an ISP network.
- When a router has only one other router to which it is connected. This condition is known as a stub router.

The syntax for a default static route is like the one that is demonstrated for any other static route, except that the network address is 0.0.0.0 and the subnet mask is 0.0.0.0.

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

or

```
RouterB(config)# ip route 0.0.0.0 0.0.0.0 serial0/0/1
```

The 0.0.0.0 network address and 0.0.0.0 mask is called a *quad-zero* route.

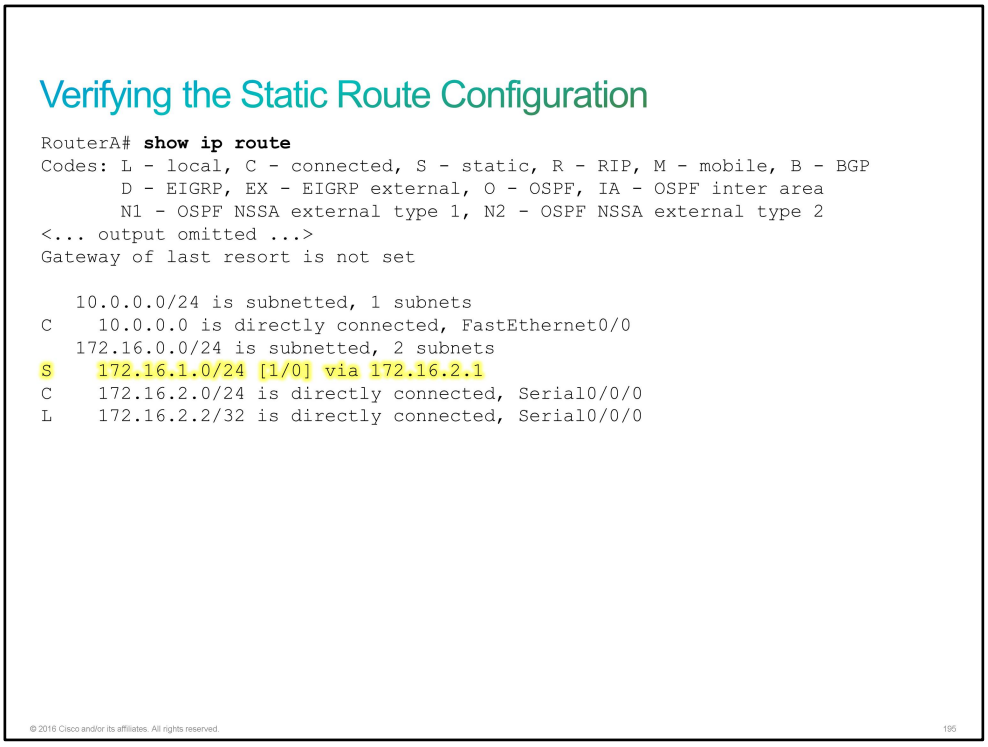
In the figure, router B is configured to forward to router A all packets that do not have the destination network that is listed in the router B routing table.

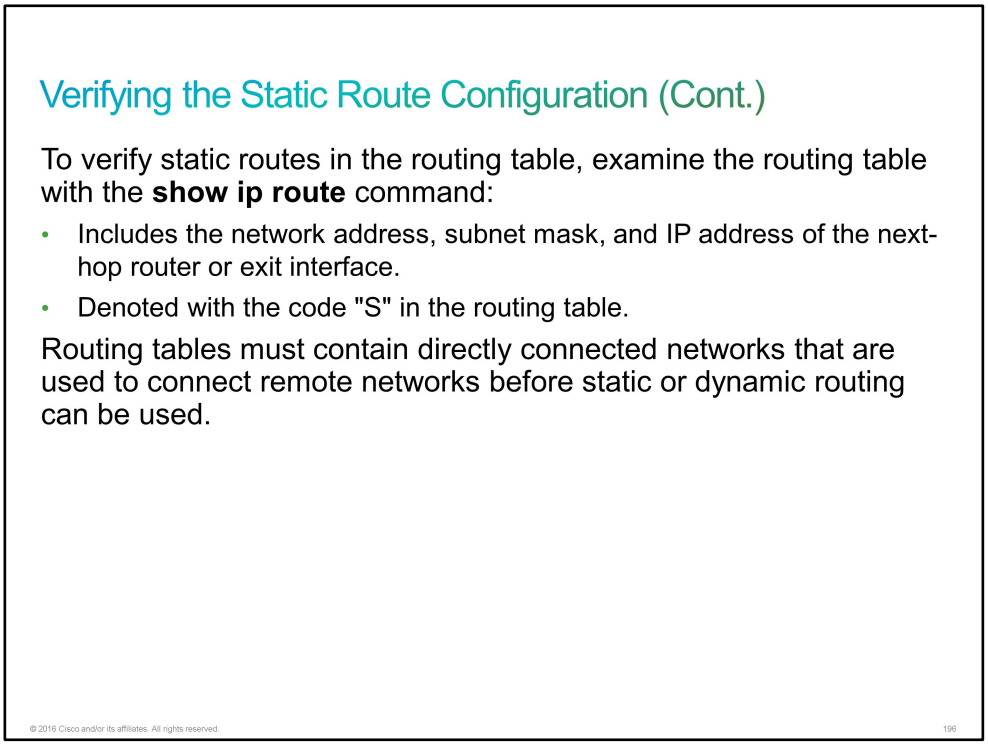
This table lists the **ip route** command parameters for this example.

Command Parameters	Description
ip route	Identifies the static route
0.0.0.0	Routes to networks that are not in the routing table
0.0.0.0	Special mask that indicates the default route
172.16.2.2	IP address of the next-hop router to be used as the default for packet forwarding

Verifying the Static Route Configuration

Most routing tables contain a combination of static routes and dynamic routes. However, the routing table must first contain the directly connected networks that are used to access the remote networks before any static or dynamic routing can be used.





A static route includes the network address and subnet mask of the remote network, along with the IP address of the next-hop router or exit interface. Static routes are denoted with the code "S" in the routing table, as shown in the figure.

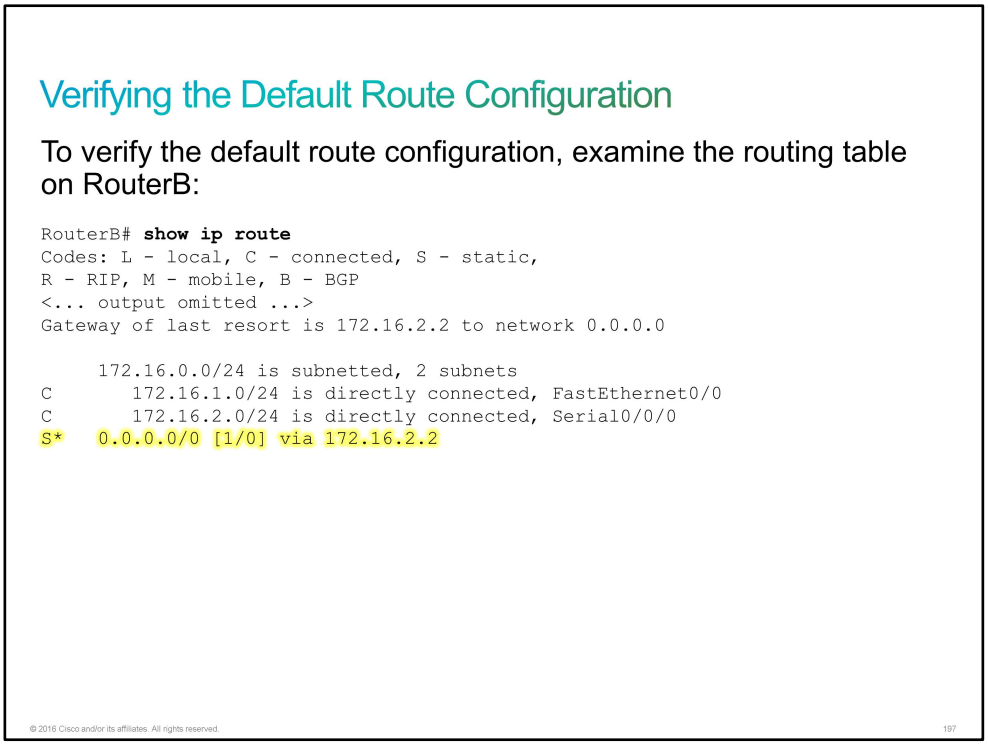
When you configure a static route to use an exit interface instead of a next-hop IP address, the routing table entry is changed as follows:

```
RouterB# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 172.16.2.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.2.2
      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.16.1.0/24 is directly connected, Ethernet0/0
L        172.16.1.1/32 is directly connected, Ethernet0/0
C        172.16.2.0/24 is directly connected, Serial2/0
L        172.16.2.1/32 is directly connected, Serial2/0
RouterB#
```

Note that the entry in the routing table no longer refers to the next-hop IP address but refers directly to the exit interface. This exit interface is the same one to which the static route was resolved when it used the next-hop IP address. Now that the routing table process has a match for a packet and this static route, it is able to resolve the route to an exit interface in a single lookup.

Note	The static route displays the route as directly connected. It is important to understand that this does not mean that this route is a directly connected network or a directly connected route. This route is still a static route.

Verifying the Default Route Configuration



The example in the figure shows the RouterB routing table after configuration of the default route.

The asterisk (*) indicates the last path option that the router will use when forwarding a packet.

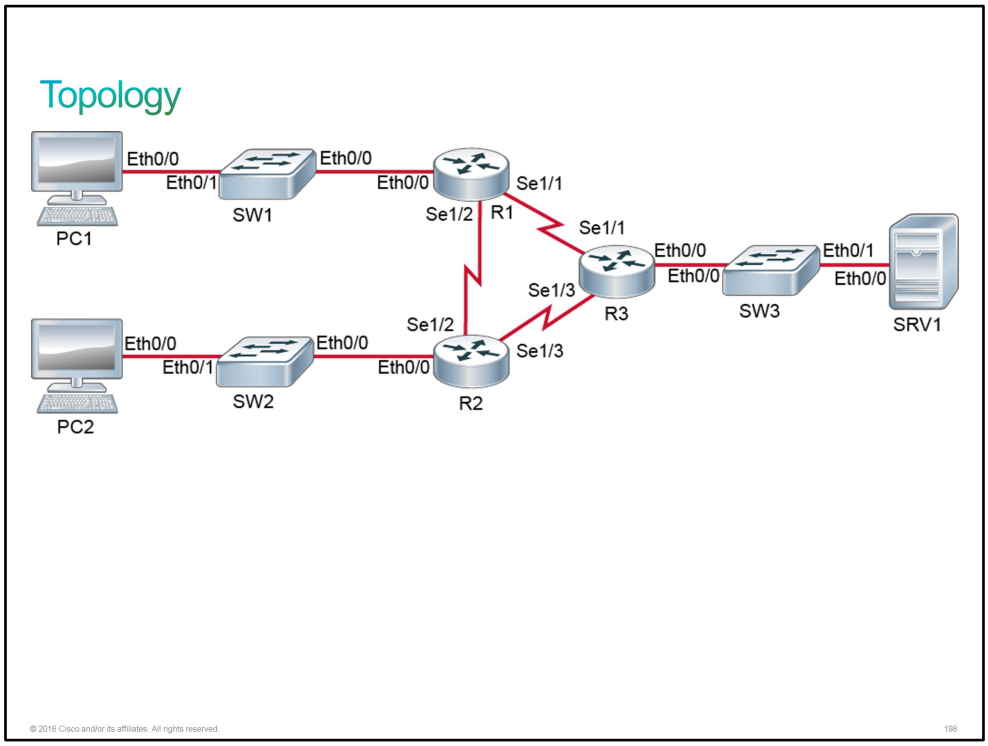
Discovery 10: Configure and Verify Static Routes

Introduction

In this discovery lab, you will explore IP routing, focusing on static routing. You will configure and verify static routes and observe the packet forwarding behavior that is associated with various routing configurations, including the use of a statically defined default route.

The lab is prepared with the devices as represented in the topology diagram and connectivity table. All devices have their basic configurations in place including hostnames and IP addresses. Default gateways are defined on PC1, PC2, and SRV1, but no other routing has been configured.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1

Device	Characteristic	Value
PC2	Hostname	PC2
PC2	IP address	10.10.2.20/24
PC2	Default gateway	10.10.2.1
SRV1	Hostname	SRV1
SRV1	IP address	10.10.3.30/24
SRV1	Default gateway	10.10.3.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.2.4/24
SW2	Default gateway	10.10.2.1
SW2	Ethernet0/0 description	Link to R2
SW2	Ethernet0/1 description	Link to PC2
SW3	Hostname	SW3
SW3	VLAN 1 IP address	10.10.3.4/24
SW3	Default gateway	10.10.3.1
SW3	Ethernet0/0 description	Link to R3
SW3	Ethernet0/1 description	Link to SRV1
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Serial1/1 description	Link to R3

Device	Characteristic	Value
R1	Serial1/1 IP address	10.1.1.2/30
R1	Serial1/2 description	Link to R2
R1	Serial1/2 IP address	10.1.1.10/30
R2	Hostname	R2
R2	E0/0 description	Link to SW2
R2	E0/0 IP address	10.10.2.1/24
R2	S1/2 description	Link to R1
R2	S1/2 IP address	10.1.1.9/30
R2	S1/3 description	Link to R3
R2	S1/3 IP address	10.1.1.6/30
R3	Hostname	R3
R3	Ethernet0/0 description	Link to SW3
R3	Ethernet0/0 IP address	10.10.3.1/24
R3	Serial1/1 description	Link to R1
R3	Serial1/1 IP address	10.1.1.1/30
R3	Serial1/3 description	Link to R2
R3	Serial1/3 IP address	10.1.1.5/30

PCs and SRV in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Verify Devices Reachability

Activity

Step 1 Before getting into the configuration of static routes, observe the connectivity when routing is not yet configured on any of the routers.

Access the console of PC1 and ping SW1 and R1.

```
PC1# ping 10.10.1.4
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.4, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
PC1# ping 10.10.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/2 ms
```

Be sure to take advantage of the Cisco IOS command recall feature when entering similar commands. Use the **Up Arrow** key to scroll through the command history and use the **Right** and **Left Arrow** and **Backspace** keys to edit commands that are similar to what you need to enter.

Consult the topology diagram whenever it is helpful to clarify the physical layout of the lab.

You probably expected to be able to ping these addresses. They are on the same subnet as PC1. So, routing is not required. The ARP protocol resolves the MAC address of the peer and communication ensues at Layer 2.

Step 2 Personal computers, and IP end hosts in general, normally have routing tables. They usually consist of a single entry, a default route to their default gateway. View the routing table on PC1 to verify that R1 is its default gateway.

On PC1, enter following command:

```
PC1# show ip route
Default gateway is 10.10.1.1

Host               Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

Step 3 From PC1, ping the IP addresses of the remote Serial1/1 and Serial1/2 interfaces of R1.

On PC1, enter following commands:

```
PC1# ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/201/1002 ms
PC1# ping 10.1.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

The ping worked without any routes configured on R1. PC1 was configured to send all remote traffic to R1, and R1 has all the respective subnets (including the subnet of PC1) in its routing table as directly connected networks.

Step 4 Try to ping R2's Serial1/2 interface, which is a point-to-point neighbor to the Serial1/2 interface of R1.

On PC1, enter following command:


```
PC1# ping 10.1.1.9
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.9, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

This ping attempt fails. Interestingly, the ICMP echo requests actually do make it to R2. PC1 is configured to use R1 as its default gateway and R1 has 10.1.1.0/30 as a directly connected network in its routing table. So, the forwarding to R2 will function. The problem is that R2 does not have a route back to 10.10.1.0/24 and as a result cannot forward the replies to R1. R2 drops the ICMP packet.

Task 2: Configure and Verify Static Routes

Activity

Step 1 Now it is time to configure some static routes. On the R1 router, configure routes to 10.10.2.0/24 and 10.10.3.0/24 through R2 and R3 respectively.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip route 10.10.2.0 255.255.255.0 10.1.1.9
R1(config)# ip route 10.10.3.0 255.255.255.0 10.1.1.1
R1(config)# end
R1#
```

Step 2 On R2, configure routes to 10.10.1.0/24 and 10.10.3.0/24 through R1 and R3 respectively.

On R2, enter the following commands:

```
R2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# ip route 10.10.1.0 255.255.255.0 10.1.1.10
R2(config)# ip route 10.10.3.0 255.255.255.0 10.1.1.5
R2(config)# end
R2#
```

Step 3 Consult the topology diagram and consider the static routes that you just configured. Should PC1 be able to ping PC2? How about SRV1? And how about 10.1.1.6 or 10.1.1.5 (the IP addresses on the subnet between R2 and R3)? Access the console of PC1 and explore the current connectivity.

On PC1, enter the following commands:

```
PC1# ping 10.10.2.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.2.20, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

PC1 can ping PC2. This fact implies bidirectional connectivity. The forwarding ICMP echoes from PC1 to PC2 was successful and the forwarding of ICMP echo replies from PC2 to PC1 was successful.

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

This ping attempt was not successful. With the current route configuration, the ICMP echoes will actually reach SRV1. R1 has a route to 10.10.3.0/24 using R3, and R3 has an interface that is directly connected to 10.10.3.0/24. But, the ICMP echo replies that SRV1 generated will be sent to R3 (the default gateway of SRV1), but R3 does not have a route back to 10.10.1.0/24. So, it will drop the echo replies.

The "." characters in the ping output indicate timeouts on the reply

```
PC1# ping 10.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.6, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

This ping attempt fails for a different reason. The subnet 10.1.1.4/30 is the point to point link between R2 and R3. R1 does not have a route to that subnet. Hence, it must drop the packets that are destined for that subnet.

The "U" characters in the ping output indicate that a router in the forwarding path returned ICMP Unreachable messages to PC1.

Step 4 Consult the topology diagram. There are six subnets. Each router has direct connectivity to three of those subnets with the remaining three subnets being remote to the router. For full connectivity, each router must have a route defined for each of the three remote subnets. Configure the third static route on both R1 and R2, and configure all three routes on R3.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip route 10.1.1.4 255.255.255.252 10.1.1.9
R1(config)# end
R1#
```

On R1 and R2, if your login session has not timed out, command recall will still function when you enter configuration mode, providing access to the previously entered route commands. But, be careful! The routes have different subnet masks, so you must change them along with the IP addresses!

On R2, enter the following commands:

```
R2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# ip route 10.1.1.0 255.255.255.252 10.1.1.10
R2(config)# end
R2#
```


The next hop specified (10.1.1.10) is an interface on R1. 10.1.1.5 on R3 would have been an equivalent option for the next hop. The choice to use R1 as the next hop was arbitrary.

On R3, enter the following commands:

```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# ip route 10.10.1.0 255.255.255.0 10.1.1.2
R3(config)# ip route 10.10.2.0 255.255.255.0 10.1.1.6
R3(config)# ip route 10.1.1.8 255.255.255.252 10.1.1.2
R3(config)# end
R3#
```

Step 5 Now is a good time to verify the routing tables on all three routers.

This example shows the routing table and configuration on R1. R2 and R3 should have similar, complimentary routes configured.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
C        10.1.1.0/30 is directly connected, Serial1/1
L        10.1.1.2/32 is directly connected, Serial1/1
S        10.1.1.4/30 [1/0] via 10.1.1.9
C        10.1.1.8/30 is directly connected, Serial1/2
L        10.1.1.10/32 is directly connected, Serial1/2
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
S        10.10.2.0/24 [1/0] via 10.1.1.9
S        10.10.3.0/24 [1/0] via 10.1.1.1
R1# show running-config | include route
ip route 10.1.1.4 255.255.255.252 10.1.1.9
ip route 10.10.2.0 255.255.255.0 10.1.1.9
ip route 10.10.3.0 255.255.255.0 10.1.1.1
```

Step 6 At this point, all three routers have routes (either directly connected or statically defined) to all six subnets. Full connectivity should be available.

Access the console of PC1 and verify that IP addresses from the different subnets are reachable with the **ping** command.

```
PC1# ping 10.10.2.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.2.20, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
PC1# ping 10.1.1.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
PC1# ping 10.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

Task 3: Demonstrate Static Route Drawback

Activity

Step 1 The pings demonstrate that there is connectivity. Use the **traceroute** command to verify the paths that are in place.

On PC1, enter the following commands:

```
PC1# traceroute 10.10.2.20
Type escape sequence to abort.
Tracing the route to 10.10.2.20
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 1 msec 1 msec
  2 10.1.1.9 1 msec 0 msec 1 msec
  3 10.10.2.20 2 msec *  2 msec
```

The path from PC1 to PC2 goes through R1 and R2.

```
PC1# traceroute 10.10.3.30
Type escape sequence to abort.
Tracing the route to 10.10.3.30
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 1 msec 1 msec
  2 10.1.1.1 1 msec 0 msec 0 msec
  3 10.10.3.30 1 msec *  1 msec
```

It is normal in the lab environment for the middle attempt to the final destination to time out with the **traceroute** command.

The path from PC1 to SRV1 goes through R1 and R3.

At this point, one of the limitations of static routes should be apparent. They do not scale well. In the lab, there are only six subnets and three routers with no path being longer than two hops. In this simple environment, it required nine static routes for full connectivity. As the network complexity grows, the number of required static routes grows very fast and quickly becomes unwieldy.

In the next series of steps, you will experience another limitation of static routes. They do not provide redundancy. Introduce an interface fault into the network.

Step 2 On R3, disable the interface Serial1/1, which connects R3 to R1.

On R3, enter the following commands:

```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface Serial 1/1
R3(config-if)# shutdown
R3(config-if)# end
R3#
*Oct 15 07:04:28.078: %SYS-5-CONFIG_I: Configured from console by console
R3#
*Oct 15 07:04:29.292: %LINK-5-CHANGED: Interface Serial1/1, changed state to
administratively down
*Oct 15 07:04:30.296: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to down
```

Step 3 Shutting down Serial1/1 on R3 will have effects on R1 and R3. Access the console of R1 and verify that a syslog message is displayed, indicating that the interface Serial1/1 has changed status to "down."

In the lab environment, this status change may take a minute to propagate.

```
R1#
*Oct 15 07:04:57.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to down
```

Step 4 View the interface status and routing table on R1.

On R1, enter the following command:

```
R1# show ip interface brief
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES NVRAM  up                    up
Ethernet0/1                unassigned      YES NVRAM  administratively down
down
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                unassigned      YES NVRAM  administratively down
down
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  10.1.1.2        YES NVRAM  up
down
Serial1/2                  10.1.1.10       YES NVRAM  up                    up
Serial1/3                  unassigned      YES NVRAM  administratively down
down
```

The protocol status of Serial1/1 is "down."

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
S        10.1.1.4/30 [1/0] via 10.1.1.9
C        10.1.1.8/30 is directly connected, Serial1/2
L        10.1.1.10/32 is directly connected, Serial1/2
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
S        10.10.2.0/24 [1/0] via 10.1.1.9
```

There are only two static routes in the routing table. With Serial1/1 being down, there is no path to 10.1.1.1 on the 10.1.1.0/30 subnet. So, the route to 10.10.3.0/24 that uses 10.1.1.1 as the next hop is invalid and has been removed from the routing table.

The **static route** command still exists in the configuration.

Step 5 Explore the connectivity from the perspective of PC1. Access the console of PC1 and attempt a **ping** and a **traceroute** to 10.10.3.30.

On PC1, enter the following command:

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

The ping is not successful. The "U" characters indicate that a router in the path (in this case R1) is sending an ICMP unreachable message back to PC1.

```
PC1# traceroute 10.10.3.30
Type escape sequence to abort.
Tracing the route to 10.10.3.30
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 1 msec 0 msec
  2 10.10.1.1 !H  *  !H
```

The path gets to R1 (10.10.1.1), but then gets stuck.

Step 6 Repair the interface fault by returning to R3 and enabling Serial1/1.

On R3, enter the following commands:


```
R3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface Serial 1/1
R3(config-if)# no shutdown
R3(config-if)# end
R3#
*Oct 15 07:13:12.022: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
R3#
*Oct 15 07:13:12.747: %SYS-5-CONFIG_I: Configured from console by console
*Oct 15 07:13:13.027: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
```

Step 7 Return to the console of R1 and verify that the display of the syslog message is indicating that Serial1/1 has changed back to the "up" state.

```
R1#
*Oct 15 07:13:18.148: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
```

Step 8 The real proof comes by verifying end to end connectivity. Return to the console of PC1 and execute a **ping** and a **traceroute** command to SRV1.

On PC1, enter the following commands:

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/11 ms
PC1# traceroute 10.10.3.30
Type escape sequence to abort.
Tracing the route to 10.10.3.30
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 1 msec 1 msec
  2 10.1.1.1 10 msec 10 msec 9 msec
  3 10.10.3.30 11 msec *  9 msec
```

The path from PC1 to SRV1 now, again travels through R1 and R3.

Task 4: Configure and Verify the Backup Static Route

Activity

What has been demonstrated so far in this discovery covers the typical usage of static routes. The next series of steps will show an unconventional use of static routes. This task should be considered an academic exercise. It is only feasible because of the simplicity of the lab environment. Add more routers or subnets into the mix and this methodology would quickly become unwieldy.

Administrative distance is a property that is used to distinguish the trustworthiness of different routing protocols. Cisco IOS routers prefer routes with a lower administrative distance. By default static routes have an administrative distance of 1, which all but guarantees that they will be used in the routing table.

It is optional to specify a different administrative distance on static routes. In this next series of steps, you will define a set of backup static routes with an administrative distance of 2. The only way these routes will end up in the routing table is if one of the routes with an administrative distance of 1 becomes unavailable. You will also verify the behavior of when an interface fails in the new configuration.

Step 1 Access the console of R1 and add three static routes. The new routes will specify the same remote networks as the existing static routes, but they will specify a next hop on the alternate peer router and specify and administrative distance of 2.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip route 10.10.2.0 255.255.255.0 10.1.1.1 2
R1(config)# ip route 10.10.3.0 255.255.255.0 10.1.1.9 2
R1(config)# ip route 10.1.1.4 255.255.255.252 10.1.1.1 2
R1(config)# end
R1#
```

Step 2 Verify that there are now six static routes in the configuration. There are two to each of the remote networks. The second route to each remote network specifies an alternate next hop and an administrative distance of 2.

On R1, enter the following commands:

```
R1# show running-config | include route
ip route 10.1.1.4 255.255.255.252 10.1.1.9
ip route 10.1.1.4 255.255.255.252 10.1.1.1 2
ip route 10.10.2.0 255.255.255.0 10.1.1.9
ip route 10.10.2.0 255.255.255.0 10.1.1.1 2
ip route 10.10.3.0 255.255.255.0 10.1.1.1
ip route 10.10.3.0 255.255.255.0 10.1.1.9 2
```

Step 3 Verify that only three of the static routes appear in the routing table. Only the routes that have the default administrative distance of 1 are selected for the routing table.

On R1, enter the following command:

```
R1# show ip route static
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
S        10.1.1.4/30 [1/0] via 10.1.1.9
S        10.10.2.0/24 [1/0] via 10.1.1.9
S        10.10.3.0/24 [1/0] via 10.1.1.1
```

The values that you see within the brackets are [Administrative Distance / Metric]. These three routes all have an administrative distance of 1.

Step 4 Repeat the respective configuration of static routes on R2 and R3.

On R2, enter the following commands:

```
R2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# ip route 10.10.1.0 255.255.255.0 10.1.1.5 2
R2(config)# ip route 10.10.3.0 255.255.255.0 10.1.1.10 2
R2(config)# ip route 10.1.1.0 255.255.255.252 10.1.1.5 2
R2(config)# end
R2#
```

On R3, enter the following commands:

```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# ip route 10.10.1.0 255.255.255.0 10.1.1.6 2
R3(config)# ip route 10.10.2.0 255.255.255.0 10.1.1.2 2
R3(config)# ip route 10.1.1.8 255.255.255.252 10.1.1.6 2
R3(config)# end
R3#
```

Step 5 Repeat the fault experiment that was performed earlier in the discovery by disabling interface Serial1/1 on R3.

On R3, enter the following commands:

```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface Serial 1/1
R3(config-if)# shutdown
R3(config-if)# end
R3#
*Oct 15 07:29:34.297: %SYS-5-CONFIG_I: Configured from console by console
*Oct 15 07:29:35.080: %LINK-5-CHANGED: Interface Serial1/1, changed state to
administratively down
R3#
*Oct 15 07:29:36.084: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to down
```

Step 6 Access the console of R1 to verify that the display of the syslog message indicating that the Serial1/1 interface of R1 has "changed state to down."

```
R1#
*Oct 15 07:29:58.519: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to down
```

Step 7 View the routing table on R1.

On R1, enter the following command:

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 7 subnets, 3 masks
S        10.1.1.4/30 [1/0] via 10.1.1.9
C        10.1.1.8/30 is directly connected, Serial1/2
L        10.1.1.10/32 is directly connected, Serial1/2
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
S        10.10.2.0/24 [1/0] via 10.1.1.9
S        10.10.3.0/24 [2/0] via 10.1.1.9
```

There is a route to 10.10.3.0/24. The route through R2 (10.1.1.9) with an administrative distance of 2 replaced the route through R3 (10.1.1.1) with an administrative distance of 1 when the connection to the 10.1.1.0/30 network was lost.

Step 8 Access the console of PC1 and verify connectivity between PC1 and SRV1 using the **ping** and **traceroute** commands.

On PC1, enter the following commands:

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 17/17/18 ms
PC1# traceroute 10.10.3.30
Type escape sequence to abort.
Tracing the route to 10.10.3.30
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 0 msec 1 msec
  2 10.1.1.9 9 msec 9 msec 9 msec
  3 10.1.1.5 17 msec 18 msec 17 msec
  4 10.10.3.30 15 msec *  18 msec
```

Connectivity remains even with the loss of the link between R1 and R3. The path is now longer. The path from PC1 to SRV1 traverses R1, R2, and R3.

Step 9 Return to R3 to repair the interface fault.

On R3, enter the following commands:


```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface Serial 1/1
R3(config-if)# no shutdown
R3(config-if)# end
R3#
*Oct 15 07:34:30.570: %SYS-5-CONFIG_I: Configured from console by console
R3#
*Oct 15 07:34:30.968: %LINK-3-UPDOWN: Interface Serial1/1, changed state to up
*Oct 15 07:34:31.972: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
```

Step 10 Access the console of R1 and verify the display of the syslog message indicating that its interface Serial1/1 returns to an "up" state.

```
R1#
*Oct 15 07:34:38.628: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial1/1, changed state to up
```

Step 11 View the routing table on R1.

On R1, enter the following command:

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 9 subnets, 3 masks
C        10.1.1.0/30 is directly connected, Serial1/1
L        10.1.1.2/32 is directly connected, Serial1/1
S        10.1.1.4/30 [1/0] via 10.1.1.9
C        10.1.1.8/30 is directly connected, Serial1/2
L        10.1.1.10/32 is directly connected, Serial1/2
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
S        10.10.2.0/24 [1/0] via 10.1.1.9
S        10.10.3.0/24 [1/0] via 10.1.1.1
```

The original route to 10.10.3.0 using R3 as the next hop and with an administrative distance of 1 has returned to the routing table.

Task 5: Configure and Verify the Default Route

Activity

A default route is a route to the network 0.0.0.0 with the subnet mask 0.0.0.0. Default routes can be defined statically. Default routes are most commonly used when there is a hierarchy in the network. For example, to get from a branch office network to the headquarters network (and the rest of the world), or to get from the corporate network to the Internet (and the rest of the world).

The lab environment is not hierarchical. In fact, it is perfectly symmetrical. So the use of a default route on R1, R2, or R3 is not very practical. But, even so, it can be enlightening to explore the behavior of a default route within the lab environment.

Step 1 Access R1 and remove all the static routes that are configured. Unfortunately, removing those routes is a tedious operation. Be sure to make good use of the Cisco IOS command history feature to ease the burden.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# no ip route 10.1.1.4 255.255.255.252 10.1.1.9
R1(config)# no ip route 10.1.1.4 255.255.255.252 10.1.1.1 2
R1(config)# no ip route 10.10.2.0 255.255.255.0 10.1.1.9
R1(config)# no ip route 10.10.2.0 255.255.255.0 10.1.1.1 2
R1(config)# no ip route 10.10.3.0 255.255.255.0 10.1.1.1
R1(config)# no ip route 10.10.3.0 255.255.255.0 10.1.1.9 2
R1(config)# end
R1#
```

Step 2 Verify that there are no route commands left in the configuration and that only local and connected routes appear in the routing table.

On R1, enter the following commands:

```
R1# show running-config | include route
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C        10.1.1.0/30 is directly connected, Serial1/1
L        10.1.1.2/32 is directly connected, Serial1/1
C        10.1.1.8/30 is directly connected, Serial1/2
L        10.1.1.10/32 is directly connected, Serial1/2
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
```


Step 3 Configure a default route using 10.1.1.1 (Serial1/1 on R3) on R1.

On R1, enter following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1
R1(config)# end
R1#
```

Step 4 Verify that this route is the only route in the running configuration and that there is a default route in the routing table.

On R1, enter the following commands:

```
R1# show running-config | include route
ip route 0.0.0.0 0.0.0.0 10.1.1.1
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 10.1.1.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.1.1.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
C        10.1.1.0/30 is directly connected, Serial1/1
L        10.1.1.2/32 is directly connected, Serial1/1
C        10.1.1.8/30 is directly connected, Serial1/2
L        10.1.1.10/32 is directly connected, Serial1/2
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
```

Step 5 Access the console of PC1 and use the **ping** command to verify connectivity with other IP addresses in the network.

On PC1, enter the following commands:

```
PC1# ping 10.10.3.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.3.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/10 ms
PC1# ping 10.10.2.20
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.2.20, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/13/15 ms
PC1# ping 10.1.1.9
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.9, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/10 ms
```

Replacing the specific routes in the configuration of R1 with a default route to R3 does provide connectivity throughout the network as long as there are no failed interfaces.

Step 6 Examine the path from PC1 to PC2 using the **traceroute** command.

On PC1, enter the following command:

```
PC1# traceroute 10.10.2.20
Type escape sequence to abort.
Tracing the route to 10.10.2.20
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 1 msec 0 msec 1 msec
  2 10.1.1.1 9 msec 5 msec 9 msec
  3 10.1.1.6 13 msec 13 msec 13 msec
  4 10.10.2.20 14 msec *  15 msec
```

R1 is the first hop in the path. R1 no longer has an explicit and efficient static route to 10.10.2.0/24, so it uses its default route and forwards this traffic to R3. R3 has a static route to the 10.10.2.0/24 network via R2. So, the path from PC1 to PC2 goes through R1, R3, and then R2.

Step 7 Examine the path from PC1 to 10.1.1.9 (Serial1/2 on R2) using **traceroute** command.

On PC1, enter the following command:

```
PC1# traceroute 10.1.1.9
Type escape sequence to abort.
Tracing the route to 10.1.1.9
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.1.1 0 msec 0 msec 0 msec
  2 10.1.1.9 11 msec *  9 msec
```

R1 did not need to use its default route to reach 10.1.1.9. R1 has a connected route to 10.1.1.8/30 in its routing table. This specific route is preferred over the default route and is used in this case.

Step 8 You have examined connectivity and the path from PC1 to PC2. Now access the console of PC2 and examine connectivity and the path from PC2 to PC1 using the **ping** and **traceroute** commands.

On PC2, enter the following command:

```
PC2# ping 10.10.1.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/13/14 ms
```

The explicit static routes to 10.10.1.0/24 that are defined on R2 and R3 will sustain connectivity to that subnet within the lab.

On PC2, enter the following command:


```
PC2# traceroute 10.10.1.10
Type escape sequence to abort.
Tracing the route to 10.10.1.10
VRF info: (vrf in name/id, vrf out name/id)
  1 10.10.2.1 1 msec 0 msec 0 msec
  2 10.1.1.10 13 msec 14 msec 13 msec
  3 10.10.1.10 15 msec *  13 msec
```

R2 is the first hop in this path, and R2 has an explicit and optimized static route to 10.10.1.0/24 that uses R1 as the next hop. Hence the path from PC2 to PC1 traverses R2 and then R1. R3 is not involved.

Contrast this situation to the path that was previously displayed for PC1 to PC2. That path required the default route on R1. The path traversed R1, R3, and then R2.

When the path from Host A to Host B is not simply the reverse of the path that is taken from Host B to Host A, it is called asymmetric routing. Asymmetric routing is generally an undesirable behavior.

At this point, you have experimented extensively with static routes. You have configured typical static routes and redundant static routes and a default static route. In each case, you have seen how to verify the status of the configuration and the routing table. You have also examined the packet forwarding behavior in each case including scenarios where there is an interface fault in place. Feel free to continue to independently explore the configuration and function of static routes in the lab environment.

This is the end of the discovery lab.

Challenge

1. Routers obtain information to determine what is entered into its routing table. What sources do the routers use to obtain this information? (Choose two)

   A. Administrators / Engineers
   B. The internet / cloud infrastructure
   C. Other Routers
   D. Servers

2. Static Routes are automatically updated when the network changes. True or False?

   A. True
   B. False

3. When does one use static routes?

   A. In large networks
   B. When the network is expected to scale
   C. In a small network

4. Which one is a disadvantage of Static Routing?

   A. Scalability
   B. Security
   C. Simplicity
   D. Conserving router resources

5. Which of the following approaches is tougher to maintain?

   A. Static Routing
   B. Dynamic Routing

6. When a dynamic routing protocol is used, routers discover networks by sharing routing table information. True or False?

   A. True
   B. False

7. Routers rely on which of the following to make decisions on forwarding packets at layer 3?

   A. MAC Address Table
   B. CDP Table
   C. Routing Table

Answer Key

Challenge

1. A, C
2. B
3. C
4. A
5. A
6. A
7. C

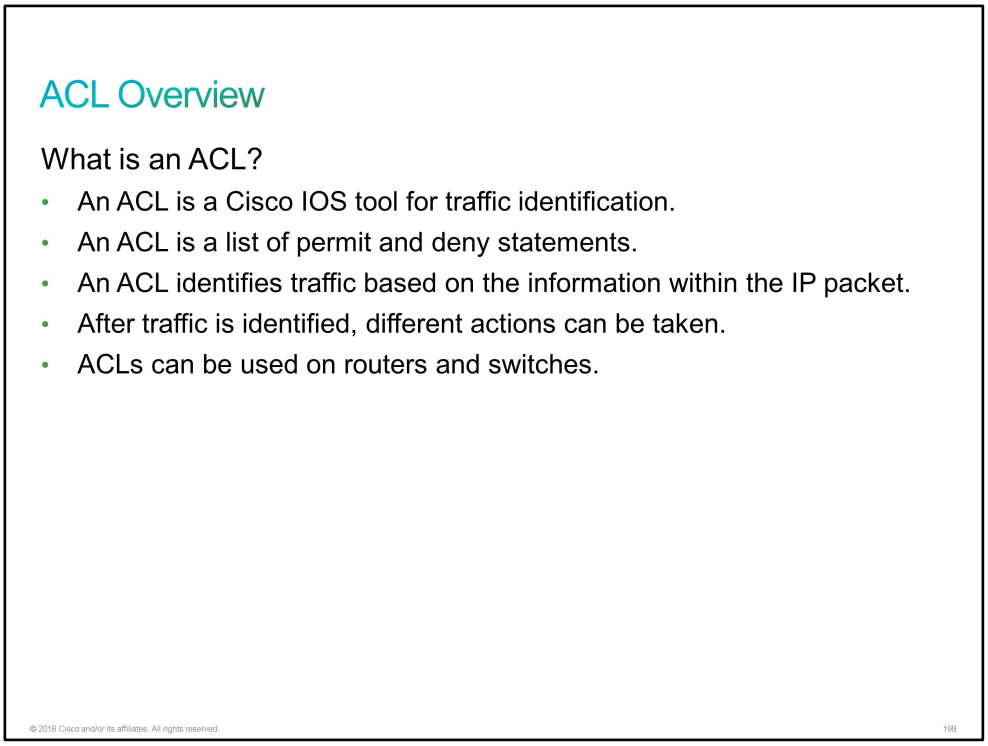
Lesson 8: Learning the Basics of ACL

Introduction

Your boss sends you to your customer to update access control lists. You need to understand the general ACL operation. When updating access control list, you will need to compare wildcard masks with the subnet masks. You will also need to consider different types of ACLs. You will need to configure and verify standard numbered ACLs, which are needed for NAT and vty protection.

ACL Overview

An ACL is a Cisco IOS feature that is used for traffic identification. The ACL enables an administrator to create a set of rules in the form of permit and deny statements that describe which traffic should be identified.

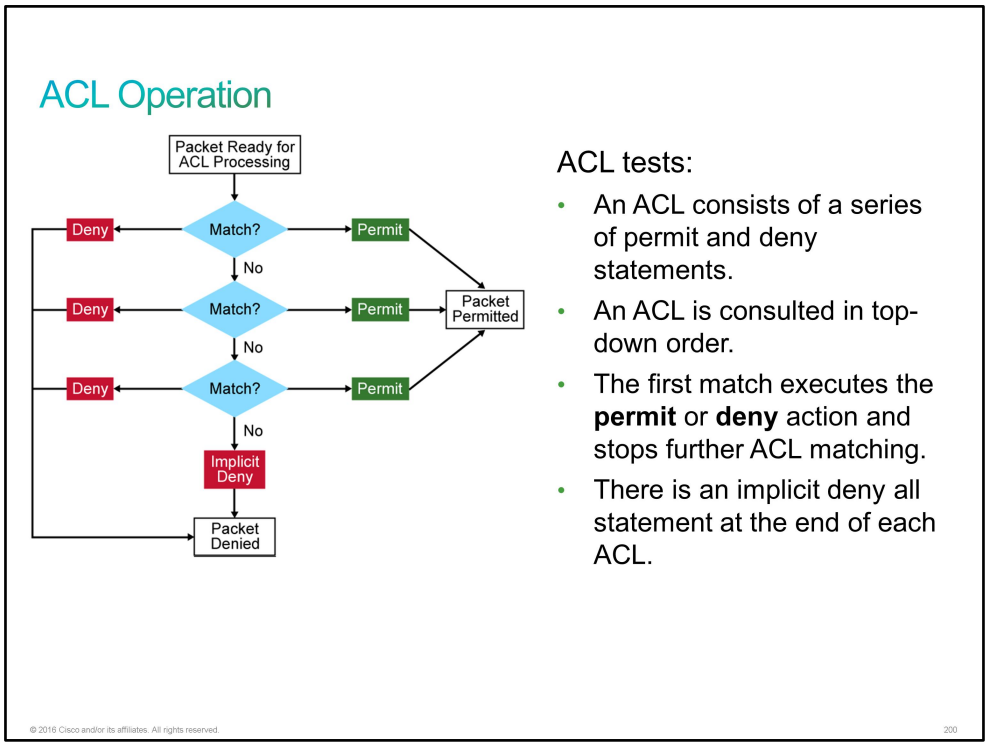


Traffic identification is based on the header values in the IP packet. Identified traffic can receive different treatment, depending on which Cisco IOS function is using the ACLs.

ACLs are supported on a wide range of products, including routers and switches.

ACL Operation

ACL statements operate in a sequential, logical order. They evaluate packets from top down, one statement at a time. If a packet header and an ACL statement match, the rest of the statements in the list are skipped. The packet is then permitted or denied, as determined by the matched statement. If a packet header does not match an ACL statement, the packet is tested against the next statement in the list. This matching process continues until the end of the list is reached.



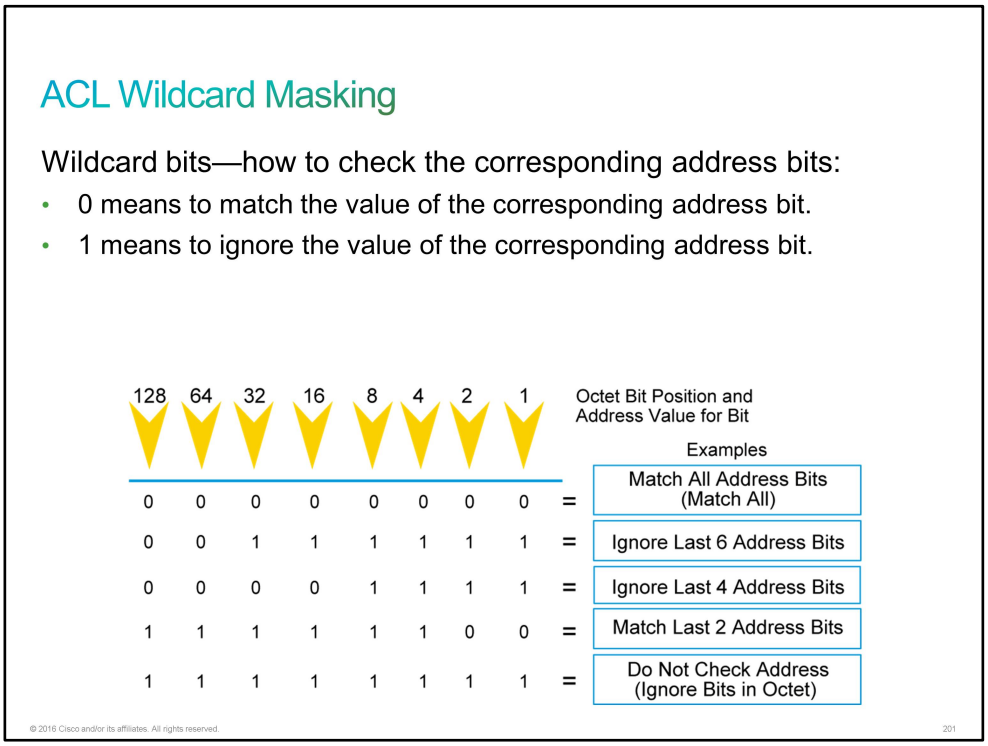
A final implied statement covers all packets for which conditions did not test true. This final test condition matches all other packets and results in a deny instruction. The router denies all these remaining packets. This final statement is often referred to as the "implicit deny any" statement. Because of this statement, an ACL should have at least one permit statement in it; otherwise, the ACL denies all packets.

ACL Wildcard Masking

When processing ACLs, a router needs a mechanism to determine which bits of an IP address must match. A wildcard mask describes which bits of an IP address must match in an IP packet to result in a match for a permit or deny statement.

ACL statements include masks, which are also called *wildcard masks*. A wildcard mask is a string of binary digits that tell the router which parts of the subnet number to look at. Although wildcard masks have no functional relationship with subnet masks, they do provide a similar function. The mask determines how much of an IP source or destination address to apply to the address match. The numbers 1 and 0 in the mask identify how to treat the corresponding IP address bits. However, they are used for different purposes and follow different rules.

Wildcard masks and subnet masks are both 32 bits long and use binary 1s and 0s. Subnet masks use binary 1s and 0s to identify the network, subnet, and host portion of an IP address. Wildcard masks use binary 1s and 0s to filter individual or groups of IP addresses to permit or deny access to resources based on an IP address. By carefully setting wildcard masks, you can permit or deny a single or several IP addresses.

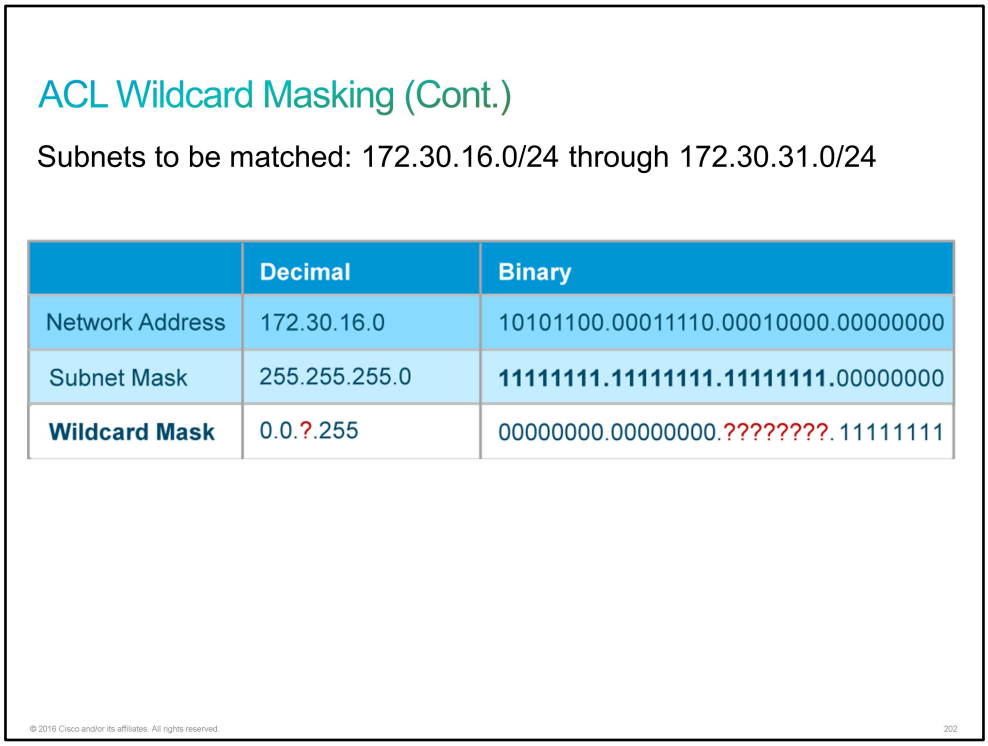


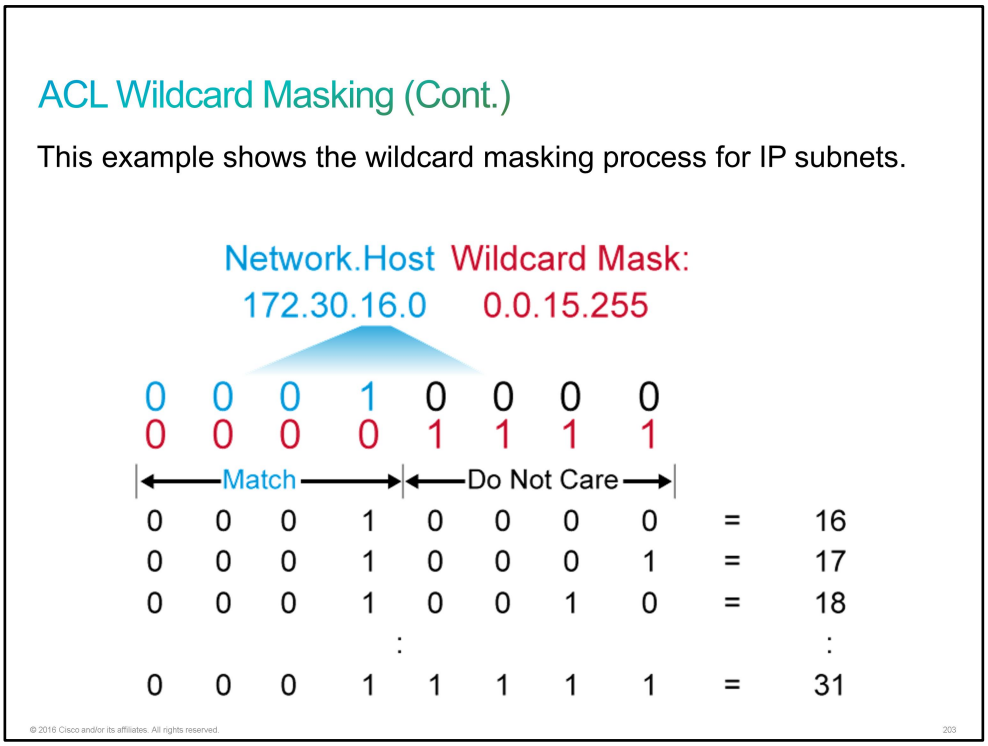
The figure shows how different wildcard masks filter IP addresses. As you look at the example, remember that binary 0 signifies a match and that binary 1 signifies ignore.

Note	A wildcard mask is sometimes referred to as an *inverse mask*. In a subnet mask, binary 1 is equal to a match and binary 0 is not a match. The reverse is true for wildcard masks. A 0 in a bit position of the wildcard mask indicates that the corresponding bit in the address must be matched. A 1 in a bit position of the wildcard mask indicates that the corresponding bit in the address is not interesting and can be ignored.

By carefully setting wildcard masks, you can permit or deny with one ACL statement. You can select a single IP address or many IP addresses.

Assume that you have subnetted your standard Class B address, and you want to create a wildcard mask that matches subnets 172.30.16.0/24 through 172.30.31.0/24. To use one ACL statement to match this range of subnets, use the IP address 172.30.16.0 (the first subnet to be matched) in the ACL, followed by the required wildcard mask. To better understand the process of creating the wildcard mask, look at the figure that follows. The wildcard mask must definitely match the first two octets because the numbers in those two octets are consistent throughout the subnets to be matched. Therefore, the wildcard mask must have all 0s in the first two octets. The wildcard mask must have all 1s in the last octet because it is used for host addresses, and there is no interest in individual hosts. With 1s in the last octet, the wildcard mask ignores the final octet.





In the figure, an administrator wants to test a range of IP subnets that will be either permitted or denied. Assume that the IP address is a Class B address (the first two octets are the network number), with 8 bits of subnetting (the third octet is for subnets). The administrator wants to use the IP wildcard masking bits to match subnets 172.30.16.0/24 to 172.30.31.0/24.

To use one ACL statement to match this range of subnets, use the IP address 172.30.16.0 in the ACL, which is the first subnet to be matched, followed by the required wildcard mask.

First, the wildcard mask matches the first two octets (172 and 30) of the IP address using corresponding 0 bits in the first two octets of the wildcard mask.

Because there is no interest in an individual host, the wildcard mask ignores the final octet by using the corresponding 1 bit in the wildcard mask. For example, the final octet of the wildcard mask is 255 in decimal.

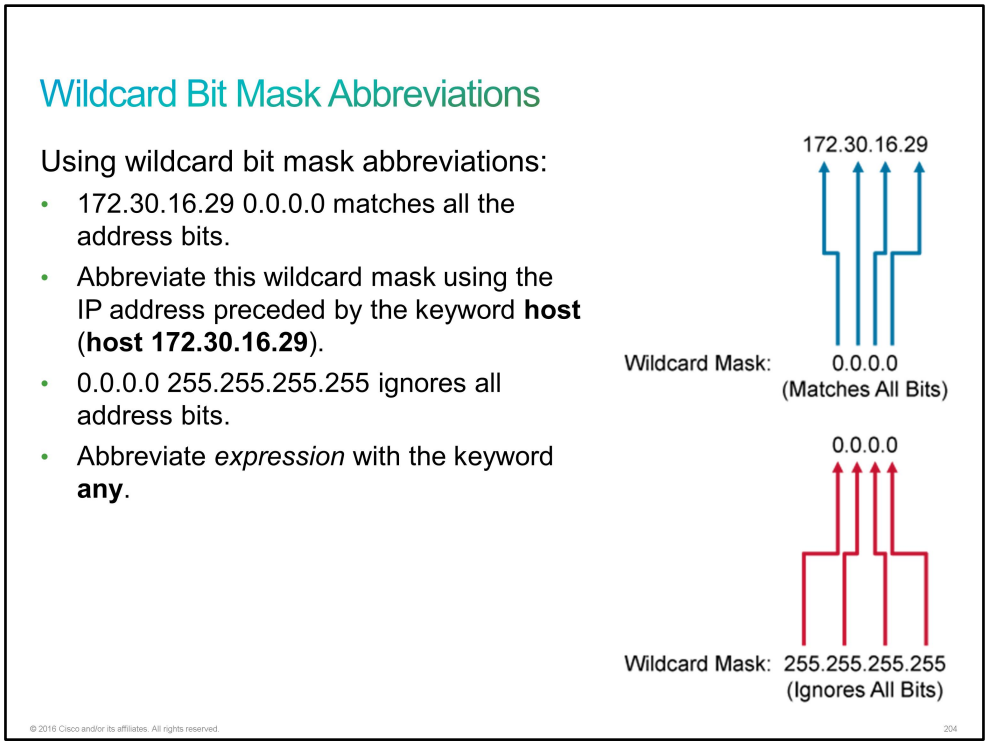
In the third octet, where the subnet address occurs, the wildcard mask of decimal 15, or binary 00001111, matches the high-order 4 bits of the IP address. In this case, the wildcard mask matches subnets starting with the 172.30.16.0/24 subnet. For the final (low-end) 4 bits in this octet, the wildcard mask indicates that the bits can be ignored. In these positions, the address value can be binary 0 or binary 1. Thus, the wildcard mask matches subnet 16, 17, 18, and so on, up to subnet 31. The wildcard mask does not match any other subnets.

In the example, the address 172.30.16.0 with the wildcard mask 0.0.15.255 matches subnets from 172.30.16.0/24 to 172.30.31.0/24.

Sometimes, you must use more than one ACL statement to match a range of subnets. For example, to match 10.1.4.0/24 to 10.1.8.0/24, use 10.1.4.0 0.0.3.255 and 10.1.8.0 0.0.0.255.

Wildcard Bit Mask Abbreviations

The 0 and 1 bit in an ACL wildcard mask cause the ACL to either match or ignore the corresponding bit in the IP address. Working with decimal representations of binary wildcard mask bits can be tedious. For the most common uses of wildcard masking, you can use abbreviations so that you do not have to enter as many numbers when configuring address test conditions.

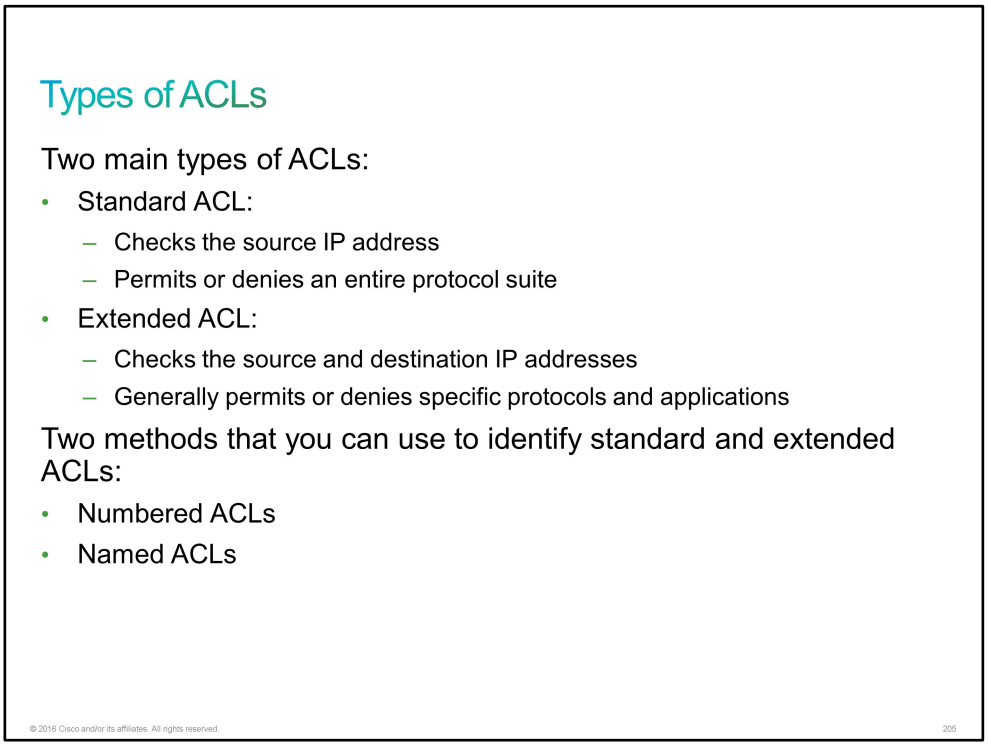


In the example, instead of entering 172.30.16.29 0.0.0.0, you can use the string **host 172.30.16.29**. Using the abbreviation **host** communicates the same test condition to Cisco IOS Software.

In the example, instead of entering 0.0.0.0 255.255.255.255, you can use the keyword **any** by itself. Using the abbreviation **any** communicates the same test condition to Cisco IOS Software.

Types of ACLs

There are two main types of ACLs, standard and extended, and two methods of identifying ACLs, numbering and naming.



ACLs can be categorized into the following types:

- **Standard ACLs:** Standard IP ACLs check the source addresses of the packets that can be routed. The result either permits or denies the output for an entire protocol suite, which is based on the source network, subnet, or host IP address.
- **Extended ACLs:** Extended IP ACLs check both the source and destination packet addresses. They can also check for specific protocols, port numbers, and other parameters, which allows administrators more flexibility and control.

There are two methods that you can use to identify standard and extended ACLs:

- **Numbered ACLs:** Use a number for identification
- **Named ACLs:** Use a descriptive name or number for identification

Types of ACLs (Cont.)

How to identify ACLs:

- Numbered standard IPv4 ACLs (1 to 99) test conditions of all IP packets for source addresses. The expanded range is 1300 to 1999.
- Numbered extended IPv4 ACLs (100 to 199) test conditions of source and destination addresses, specific TCP/IP protocols, and source and destination ports. The expanded range is 2000 to 2699.
- Named ACLs identify IP standard and extended ACLs with an alphanumeric string (name).

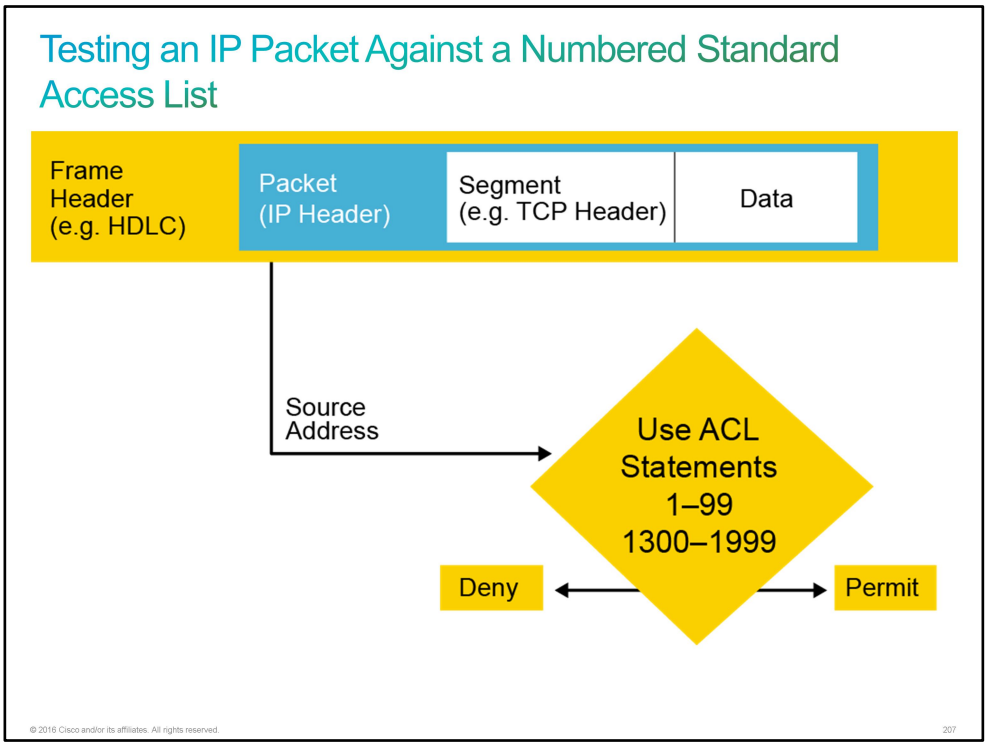
IPv4 ACL Type	Number Range or Identifier
Numbered Standard	1–99, 1300–1999
Numbered Extended	100–199, 2000–2699
Named (Standard and Extended)	Name

You can create many ACLs for a protocol. Select a different ACL number for each new ACL within a given protocol. However, on an interface, you can apply only one ACL per protocol, per direction.

Specifying an ACL number from 1 to 99 or 1300 to 1999 instructs the router to accept numbered standard IPv4 ACL statements. Specifying an ACL number from 100 to 199 or 2000 to 2699 instructs the router to accept numbered extended IPv4 ACL statements.

Testing an IP Packet Against a Numbered Standard Access List

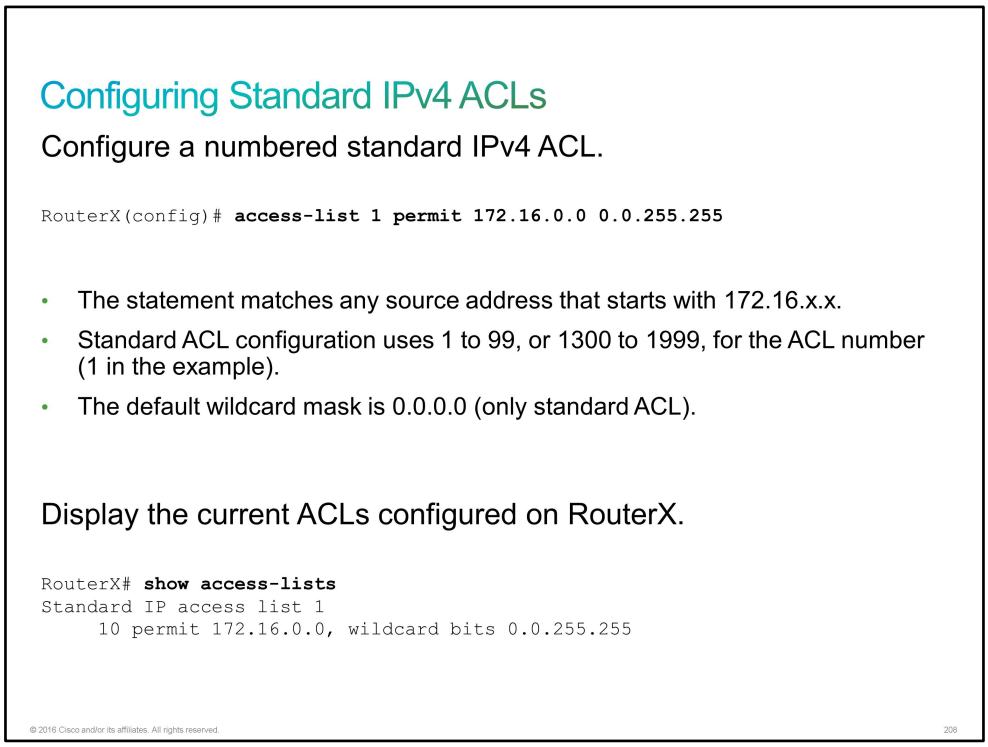
Standard IPv4 ACLs, whether numbered (1 to 99 and 1300 to 1999) or named, filter packets that are based on a source address and mask, and they permit or deny the entire TCP/IP protocol suite.



Note	The standard ACL may not provide the required level of control that you require. You may need a more precise tool for selecting network traffic.

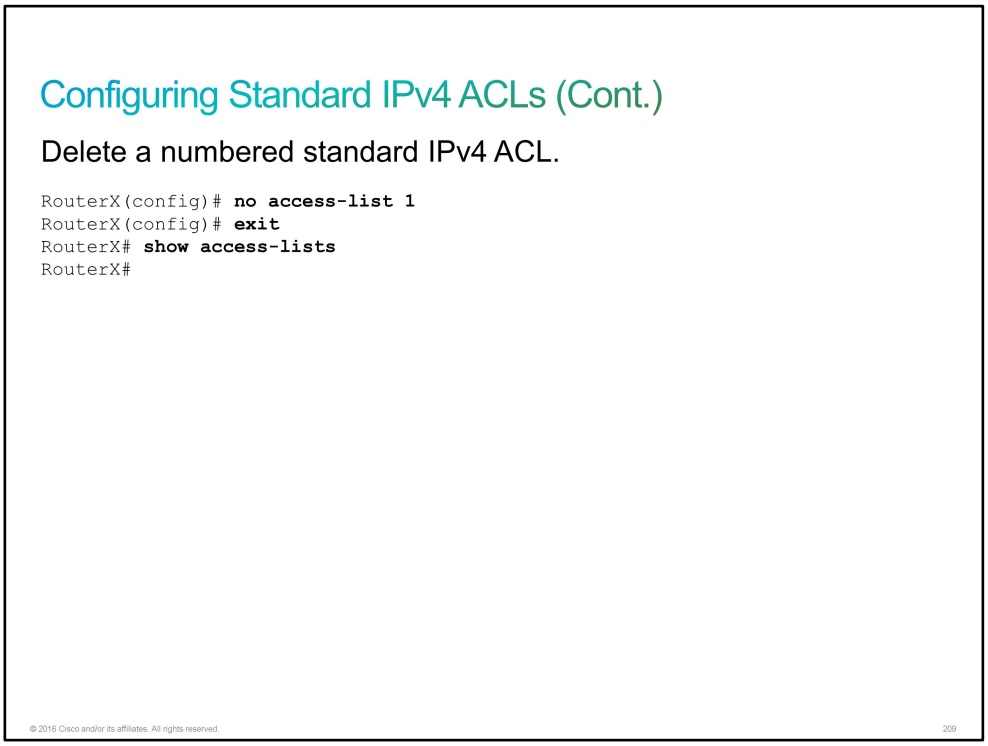
Configuring Standard IPv4 ACLs

You can configure numbered standard IPv4 ACLs on a Cisco router in the global configuration mode. The **access-list** command creates an entry in a standard IPv4 filter list. The following example shows the syntax of this command.



The output of the **show access-list** command displays the current ACLs that are configured on router RouterX.

Use the **no access-list 1** command to remove the entire ACL 1.



To remove the ACL, use the **no access-list** *number* command in the global configuration mode. Issue the **show access-list** command to confirm that ACL 1 has been removed. With numbered ACLs, you cannot remove individual entries with the **no access-list** command, because this command removes the entire ACL. The traditional way of removing or modifying a single numbered ACL entry would be to copy the whole ACL to a text editor, make the changes that are needed, and remove the entire ACL from the router using the **no access-list** command. You can then copy and paste the modified ACL from the text editor. Newer Cisco IOS Software releases allow easier editing by using sequence numbering.

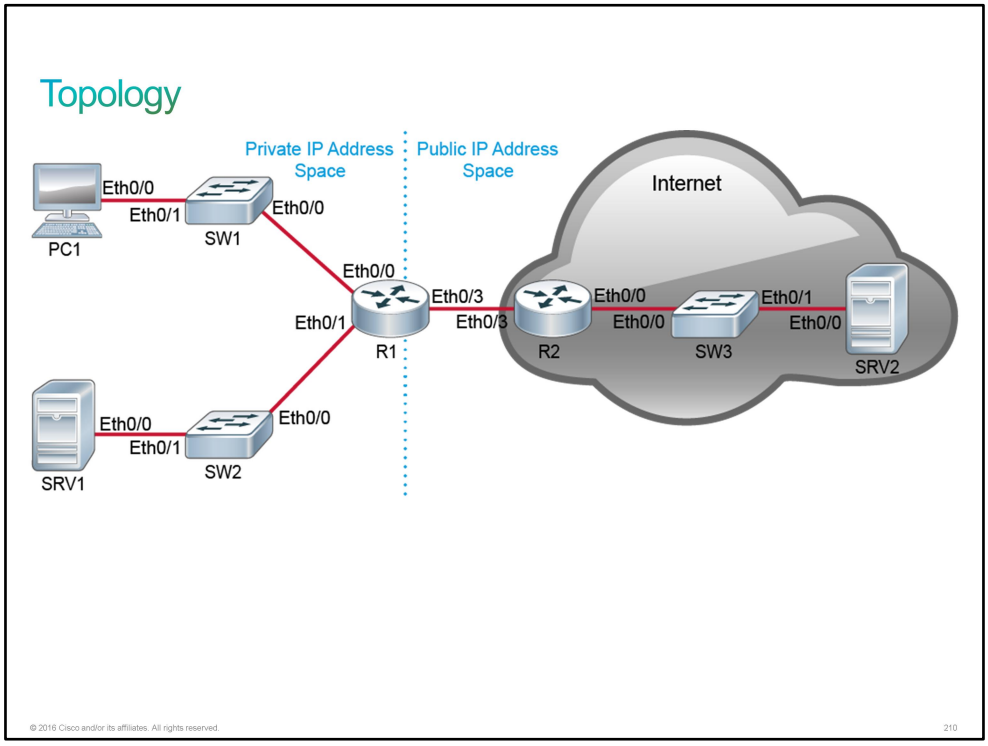
Discovery 11: Configure and Verify ACLs

Introduction

In this discovery lab, you will explore the basics of ACLs. ACLs are also commonly referred to simply as access lists. The lab is prepared with the devices represented in the topology diagram and the connectivity table. All devices have their basic configurations in place including hostnames and IP addresses. Note, neither routing nor NAT has yet been implemented, so there is no connectivity between the private IP address space and the public IP address space.

This discovery lab is broken into two main sections. The first section makes use of an access list that has already been prepared on R1 to demonstrate the importance of statement order in an access list and to demonstrate the effectiveness of using wildcard masks to specify ranges of IP addresses. In the second section of the discovery, you will create a new access list yourself. In both sections, you will demonstrate the function of the ACLs by applying them to the vty lines of R1 using the **access-class** command. When applied in this fashion, the ACL controls which IP addresses are allowed to initiate connections to the EXEC of the router. You will use other devices as the source of Telnet connection attempts to the EXEC of R1.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
SRV1	Hostname	SRV1
SRV1	IP address	10.10.2.20/24
SRV1	Default gateway	10.10.2.1
SRV2	Hostname	SRV2
SRV2	IP address	203.0.113.30/24
SRV2	Default gateway	203.0.113.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.2.4/24
SW2	Default gateway	10.10.2.1
SW2	Ethernet0/0 description	Link to R1
SW2	Ethernet0/1 description	Link to SRV1
SW3	Hostname	SW3
SW3	VLAN 1 IP address	203.0.113.4/24
SW3	Default gateway	203.0.113.1

Device	Characteristic	Value
SW3	Ethernet0/0 description	Link to R2
SW3	Ethernet0/1 description	Link to SRV2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/1 description	Link to SW2
R1	Ethernet0/1 IP address	10.10.2.1/24
R1	Ethernet0/3 description	Link to R2
R1	Ethernet0/3 IP address	198.51.100.2/30
R2	Hostname	R2
R2	Ethernet0/0 description	Link to SW3
R2	Ethernet0/0 IP address	203.0.113.1/24
R2	Ethernet0/3 description	Link to R1
R2	Ethernet0/3 IP address	198.51.100.1/30

PC and SRVs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Configure Numbered Standard IPv4 ACLs

Activity

Step 1 On the R1 router, display access list 10 by passing the output of the **show run** command through the **include** filter, specifying the string **access-list 10**.

To specify a string that includes space characters to the include or section filters, enclose the string in a pair of parentheses.

```
R1# sh run | include access-list 10
access-list 10 permit 10.10.1.10
access-list 10 deny   10.10.1.0 0.0.0.255
access-list 10 permit 10.10.0.0 0.0.255.255
access-list 10 deny   10.0.0.0 0.255.255.255
access-list 10 permit any
```

This ACL demonstrates the use of progressively less restrictive wildcard masks.

Each line in the ACL becomes progressively less specific. Using the asterisk (*) character to represent any octet (0–255), the access list could be interpreted as follows:

- Permit 10.10.1.10
- Deny 10.10.1.*
- Permit 10.10.*.*
- Deny 10.*.*.*
- Permit *.*.*.*

Step 2 To demonstrate this access list in action, you will enable Telnet access on R1 and assign this access list to its vty lines. You will perform this task in the next several steps. First, enter the global configuration mode on R1, then enter the line configuration mode for the five vty lines.

Configuration of vty lines for remote access may not be familiar to you yet. Do not worry. You will be walked through the example.

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line vty 0 4
R1(config-line)#
```

Step 3 The simplest way to enable Telnet on the vty lines is to define a line password. Set the vty line password to "Cisco123."

Cisco IOS passwords are case-sensitive.

```
R1(config-line)# password Cisco123
```

Step 4 Apply access list 10 to the vty lines in the inbound direction using the **access-class** command.

On R1, enter the following commands:

```
R1(config-line)# access-class 10 in
```

Defining an ACL on a Cisco IOS device does not change the behavior of the device until the ACL is assigned in some fashion. Different commands are used in different configuration modes to assign ACLs for different reasons. In this case, access list 10 will control the addresses that are permitted to initiate remote access connections to the EXEC of R1.

Step 5 Leave the configuration mode on R1.

On R1, enter the following commands:

```
R1(config-line)# end
R1#
```

Step 6 View access list 10, this time using the **show access-list** command.

On R1, enter the following command:


```
R1# show access-list 10
Standard IP access list 10
    10 permit 10.10.1.10
    20 deny   10.10.1.0, wildcard bits 0.0.0.255
    30 permit 10.10.0.0, wildcard bits 0.0.255.255
    40 deny   10.0.0.0, wildcard bits 0.255.255.255
    50 permit any
```

The **show access-list** command shows the numbered ACL in a different format than is used to configure it in the global configuration mode. You will soon see that there is another important aspect to the **show access-list** command.

Step 7 The first line in access list 10 permits the unique address 10.10.1.10. This address is the IP address of PC1. Access the console of PC1 and execute a Telnet to R1. Authenticate with the "Cisco123" password. Note the change in the system prompt, indicating that you are connected to R1.

On PC1, enter the following commands:

```
PC1# telnet 10.10.1.1
Trying 10.10.1.1 ... Open


User Access Verification

Password: Cisco123
R1>
```

Step 8 You are now using the console of PC1 to access the EXEC of R1 via a Telnet session. Execute the **show ip interface brief** command to provide further evidence that you are connected to R1 via PC1. You should see the status of the R1 interfaces, but not the PC1 interfaces.

```
R1> sh ip int brief
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES NVRAM  up                    up
Ethernet0/1                10.10.2.1       YES NVRAM  up                    up
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                198.51.100.2    YES NVRAM  up                    up
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  unassigned      YES NVRAM  administratively down
down
Serial1/2                  unassigned      YES NVRAM  administratively down
down
Serial1/3                  unassigned      YES NVRAM  administratively down
down
```

Step 9 Use the **exit** command (or equivalently the **logout** command) to terminate the Telnet session and return to the CLI of PC1. Note that the system prompt returns to PC1>.

```
R1> exit

[Connection to 10.10.1.1 closed by foreign host]
PC1#
```

Step 10 Return to the console of R1 and once again show access list 10.

On R1, enter the following command:

```
R1# show access-list 10
Standard IP access list 10
    10 permit 10.10.1.10 (2 matches)
    20 deny   10.10.1.0, wildcard bits 0.0.0.255
    30 permit 10.10.0.0, wildcard bits 0.0.255.255
    40 deny   10.0.0.0, wildcard bits 0.255.255.255
    50 permit any
```

Routers maintain a hit counter for each line in an ACL. The **show access-list** command displays the number of times each line in an applied ACL has been matched.

When an ACL is applied to the vty lines with an access class, every permitted connection increments the appropriate line match counter by two, while every denied connection increments the appropriate line match counter by only one.

Understand that when an ACL is processed, the first line that matches is used. For example, note that the address 10.10.1.10 actually matches each line in access list 10, but in processing, it was only the first line that was activated.

Step 11 Assuming that the first line did not match, the second line of access list 10 denies access from 10.10.1.0 0.0.0.255. That is, it denies 10.10.1.*. The SW1 IP address is 10.10.1.4. That address should pass by the first line and match this second line. Access the console of SW1 and use Telnet to attempt to connect to R1. The attempt should fail.

On SW1, enter the following command:

```
SW1# telnet 10.10.1.1
Trying 10.10.1.1 ...
% Connection refused by remote host
```

The connection was refused.

Step 12 Return to the console of R1 and view access list 10 again.

On R1, enter the following command:

```
R1# show access-list 10
Standard IP access list 10
    10 permit 10.10.1.10 (2 matches)
    20 deny   10.10.1.0, wildcard bits 0.0.0.255 (1 match)
    30 permit 10.10.0.0, wildcard bits 0.0.255.255
    40 deny   10.0.0.0, wildcard bits 0.255.255.255
    50 permit any
```

There is a match on the second line due to the connection attempt from SW1. Again, permitted connections increment the hit counter by two while denied connections increment the hit counter only by one.

Step 13 Assuming that neither of the first two lines are matched, the third line of access list 10 permits access from 10.10.0.0 0.0.255.255. That is it permits 10.10.*.*. An example of an address that meets these conditions belongs to SRV1 (10.10.2.20). Access the console of SRV1, use Telnet to connect to R1 (10.10.1.1) and authenticate with the password "Cisco123." At the R1> system prompt, simply exit the session.

On SRV1, enter the following command:

```
SRV1# telnet 10.10.1.1
Trying 10.10.1.1 ... Open
User Access Verification

Password: Cisco123

R1> exit

[Connection to 10.10.1.1 closed by foreign host]
SRV1#
```

Step 14 Return to the console of R1 and show access list 10 one more time.

On R1, enter the following command:

```
R1# show access-list 10
Standard IP access list 10
    10 permit 10.10.1.10 (2 matches)
    20 deny   10.10.1.0, wildcard bits 0.0.0.255 (1 match)
    30 permit 10.10.0.0, wildcard bits 0.0.255.255 (2 matches)
    40 deny   10.0.0.0, wildcard bits 0.255.255.255
    50 permit any
```

The hit counter for the third line was incremented due to the successful connection from SRV1.

Step 15 The fourth line in access list 10 will deny connections from 10.0.0.0 0.255.255.255, assuming that there was not a match on any of the first three lines. That is, it denies 10.*.*.*. There are no appropriate addresses in the lab topology to test the fourth line. The fifth line permits connections from any address that did not match the first four lines. That is, any address that starts with something other than 10. R2, on the public address side of the topology, is an example. Access the console of R2 and execute a Telnet session to the R1 public IP address (198.51.100.2). Authenticate with the password "Cisco123." Exit the session when you reach the R1 command prompt.

On R2, enter the following commands:

```
R2# telnet 198.51.100.2
Trying 198.51.100.2 ... Open
User Access Verification

Password: Cisco123

R1> exit

[Connection to 198.51.100.2 closed by foreign host]
R2#
```

Step 16 Return to the console of R1 and show access list 10 one more time to verify the hit counter on the last line of the ACL.

On R1, enter the following command:

```
R1# show access-list 10
Standard IP access list 10
    10 permit 10.10.1.10 (2 matches)
    20 deny   10.10.1.0, wildcard bits 0.0.0.255 (1 match)
    30 permit 10.10.0.0, wildcard bits 0.0.255.255 (2 matches)
    40 deny   10.0.0.0, wildcard bits 0.255.255.255
    50 permit any (2 matches)
```

Task 2: Filter Traffic Using ACLs

Activity

Step 1 Now it is time to create an ACL yourself. Define a new, numbered standard IP access list. Use the number 20. The ACL should permit the IP addresses of PC1 (10.10.1.10) and SRV1 (10.10.2.20). The ACL should be applied to the vty lines with the **access-class** command. Return to the privileged EXEC after configuring this ACL.

On R1, enter the following commands:

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# access-list 20 permit 10.10.1.10
R1(config)# access-list 20 permit 10.10.2.20
R1(config)# line vty 0 4
R1(config-line)# access-class 20 in
R1(config-line)# end
```

When you applied access list 20 to the vty lines, it overrode the previous application of access list 10 on the vty lines. Only one ACL can be applied inbound on the vty lines at one time. Access list 20 is the only ACL that is applied to the vty lines now.

Step 2 View your new access list.

On R1, enter the following commands:

```
R1# show access-list 20
Standard IP access list 20
    10 permit 10.10.1.10
    20 permit 10.10.2.20
```

Step 3 The ACL explicitly permits the address 10.10.1.10. This address belongs to PC1. Access the console of PC1 and verify that you can use Telnet to connect from there to R1. Exit the Telnet session at the R1> prompt.

On PC1, enter the following commands:


```
PC1# telnet 10.10.1.1
Trying 10.10.1.1 ... Open
User Access Verification

Password: Cisco123

R1> exit

[Connection to 10.10.1.1 closed by foreign host]
PC1#
```

Step 4 The SW1 IP address (10.10.1.4) will not match any lines of access list 20. Understand that there is an implicit "deny everything else" at the bottom of every ACL. Access the console of SW1 and verify that you cannot connect from SW1 to R1 using Telnet.

On SW1, enter the following commands:

```
SW1# telnet 10.10.1.1
Trying 10.10.1.1 ...
% Connection refused by remote host

SW1#
```

Step 5 Return to R1 and show access list 20 to examine the line match counters.

On R1, enter the following commands:

```
R1# show access-list 20
Standard IP access list 20
    10 permit 10.10.1.10 (2 matches)
    20 permit 10.10.2.20
```

The connection from PC1 incremented the match counter on the first line.

There is no record of the failed connection attempt from SW1.

Step 6 Sometimes it is advantageous to add an explicit **deny any** to the end of an ACL. For example, it will allow you to track the number of matches that passed through the previous lines. You can also add the **log** argument to the line, which will generate a syslog message providing further detail about the deny event. Add an explicit **deny any** to the end of access list 20 and specify the **log** argument.

On R1, enter the following commands:

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# access-list 20 deny any log
R1(config)# end
```

Step 7 View the updated ACL.

On R1, enter the following command:

```
R1# show access-list 20
Standard IP access list 20
    10 permit 10.10.1.10 (2 matches)
    20 permit 10.10.2.20
    30 deny   any log
```

Step 8 Return to the console of SW1 and use Telnet to attempt to connect to R1 one more time. Again, the connection should be refused, but this time, it is not blocked implicitly but instead by the explicit deny.

On SW1, enter the following command:

```
SW1# telnet 10.10.1.1
Trying 10.10.1.1 ...
% Connection refused by remote host

SW1#
```

Step 9 Return to the console of R1. You should see that the syslog message has already been displayed to the console.

```
R1#
*Oct 19 12:48:15.906: %SEC-6-IPACCESSLOGNP: list 20 denied 0 10.10.1.4 ->
0.0.0.0, 1 packet
```

The syslog message shows the source IP address of the denied connection (10.10.1.4).

You can configure Cisco IOS devices to send syslog messages to a central event management server for real-time processing and long-term archiving of syslog events. Exploration of this concept is beyond the scope of this discovery.

Step 10 Show access list 20 to verify the incremented match counter on the explicit deny statement.

On R1, enter the following command:

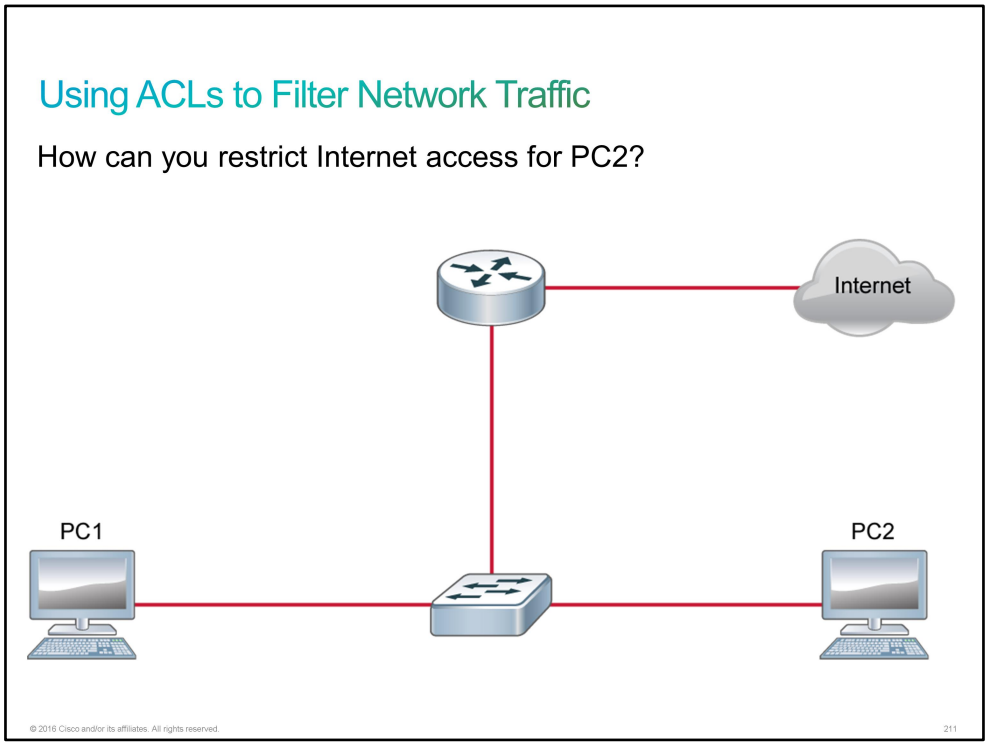
```
R1# show access-list 20
Standard IP access list 20
    10 permit 10.10.1.10 (2 matches)
    20 permit 10.10.2.20
    30 deny   any log (1 match)
```

You have now experimented with some basic ACL implementations. You worked with an existing ACL to test the order of precedence in a standard IP ACL. You also created your own numbered standard IP ACL and tested it. It also included a demonstration of the implicit **deny any** and the utility of adding an explicit **deny any**. Feel free to continue with independent exploration within the lab environment.

This is the end of the discovery lab.

Using ACLs to Filter Network Traffic

The figure introduces a common task for network administrators: a need to implement network traffic filtering to allow, limit, or restrict access to a network resource. A common mechanism that is used for traffic filtering is ACLs, which enable you to control access based on Layer 3 packet-header information.



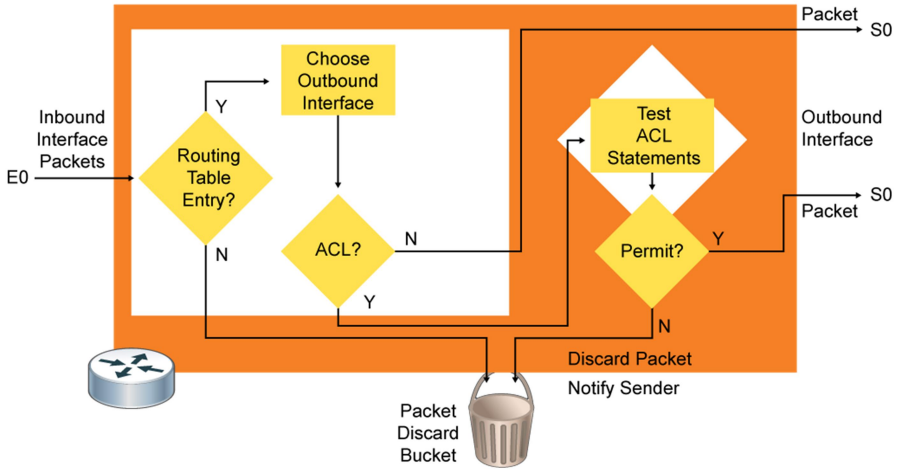
In the scenario in the figure, you can implement traffic filtering on the router either inbound on an interface that is connected to the LAN or outbound on an interface that is connected to the Internet. By using a simple standard ACL, you can prevent packets from PC2 entering or leaving the interface. You should not forget to explicitly allow traffic for other devices in the LAN, such as PC1.

When you use ACLs for traffic filtering, they can operate inbound or outbound. The direction determines at which point packets are tested against the ACL as they pass through the router.

- **Outbound ACLs:** Incoming packets are routed to the outbound interface and then are processed through the outbound ACL. If packets match a permit statement, they are forwarded through the interface. If packets match a deny statement or if there is no match, they are discarded.
- **Inbound ACLs:** Incoming packets are processed by the ACL before they are routed to an outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the filtering tests deny the packet and it is discarded. If the tests permit the packet, it is processed for routing.

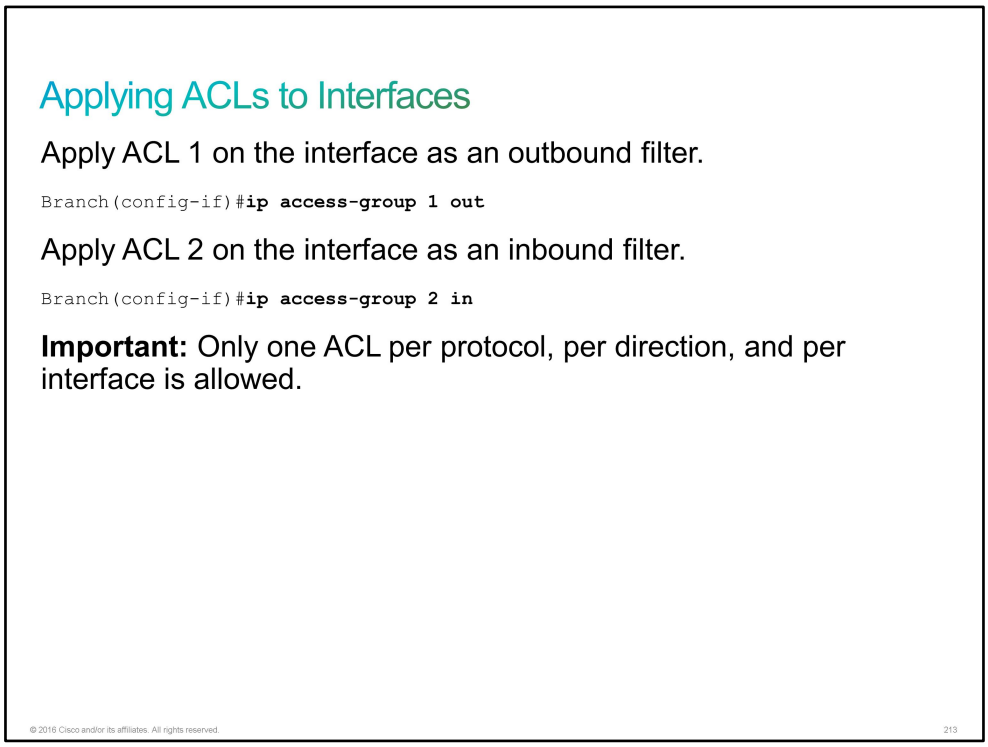
Using ACLs to Filter Network Traffic (Cont.)

ACL operation outbound



Applying ACLs to Interfaces

After you have configured an ACL, you link the ACL to an interface using the **ip access-group** command. Only one ACL per protocol, per direction, and per interface is allowed. The following figure shows examples of this command, showing how to apply the ACL as an inbound and outbound filter.



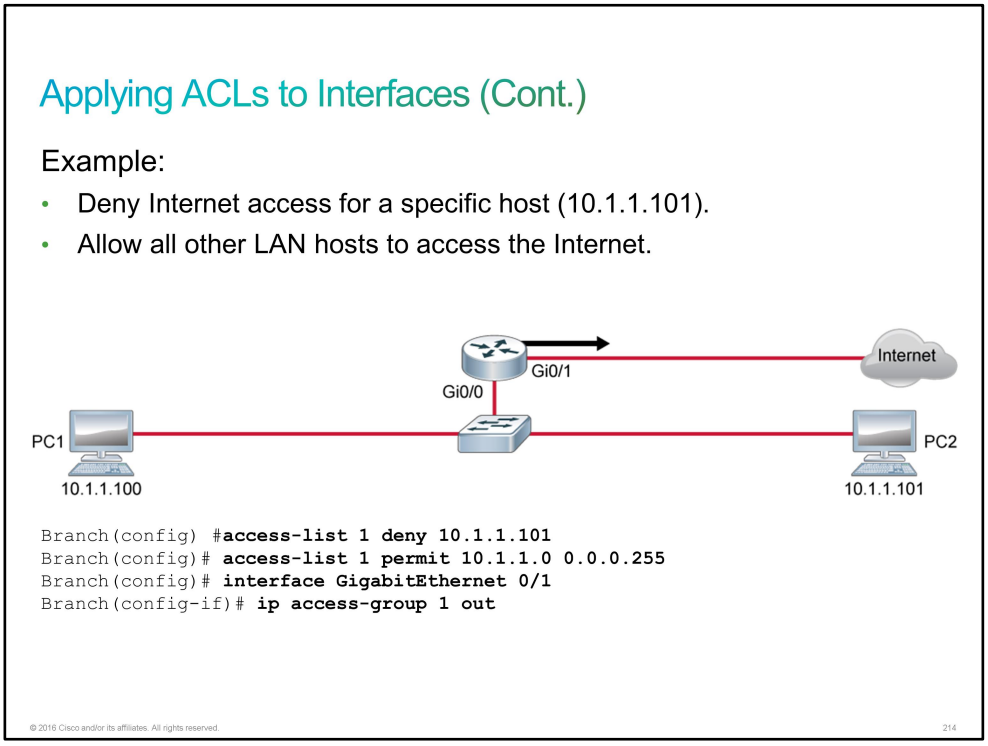
Note	To remove an ACL from an interface, enter the **no ip access-group** command on the interface, then enter the global **no access-list** command to remove the entire ACL if needed.

The table provides an example of the steps that are required to configure and apply a numbered standard ACL on a router.

Numbered Standard ACL Configuration Procedure

Step	Action	Notes
1	Use the **access-list** global configuration command to create an entry in a standard IPv4 ACL. `Branch(config)#` **access-list 1 permit 10.1.1.0 0.0.0.255**	The example ACL statement matches any address that starts with 10.1.1.x.

Step	Action	Notes
2	Use the **interface** configuration command to choose an interface in which to apply the ACL. `Branch(config)#` **interface GigabitEthernet 0/0**	After you enter the **interface** command, the CLI prompt changes from `(config)#` to `(config-if)#`.
3	Use the **ip access-group** interface configuration command to activate the existing ACL on the interface. `Branch(config-if)#` **ip access-group 1 in**	This example activates the standard IPv4 ACL 1 on the interface as an inbound filter.

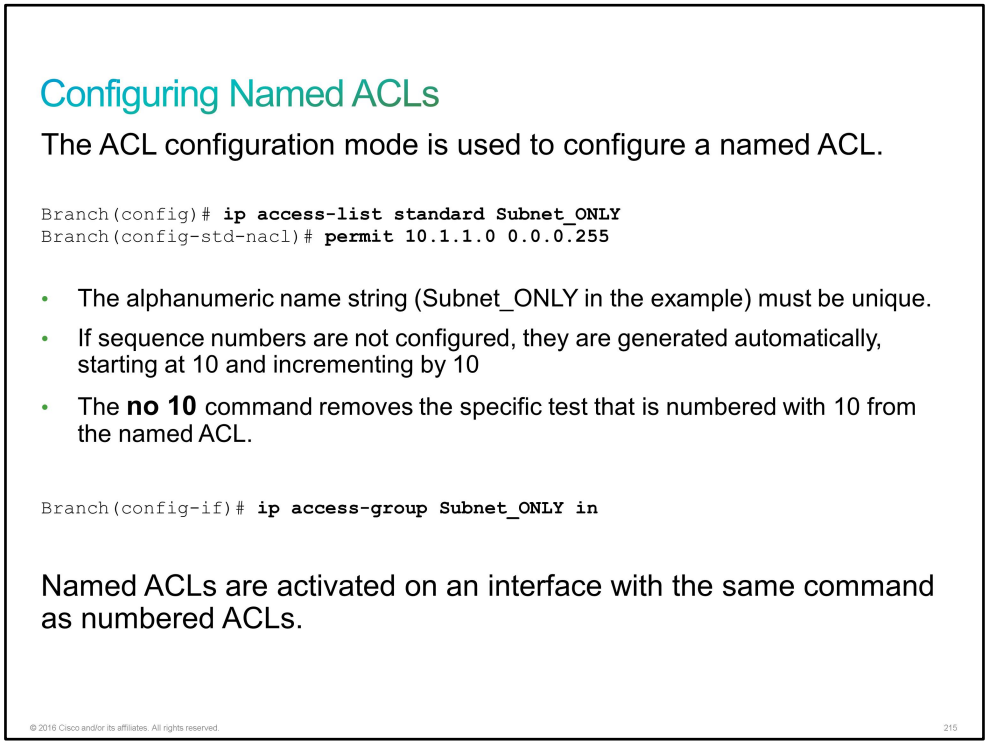


The figure shows a scenario in which ACL 1 is applied in the outbound direction on router Branch to provide traffic filtering. ACL 1 includes a deny statement that matches traffic from a specific host with the IP address 10.1.1.101. The second line in the ACL permits traffic from hosts within the network 10.1.1.0 /24. It is important to specify a permit statement because ACLs end with an implicit deny all statement.

Note	Alternatively, the ACL could be applied in the inbound direction on the interface GigabitEthernet0/0. This solution would not only prevent host PC2 from accessing the Internet but would also deny all communication between PC2 and the router.

Configuring Named ACLs

Designating ACLs by a descriptive name instead of a number provides a better way to describe the intention of the ACL.



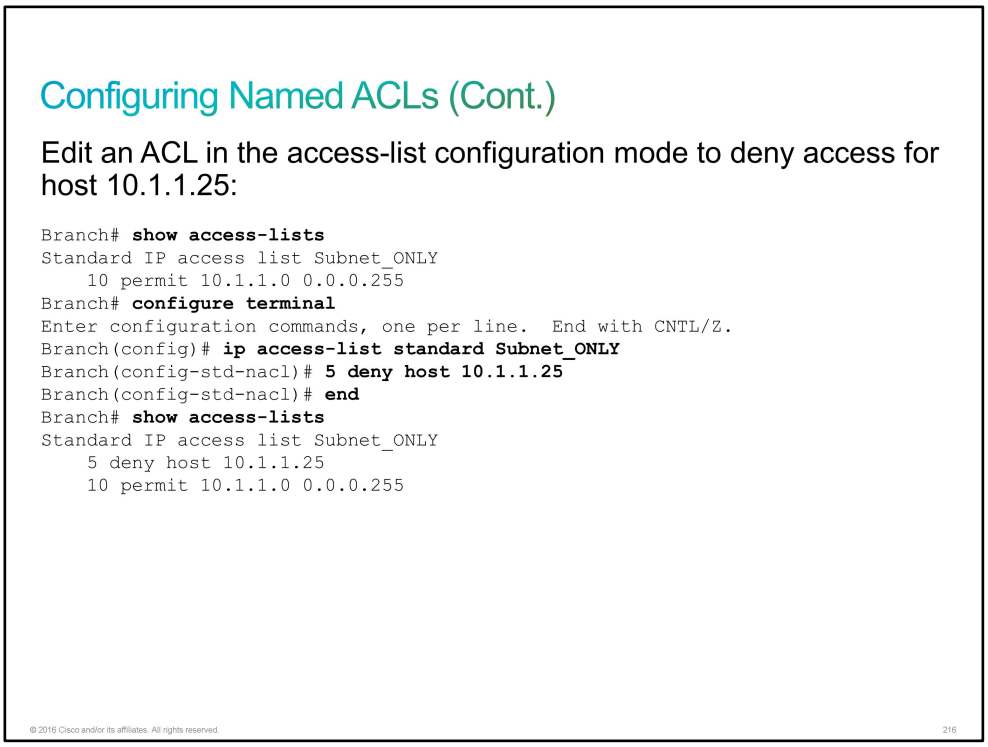
Naming an ACL makes it easier to understand its function. For example, an ACL to deny one subnet could be called "NO_Subnet." When you identify your ACL with a name instead of a number, the configuration mode and command syntax are slightly different.

You use the access-list configuration mode to define named ACLs. To enter this mode, use the **ip access-list** command.

Note	You can also define numbered ACLs using the access-list configuration mode. You simply specify an ACL number instead of a unique name.

In the figure, the command output shows the commands that are used to configure a standard ACL that is named Subnet_ONLY on the Branch router. The ACL permits traffic from hosts on the 10.1.1.0/24 subnet.

Capitalizing ACL names is not required, but it makes them stand out when you view the running configuration output.



Named IP ACLs allow you to add, modify, or delete individual entries in a specific ACL. You can use sequence numbers to insert statements anywhere in the named ACL.

When you add statements to the ACL, the default increment is 10. The figure shows an additional entry that is numbered 5 in the Subnet_ONLY ACL, which is inserted in front of line 10.

Note that a reload will change the sequence numbers in the ACL. The sequence numbers will be 10 and 20 instead of 5 and 10 after the reload. Use the **access-list resequence** command to renumber the ACL entries in an ACL without having to reload.

Challenge

1. Based on the information within the IP packet, an ACL identifies the traffic. True or False?

   A. True
   B. False

2. Which statement about ACLs is true?

   A. An ACL must have at least one permit action, else it just blocks all traffic
   B. ACLs go bottom-up through the entries looking for a match
   C. An ACL has an implicit permit at the end of the ACL.
   D. ACLs will check the packet against all entries looking for a match

3. Which of the following ACL statement will permit only 192.168.1.123 and nothing else?

   A. **access-list 10 permit 192.168.1.123**
      access-list 10 permit 192.168.1.0 0.0.0.255
   B. **access-list 10 permit 192.168.1.123**
      access-list 10 deny 192.168.1.0 0.0.0.255
   C. **access-list 10 deny 192.168.1.0 0.0.0.255**
      access-list 10 permit 192.168.1.123

4. Look at the following ACL Statements and choose the statement that is True.

```
access-list 20 permit 192.168.1.1
access-list 20 deny 192.168.1.0 0.0.0.255
access-list 20 permit 192.0.0.0 0.255.255.255
```

   A. Everything within 192.168.1.0 except 192.168.1.1 will be permitted. Hosts within 192.169.0.0/24 will be blocked.
   B. Everything within 192.168.1.0 except 192.168.1.1 will be permitted. Hosts within 192.169.0.0/24 will be permitted as well.
   C. Everything within 192.168.1.0 except 192.168.1.1 will be blocked. Hosts within 192.169.0.0/24 will be permitted.
   D. Everything except 192.168.1.1 will be blocked.

5. When using wildcard bits, which of the following is true?

   A. 0 means ignore the value of the corresponding address bit.
   B. 1 means match the value of the corresponding address bit.
   C. 1 means ignore the value of the corresponding address bit.
   D. 2 means match the value of the corresponding address bit.

6. In which type of ACL the packets are allowed pass through the router adding extra overhead of routing lookups when ACL filtering discards the packet?

   A. Inbound ACLs
   B. Outbound ACLs

7. Which of the following actions are possible in Named ACLs?

   A. Add a specific ACL entry
   B. Delete a specific ACL entry
   C. Modify a specific ACL entry
   D. All the above

Answer Key

Challenge

1. A
2. B
3. B
4. B
5. C
6. B
7. D

Module 2: Establishing Internet Connectivity

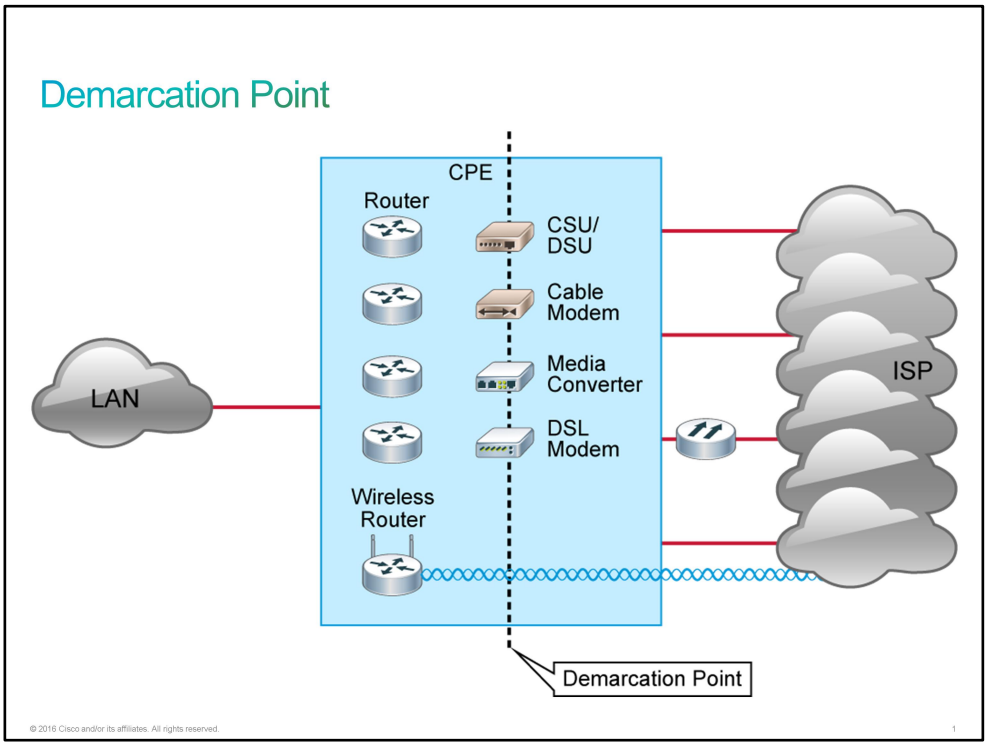
Lesson 9: Enabling Internet Connectivity

Introduction

Your boss dedicates you to the project with a customer who wants to connect to the Internet. The customer is going through the process of obtaining public IP addresses. They are asking about differences between manual IP address assignment and DHCP. Your focus is only to explain to them how DHCP can be used for address assignment by an ISP and what is needed on the customer side. The customer also wants to know about NAT and PAT in case they experience problems with public IP addressing. You should explain to them static and dynamic NAT configuration examples.

Demarcation Point

Although functions within the service provider network are not usually of concern to customers, there are some terms and concepts that you should be familiar with.



Service providers install a connection point (usually in the form of an RJ-45 jack) that physically connects a circuit to their nearest switching office. This link is known as the *demarcation point* and it represents the point at which the responsibility of the service provider is said to end. In other words, the service provider ensures that the link functions correctly up to that point. The other end of this link connects to the service provider network. These links are part of what is known as the local loop or last mile. The local loop may consist of various technologies, including DSL, cable, fiber optics, traditional twisted-pair wiring, and others.

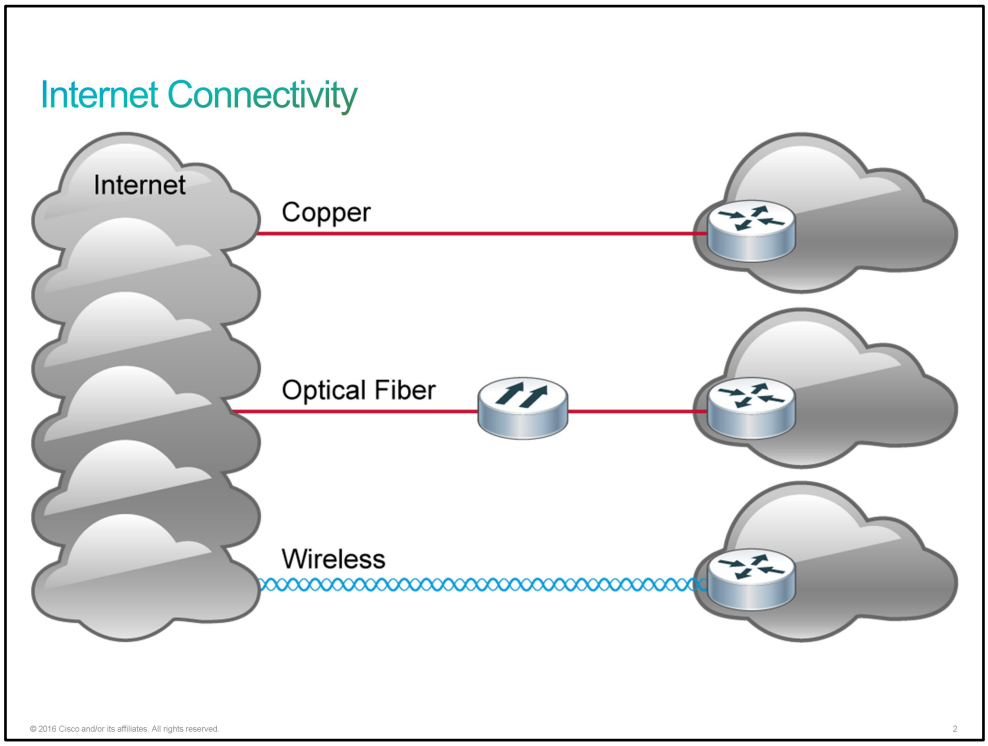
The customer side of the demarcation point is the location of the CPE. The term CPE is often used quite loosely, but it traditionally refers to equipment that is owned and managed by the customer for connecting to the service provider network. However, many companies lease CPE from their service providers, and this equipment is still considered to be CPE. Before physically connecting to a service provider network, a company needs to determine the type of WAN service or connectivity that it requires.

Note	The exact demarcation point is different from country to country. The example that is described is for the United States.

Internet Connectivity

There are three common methods of connecting a small office to the Internet: by a copper medium, optical cable, or by wireless connection.

In wired connections, the medium is either copper, which carries electrical signals, or optical fiber, which carries signals in light. For wireless connections, the medium is the atmosphere of the earth, and radio frequencies carry the signals.



The copper medium includes cables, such as twisted-pair telephone wire, coaxial cable, or (most commonly), Category 6 UTP cable.

Copper medium is used by DSL, cable (DOCSIS), and serial connectivity methods. DSL sends signals across existing telephone lines, whereas cable Internet services leverage the CATV infrastructure.

Optical fibers are thin strands of glass or plastic that transmit digital signals by modulated pulses of light.

Due to its immunity to EMI and RFI, fiber-optic cabling is well suited for harsh environments. The fiber-optic cable medium has the added benefit of extending the distance of cable runs far beyond the capabilities of copper cable.

Wireless Internet carriers offer several different connectivity choices. One option is the home wireless connection between a wireless router and a computer with a wireless network card. Another option is the terrestrial wireless connection between two ground stations. Wireless Internet connectivity can also be achieved through the communication between ground receivers on earth and satellite communication between satellites in geostationary orbit.

The fourth generation (4G), is the fourth generation of mobile telecommunications technology, succeeding 3G and preceding 5G. Applications include amended mobile web access, IP telephony, gaming services, high-definition mobile TV, video conferencing, 3D television, and cloud computing.

WiMAX is a family of wireless communications standards that are designed to provide up to 1 Gbps data rates.

Copper media pros and cons:

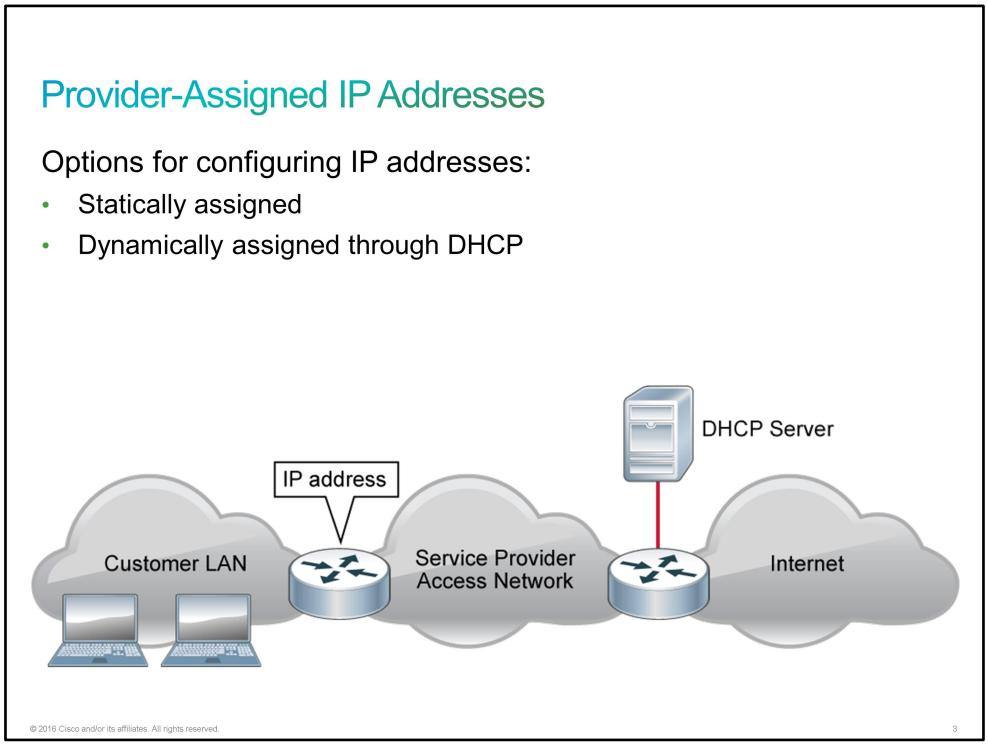
- Better control
- Better security
- More reliability
- High speed
- Relatively cost-effective
- Costly and awkward to maintain

Wireless media pros and cons:

- Freedom to move around the office
- Neater, getting rid of all those unsightly cables
- Productivity benefits
- Harder to secure

Provider-Assigned IP Addresses

The DHCP service enables devices on a network to obtain IP addresses and other information from a DHCP server. This service automates assignment of IP addresses, subnet masks, gateways, and other IP networking parameters.

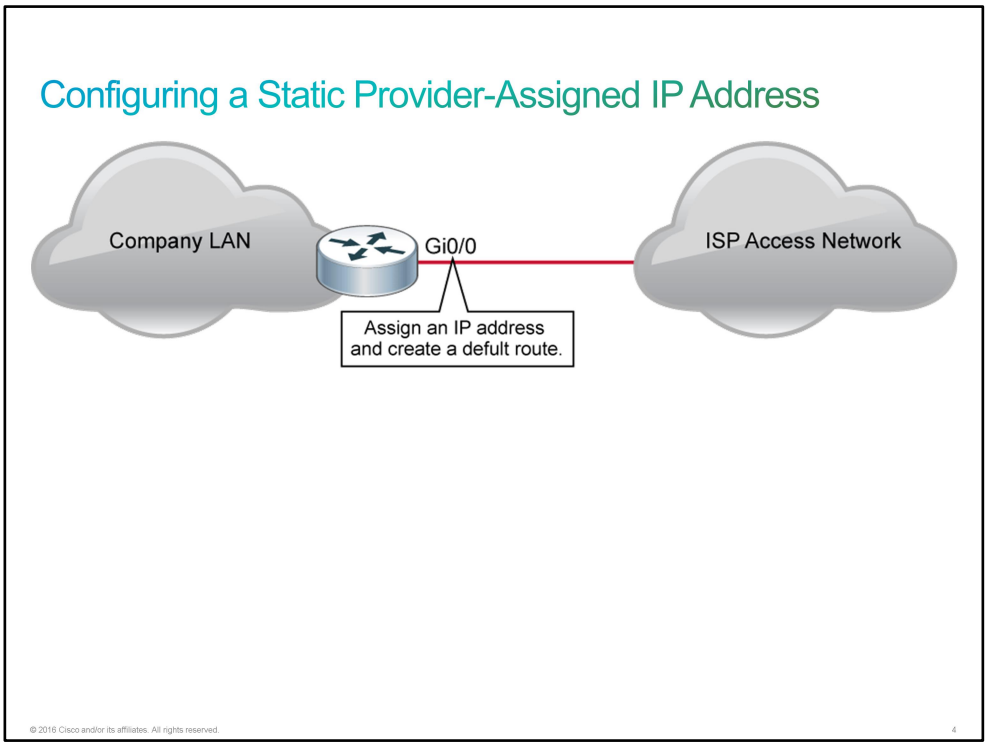


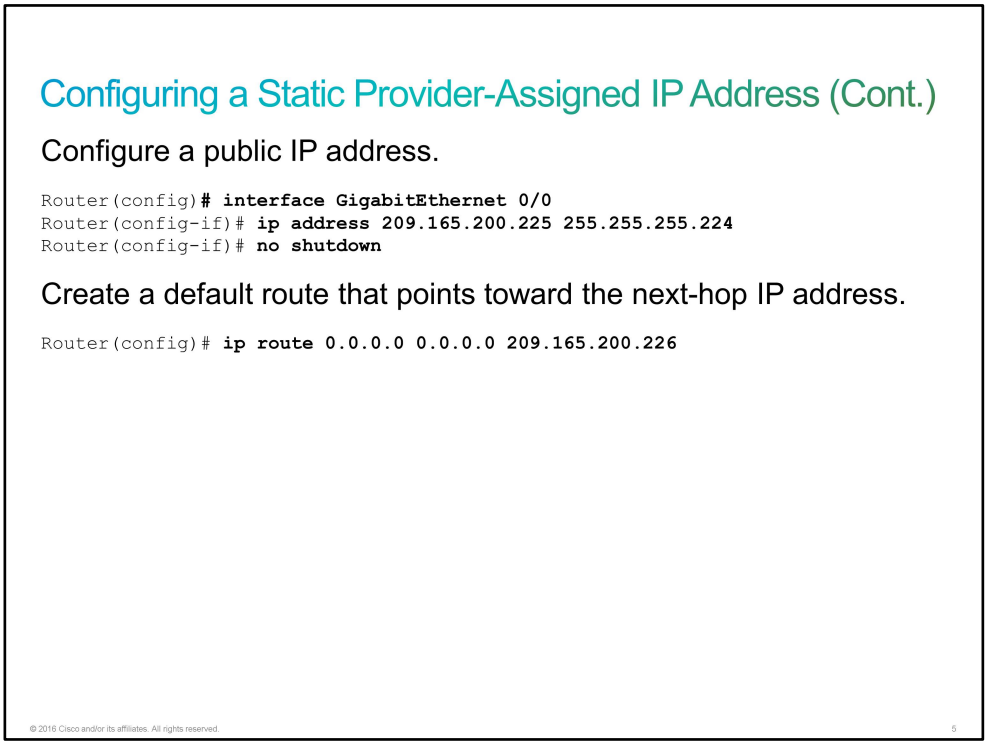
Configuring a Static Provider-Assigned IP Address

A service provider sometimes provides a static address for an interface that is connected to the Internet. In other cases, the address is provided using DHCP. On larger local networks or where the user population changes frequently, DHCP is preferred. New users may arrive with laptops and need a connection. Others have new workstations that need to be connected. Rather than have the network administrator assign an IP address for each workstation, it is more efficient to have IP addresses assigned automatically using DHCP.

If an ISP uses DHCP to provide interface addressing, no manual addresses can be configured. Instead, the interface is configured to operate as a DHCP client. This configuration means that when the router is connected to a cable modem, for example, it is a DHCP client and requests an IP address from the ISP.

Static provider-assigned IP addresses can be more useful in several respects than dynamic addresses. Static IP addresses can be linked to a domain name (such as http://www.cisco.com), and public or private servers can be run for access by outside users.





Static provider-assigned IP addresses can be more useful in several respects than dynamic addresses. Static IP addresses can be linked to a domain name (such as http://www.cisco.com), and public or private servers can be run for access by outside users.

For example, your ISP assigns you a static IP address of 209.165.200.225/27. You proceed with a two-step process. The first step is to configure the static IP address that you were assigned on the outside interface of the router. The second step is to configure a default route that forwards all traffic that is intended for the Internet to the outside interface.

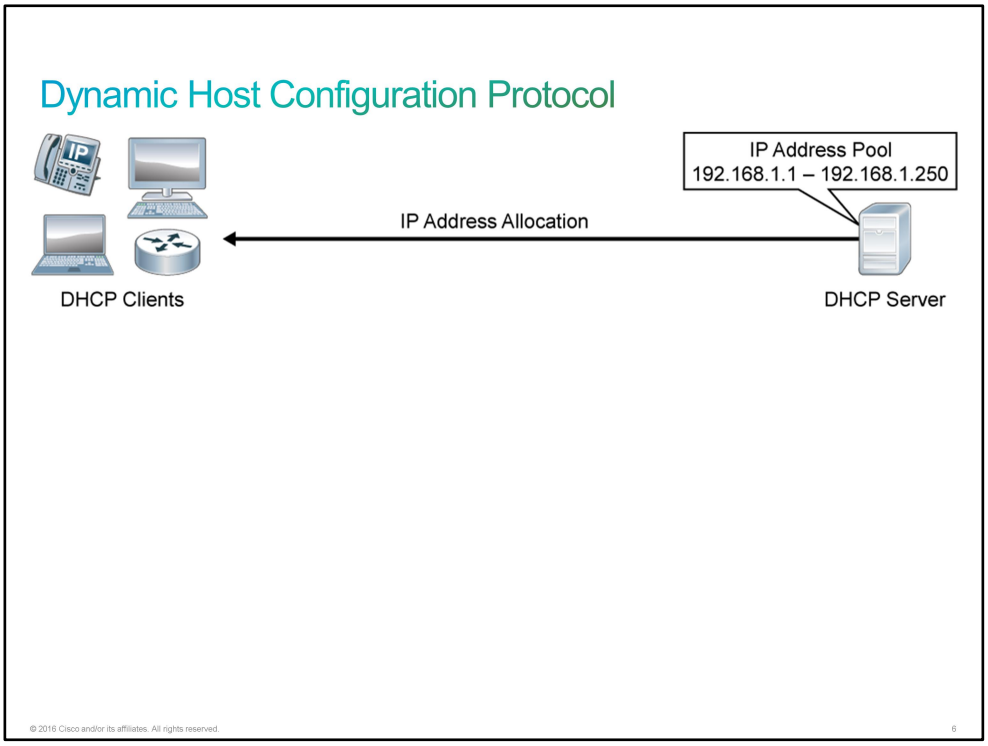
Command	Description
interface *interface*	Specifies an interface and enters the interface configuration mode
ip address *address subnet_mask*	Sets the IP address and mask of the device
no shutdown	Enables an interface
ip route *net-prefix prefix-mask next_hop_ip_address*	Establishes a static route to destination

Dynamic Host Configuration Protocol

Managing a network can be very time-consuming. Network clients break, or are moved, and new clients are purchased that need network connectivity—these tasks are all part of the network administrator job. Depending on the number of IP hosts, manual configuration of IP addresses for every device on the network is virtually impossible.

DHCP can greatly decrease the workload of the network administrator. DHCP automatically assigns an IP address from an IP address pool that the administrator defines. However, DHCP is much more than just a mechanism that allocates IP addresses. This service automates the assignment of IP addresses, subnet masks, gateways, and other IP-required networking parameters.

DHCP is built on a client/server model. The DHCP server hosts allocate network addresses and deliver configuration parameters to dynamically configured hosts. The term "client" refers to a host that is requesting initialization parameters from a DHCP server. Several different devices can be DHCP clients, including Cisco IP phones, desktop PCs, laptops, printers, and even Blu-Ray players. Just about any device that you can configure to participate on a TCP/IP network has the option of using DHCP to obtain its IP configuration.



Depending on the actual DHCP server that is in use, there are three basic DHCP IP address allocation mechanisms:

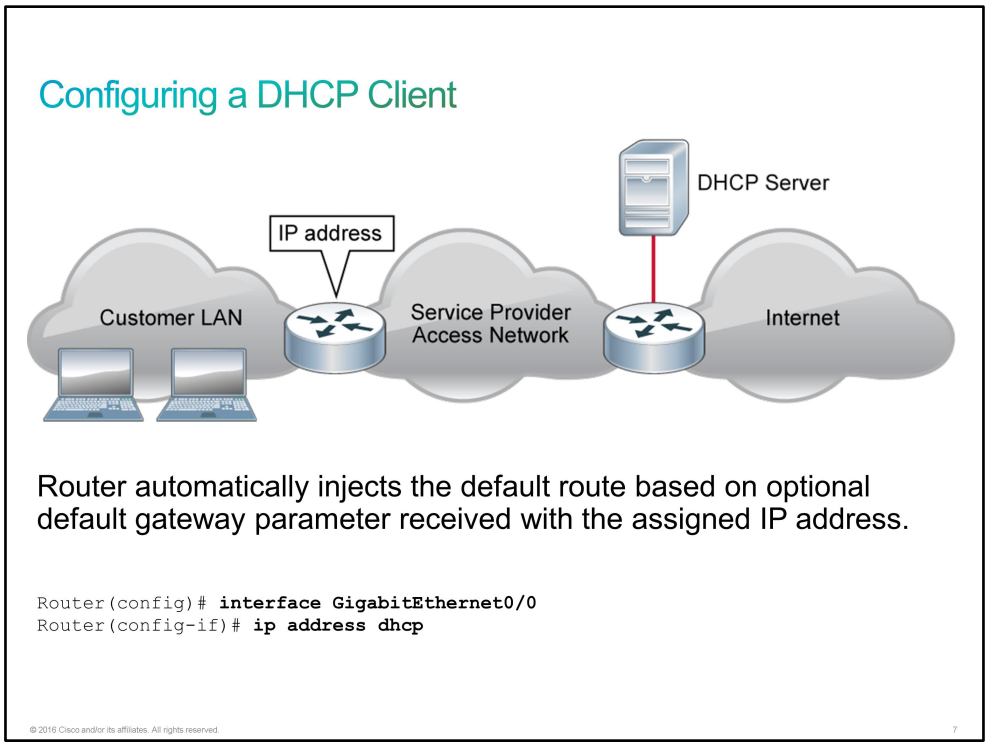
- **Dynamic allocation:** Dynamic allocation of IP addresses is the most common type of address assignment. As devices boot and activate their Ethernet interfaces, the DHCP client service triggers a DHCP Discover broadcast that includes the MAC address of the DHCP client. If a DHCP server is listening on that IP subnet, it responds with a DHCP Offer message. As the name implies, the DHCP Offer message offers an unused IP address from the IP address pool that is on the DHCP server. If the IP address is acceptable, the DHCP client then sends a DHCP Request agreeing to the offered address. The DHCP server then marks the IP address as "in use" in its database and sends a final DHCP ACK to the DHCP client. The DHCP server also starts the countdown on a "lease timer." With a dynamic allocation, a DHCP client is given its IP configuration for a specified amount of time. When the lease time expires, the DHCP server can reclaim the address and return it to the address pool and lease it to another host.
- **Automatic allocation:** Automatic allocation of IP addresses is very similar to dynamic allocation, except that the lease time is set to never expire. This setting results in the DHCP client always being associated with the same IP address.
- **Static allocation:** Static allocation is an alternative that is generally used for devices such as servers and printers, where the device needs to remain at a given address more or less permanently. A static entry is made in the DHCP database that maps the MAC address to an IP address that is not part of the DHCP lease pool.

Note	The following examples show a simplified packet capture of a DHCP request.

```
Source              Destination         Protocol   Info
0.0.0.0             255.255.255.255     DHCP       DHCP Discover
78:ac:c0:52:e8:bd   ff:ff:ff:ff:ff:ff

Source              Destination          Protocol   Info
10.10.1.1           255.255.255.255      DHCP       DHCP Offer 10.10.1.241
00:1b:d5:9c:34:27   ff:ff:ff:ff:ff:ff

Source              Destination         Protocol   Info
0.0.0.0             255.255.255.255     DHCP       DHCP Request
78:ac:c0:52:e8:bd   ff:ff:ff:ff:ff:ff

Source              Destination         Protocol    Info
10.10.1.1           255.255.255.255     DHCP        DHCP ACK
00:1b:d5:9c:34:27   ff:ff:ff:ff:ff
```

Configuring a DHCP Client

An ISP sometimes provides a static address for an interface that is connected to the Internet. In other cases, an address is provided using DHCP. If the ISP uses DHCP to provide interface addressing, no manual address can be configured. Instead, the interface is configured to operate as a DHCP client.

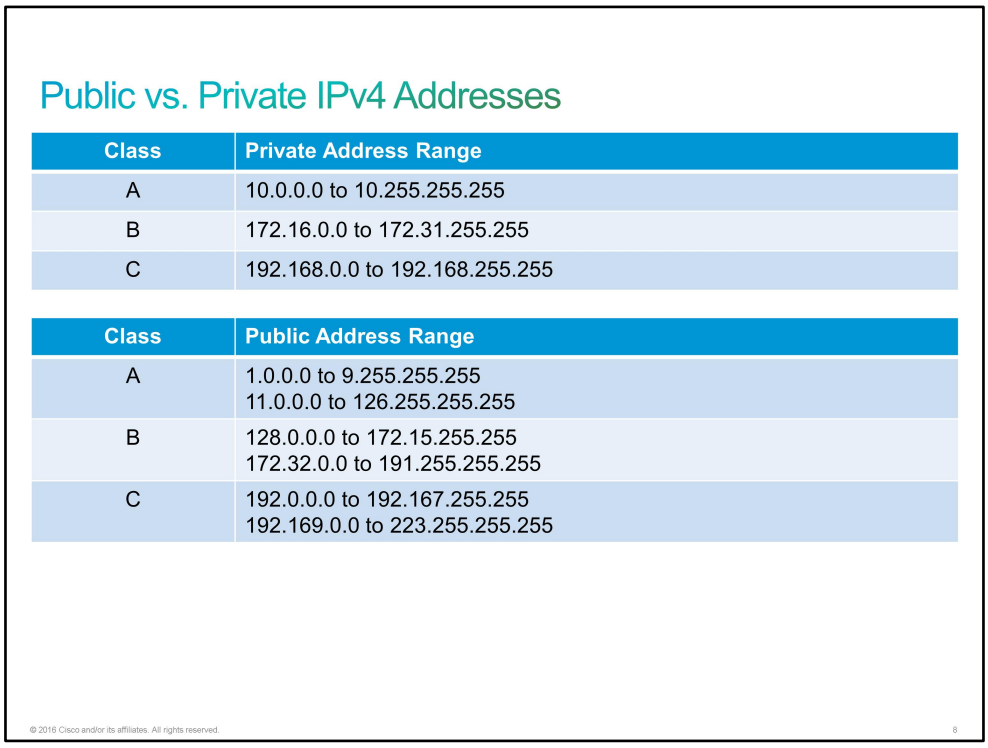


If the router receives an optional DHCP parameter that is called the default gateway with the assigned IP address, the default route will be injected into the routing table, pointing to the default gateway IP address.

Command	Description
interface *interface*	Specifies an interface and enters the interface configuration mode
ip address dhcp	Specifies that the interface acquires an IP address through DHCP

Public vs. Private IPv4 Addresses

Some networks connect to each other through the Internet, while others are private. For instance, the example addresses used in this course are private, which means that they are not assigned to public use. Public and private IP addresses are required for both of these network types.



Private IP Addresses

Internet hosts require a globally unique IP address, but private hosts that are not connected to the Internet can use any valid address, as long as it is unique within the private network. However, because many private networks exist alongside public networks, grabbing "just any address" is strongly discouraged.

Three blocks of IP addresses (one Class A network, 16 Class B networks, and 256 Class C networks) are designated for private, internal use. The table shows the address ranges for each class. Addresses in these ranges are not routed on the Internet backbone. Internet routers are configured to discard private addresses.

In a private intranet, these private addresses can be used instead of globally unique addresses.

When a network that is using private addresses must connect to the Internet, it is necessary to translate the private addresses to public addresses. This translation process is called NAT. A router is often the network device that performs NAT.

Public IP Addresses

Public IP addresses are used for the hosts that are publicly accessible from the Internet. Internet stability depends directly on the uniqueness of publicly used network addresses. Therefore, a mechanism is needed to ensure that addresses are, in fact, unique. This mechanism was originally managed by the InterNIC. IANA succeeded InterNIC. IANA carefully manages the remaining supply of IP addresses to ensure that duplication of publicly used addresses does not occur. Duplication would cause instability in the Internet and would compromise its ability to deliver datagrams to networks using the duplicated addresses.

To obtain a provider-dependent IP address or block of addresses, you must contact an ISP. To obtain provider-independent IP addresses, you must contact an LIR. LIRs obtain IP address pools from their RIRs:

- AfriNIC
- APNIC
- ARIN
- LACNIC
- RIPE NCC

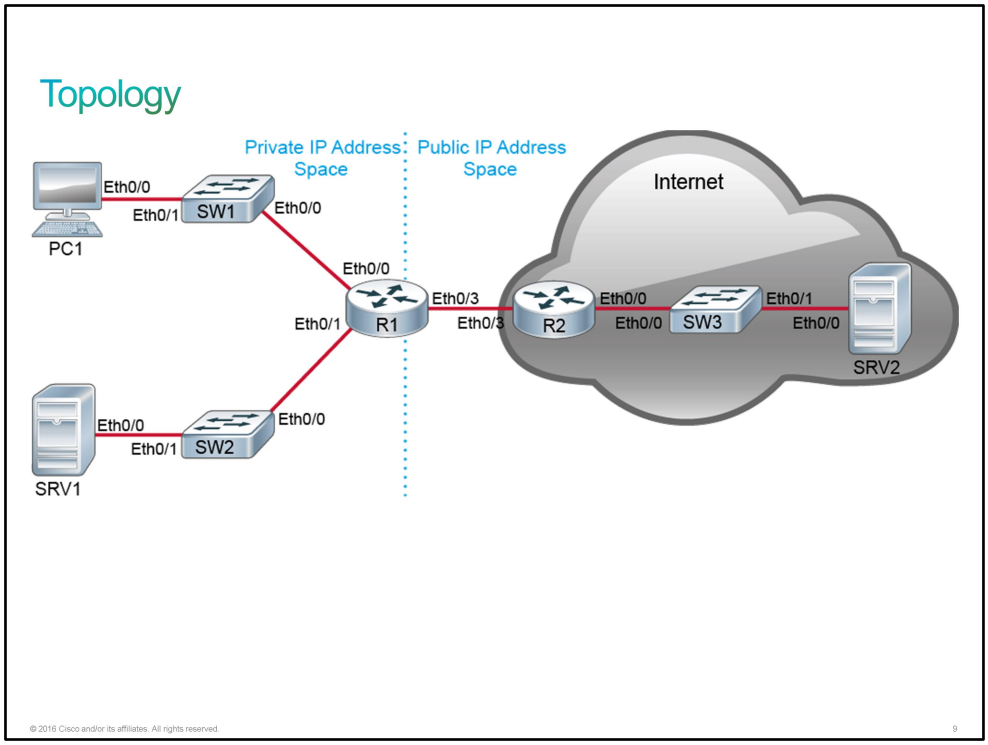
With the rapid growth of the Internet, public IP addresses began to run out. New mechanisms such as NAT, CIDR, VLSM, and IPv6 were developed to help solve the problem.

Discovery 12: Configure a Provider-Assigned IP Address

Introduction

This discovery lab will guide you through the aspects of connecting a small network to the Internet. You will implement the simplest of Internet connections where R1 will receive its IP address via DHCP.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
SRV1	Hostname	SRV1

Device	Characteristic	Value
SRV1	IP address	10.10.2.20/24
SRV1	Default gateway	10.10.2.1
SRV2	Hostname	SRV2
SRV2	IP address	203.0.113.30/24
SRV2	Default gateway	203.0.113.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.2.4/24
SW2	Default gateway	10.10.2.1
SW2	Ethernet0/0 description	Link to R2
SW2	Ethernet0/1 description	Link to PC2
SW3	Hostname	SW3
SW3	VLAN 1 IP address	203.0.113.4/24
SW3	Default gateway	203.0.113.1
SW3	Ethernet0/0 description	Link to R3
SW3	Ethernet0/1 description	Link to SRV2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/1 description	Link to SW2
R1	Ethernet0/1 IP address	10.10.2.1/24

Device	Characteristic	Value
R1	Ethernet0/3 description	Link to R2
R1	Ethernet0/3 IP address	198.51.100.2/24
R2	Hostname	R2
R2	Ethernet0/0 description	Link to SW3
R2	Ethernet0/0 IP address	203.0.113.1/24
R2	Ethernet0/3 description	Link to R1
R2	Ethernet0/3 IP address	198.51.100.1/24

PC and SRVs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Global IP Address Networks

Address Block	Host Starting Address	Host Ending Address	Broadcast Address	Subnet Mask
192.0.2.0/24	192.0.2.1	192.0.2.254	192.0.2.255	255.255.255.0
198.51.100.0/24	198.51.100.1	198.51.100.254	198.51.100.255	255.255.255.0
203.0.113.0/24	203.0.113.1	203.0.113.254	203.0.113.255	255.255.255.0

Task 1: Configure a Provider-Assigned IP Address

Activity

Step 1 To provide some insight into the functioning of a DHCP server, you will configure DHCP server services on R2, which is acting as the ISP router. On R2, access the global configuration with the **configure terminal** command.

On R2, enter the following command:

```
R2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#
```

The most commonly used commands are abbreviated in this guided discovery. For example, **en** is used for **enable** and **conf t** is used for **configure terminal**. If there is any confusion, you can perform tab completion of the commands to see the full commands during the discovery execution. For example, **en<tab>** would expand to **enable** and **conf<tab> t<tab>** would expand to **configure terminal**.

Step 2 Define a DHCP address pool for the subnet 198.51.100.0/24, specifying the interface (198.51.100.1) of R2 as the default gateway and an address lease length of 7 days. DHCP pools are identified with a name.

On R2, enter the following commands:

```
R2(config)# ip dhcp pool Clients
R2(dhcp-config)# network 198.51.100.0 /24
R2(dhcp-config)# default-router 198.51.100.1
R2(dhcp-config)# lease 7
R2(dhcp-config)# exit
R2(config)#
```

Step 3 By default, Cisco IOS DHCP server will serve all IP addresses within the subnet of a defined pool. Limit the assignable addresses from 198.51.100.101 to 198.51.100.254 by excluding the first 100 addresses in the subnet range. Then, leave the configuration mode on R2.

On R2, enter the following commands:

```
R2(config)# ip dhcp excluded-address 198.51.100.1 198.51.100.100
R2(config)# end
R2#
```

Step 4 R1 is currently configured with a static IP address on Ethernet0/3. Verify this fact.

Access the console of R1 and examine the current IP configuration on Ethernet0/3.

```
R1# sh ip int brie
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES NVRAM  up                    up
Ethernet0/1                10.10.2.1       YES NVRAM  up                    up
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                198.51.100.2    YES NVRAM  up                    up
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  unassigned      YES NVRAM  administratively down
down
Serial1/2                  unassigned      YES NVRAM  administratively down
down
Serial1/3                  unassigned      YES NVRAM  administratively down
down
```

Step 5 Routing has not been configured on R1. Verify that only local and connected routes appear in the R1 routing table and that there is no default route configured.

On R1, enter the following command:


```
R1# sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
C        10.10.2.0/24 is directly connected, Ethernet0/1
L        10.10.2.1/32 is directly connected, Ethernet0/1
      198.51.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        198.51.100.0/24 is directly connected, Ethernet0/3
L        198.51.100.2/32 is directly connected, Ethernet0/3
```

Step 6 Reconfigure the interface Ethernet0/3 to obtain its IP address and default gateway via DHCP.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# int eth0/3
R1(config-if)# ip address dhcp
R1(config-if)# end
R1#
```

Step 7 Wait up to 30 seconds. Verify that R1 displays a syslog message indicating that it has been assigned an IP address via DHCP.

```
R1#
*Oct 20 14:47:21.312: %DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/3 assigned
DHCP address 198.51.100.101, mask 255.255.255.0, hostname R1
```

Step 8 Verify that the IP address is assigned to the interface Ethernet0/3.

On R1, enter the following command:

```
R1# show ip int brief
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES NVRAM  up                    up
Ethernet0/1                10.10.2.1       YES NVRAM  up                    up
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                198.51.100.101  YES DHCP   up                    up
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  unassigned      YES NVRAM  administratively down
down
Serial1/2                  unassigned      YES NVRAM  administratively down
down
Serial1/3                  unassigned      YES NVRAM  administratively down
down
```

Step 9 Verify that there is now a default route on R1, using R2 (198.51.100.1) as the default route.

On R1, enter the following command:

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 198.51.100.1 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 198.51.100.1
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
C        10.10.2.0/24 is directly connected, Ethernet0/1
L        10.10.2.1/32 is directly connected, Ethernet0/1
      198.51.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        198.51.100.0/24 is directly connected, Ethernet0/3
L        198.51.100.101/32 is directly connected, Ethernet0/3
```

Step 10 Verify connectivity from R1 to the public IP address side of the topology by pinging SRV2 (203.0.113.30).

On R1, enter the following command:

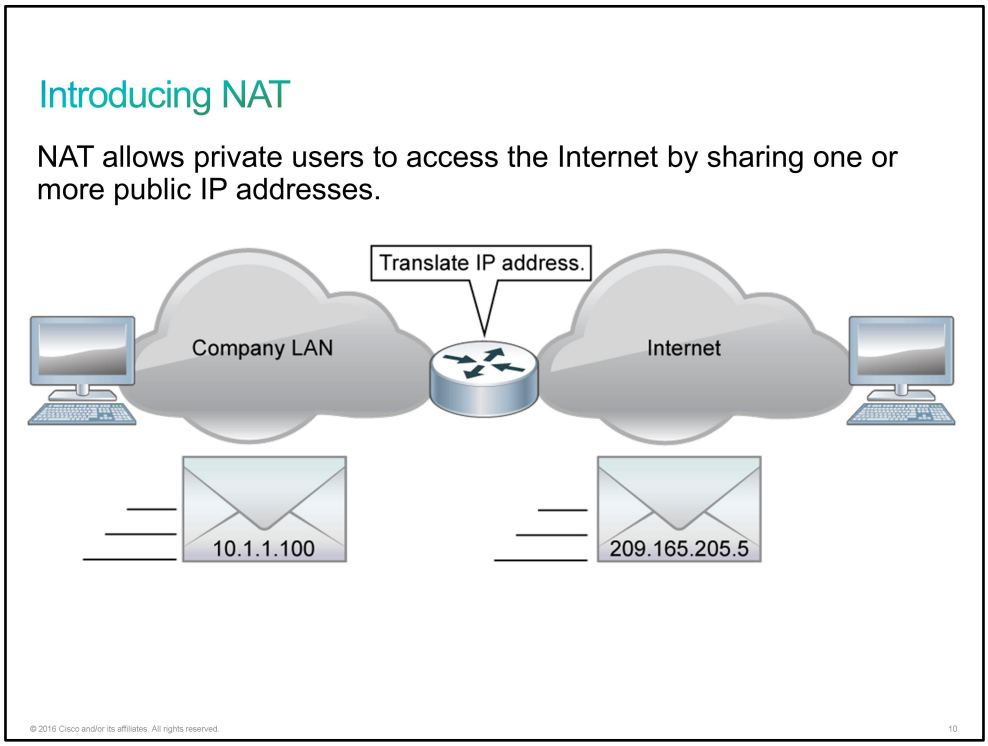
```
R1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

This is the end of the discovery lab.

Introducing NAT

Small networks are commonly implemented using private IP addressing as defined in RFC 1918. Private addressing gives enterprises considerable flexibility in a network design. This addressing enables operationally and administratively convenient addressing schemes and easier growth. However, you cannot route private addresses over the Internet, and there are not enough public addresses to allow all organizations to provide a private address to all their hosts. Therefore, network administrators need a mechanism to translate private addresses to public addresses (and back) at the edge of their network.

NAT provides this mechanism. Before NAT, a host with a private address could not access the Internet. Using NAT, companies can provide some or all their hosts with private addresses and provide address translation to allow access to the Internet.



NAT is like the receptionist in a large office. Assume that you have left instructions with the receptionist not to forward any calls to you unless you request it. Later, you call a potential client and leave a message asking the client to call you back. You tell the receptionist that you are expecting a call from this client, and you ask the receptionist to put the call through to you. The client calls the main number to your office, which is the only number that the client knows. When the caller gives the receptionist your name, the receptionist checks a lookup table that matches your name to your extension. The receptionist knows that you requested this call and forwards the caller to your extension.

Usually, NAT connects two networks and translates the private (inside local) addresses in the internal network to public (inside global) addresses before packets are forwarded to another network. You can configure NAT to advertise only one address for the entire network to the outside world. Advertising only one address effectively hides the internal network, providing additional security as a side benefit.

The network address translation process of swapping one address for another is separate from the convention that is used to determine what is public and private, and devices must be configured to recognize which IP networks should be translated. This requirement is one of the reasons why NAT can also be deployed internally when there is a clash of private IP addresses, such as, for example, when two companies merge.

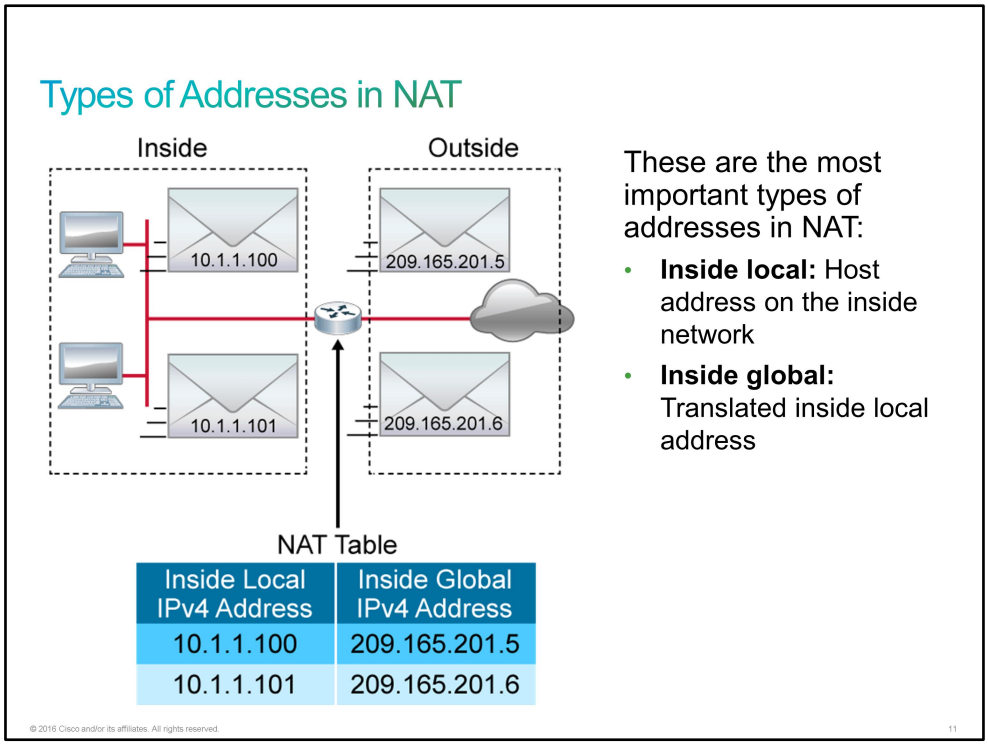
The benefits of NAT are the following:

- Eliminates the need to readdress all hosts that require external access, saving time and money.
- Conserves addresses through application port-level multiplexing. With PAT, multiple internal hosts can share a single registered IPv4 address for all external communication. In this type of configuration, relatively few external addresses are required to support many internal hosts. This characteristic conserves IPv4 addresses.
- Protects network security. Because private networks do not advertise their addresses or internal topology, they remain reasonably secure when they gain controlled external access with NAT.

The disadvantages of NAT are the following:

- Many IP addresses and applications depend on end-to-end functionality, with unmodified packets forwarded from the source to the destination. By changing end-to-end addresses, NAT blocks some applications that use IP addressing. For example, some security applications, such as digital signatures, fail because the source IP address changes. Applications that use physical addresses instead of a qualified domain name do not reach destinations that are translated across the NAT router. Sometimes, you can avoid this problem by implementing static NAT mappings.
- End-to-end IP traceability is also lost. It becomes much more difficult to trace packets that undergo numerous packet address changes over multiple NAT hops, so troubleshooting is challenging. On the other hand, hackers who want to determine the source of a packet find it difficult to trace or obtain the original source or destination address.
- Using NAT also complicates tunneling protocols, such as IPsec, because NAT modifies the values in the headers. This behavior interferes with the integrity checks that IPsec and other tunneling protocols perform.
- 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 makes a specific effort 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 NAT separates both systems from the Internet.
- The last disadvantage involves performance. NAT increases switching delays because translation of each IP address within the packet headers takes time. The first packet is process-switched, meaning that it always goes through the slower path. The router must look at each packet to decide whether it needs translation. The router needs to alter the IP header and possibly alter the TCP or UDP header. Remaining packets go through the fast-switched path if a cache entry exists; otherwise, they too are delayed.

Types of Addresses in NAT



In NAT terminology, the *inside network* is the set of networks that are subject to translation. The *outside network* refers to all other addresses. Usually, these other addresses are valid addresses that are located on the Internet.

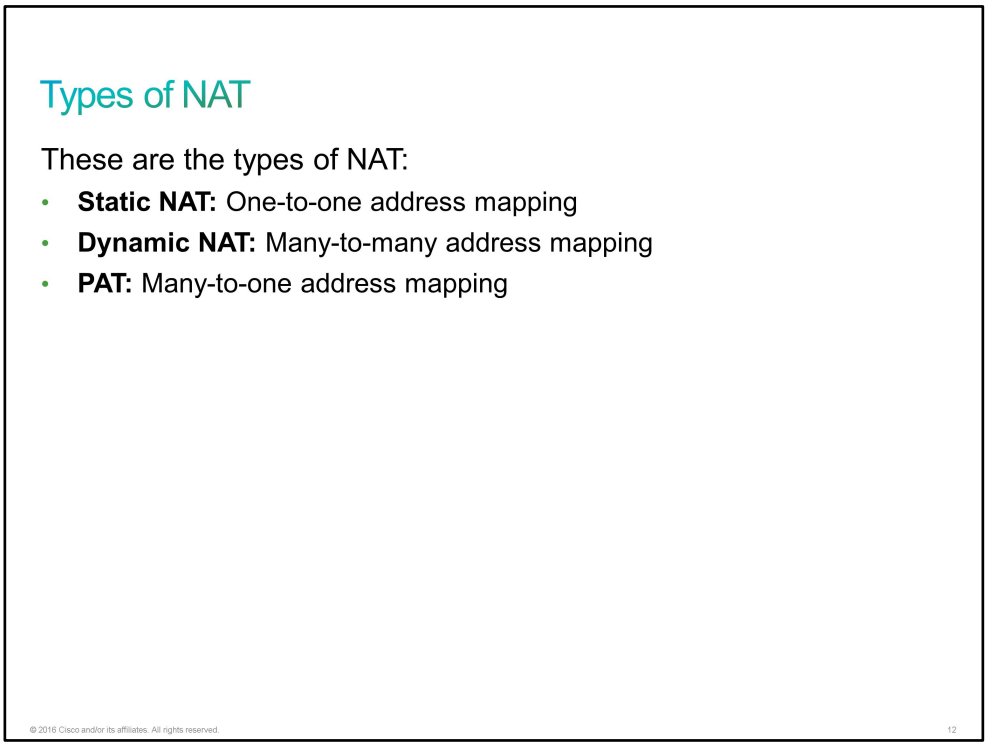
Cisco defines these NAT terms:

- **Inside local address:** The IPv4 address that is assigned to a host on the inside network. The inside local address is likely not an IPv4 address that the network information center or service provider assigns.
- **Inside global address:** The translated inside local address. It is typically a public IPv4 address.
- **Outside global address:** The IPv4 address that the host owner assigns to a host on the outside network. The outside global address is allocated from a globally routable address or network space.
- **Outside local address:** The IPv4 address of an outside host as it appears to the inside network. Not necessarily public, the outside local address is allocated from a routable address space on the inside.

A good way to remember what is local and what is global is to add the word *visible*. An address that is locally visible normally implies a private IP address, and an address that is globally visible normally implies a public IP address. The rest is simple. *Inside* means internal to your network and *outside* means external to your network. So, for example, an inside global address means that the device is physically inside your network and has an address that is visible from the Internet. It could be a web server, for instance.

Types of NAT

On a Cisco IOS router, NAT can be divided into three distinct categories, each having a clear use case.

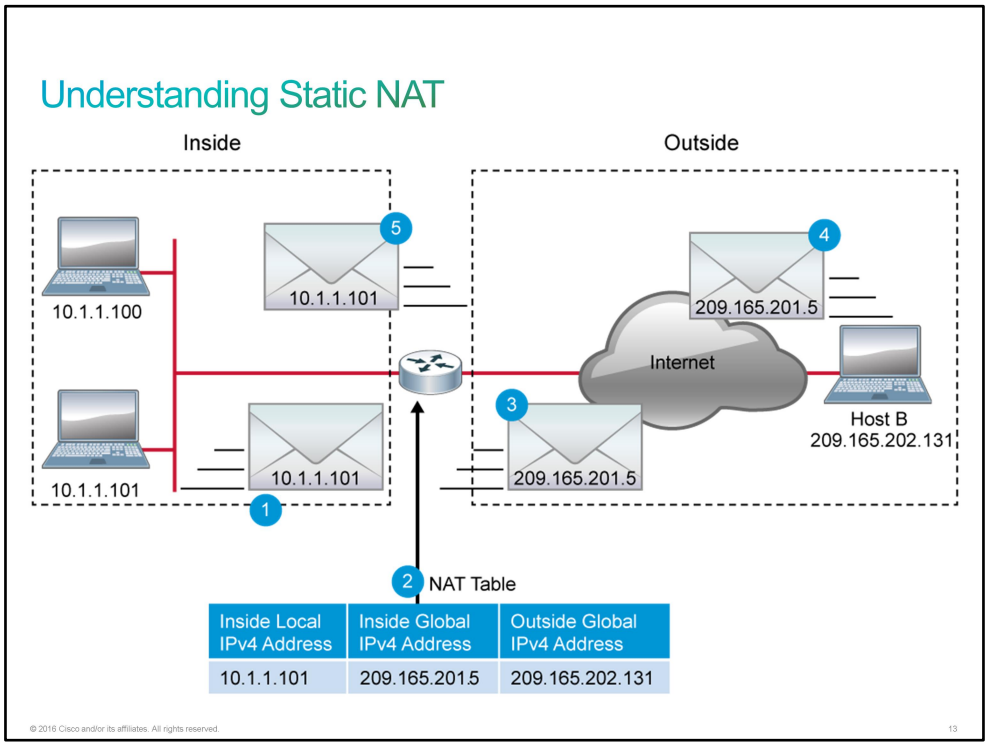


NAT can work in these ways:

- **Static NAT:** Maps a private IPv4 address to a public IPv4 address (one to one). Static NAT is particularly useful when a device must be accessible from outside the network. This type of NAT is used when a company has a server for which it needs a static IP address.
- **Dynamic NAT:** Maps a private IPv4 address to a public IPv4 address from a group of public IPv4 addresses. This type of NAT is used, for example, when two companies that are using the same private address space merge. With the use of dynamic NAT readdressing, using the entire address space is avoided or at least postponed.
- **PAT:** PAT maps multiple private IPv4 addresses to a single public IPv4 address (many to one) by using different ports. PAT is also known as NAT overloading. It is a form of dynamic NAT and is the most common use of NAT. It is used every day in your place of business or your home. Multiple users of PCs, tablets, and phones are able to access the Internet, even though only one public IP address is available for that LAN.

Understanding Static NAT

You can translate your own IPv4 addresses into globally unique IPv4 addresses when you are communicating outside your network.



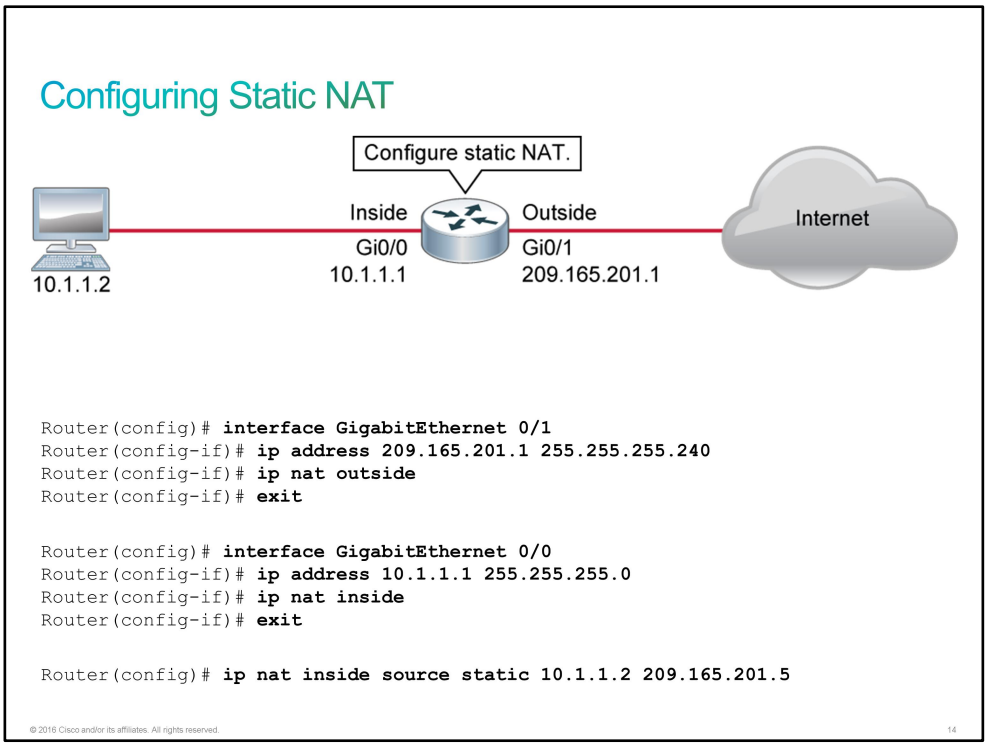
The figure illustrates a router that is translating a source address inside a network into a source address outside the network. The following are the steps for translating an inside source address:

1. The user at host 10.1.1.101 wants to open a connection to Host B (IP address 209.165.202.131).
2. The first packet that the router receives on its NAT inside-enabled interface from host 10.1.1.101 causes the router to check its NAT table.
3. The router replaces the inside local source address of host 10.1.1.101 with the translated inside global address (209.165.201.5) and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.101, using the inside global IPv4 destination address 209.165.201.5.
5. When the router receives the packet on its NAT outside-enabled interface with the inside global IPv4 address of 209.165.201.5, the router performs a NAT table lookup using the inside global address as a key. The router then translates the address back to the inside local address of host 10.1.1.101 and forwards the packet to host 10.1.1.101.
6. Host 10.1.1.101 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Configuring and Verifying Static NAT

Configuring Static NAT

Remember that 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. For instance, you may want to map an inside global address to a specific inside local address that is assigned to your web server. In the following example, you can see how to configure a static NAT.



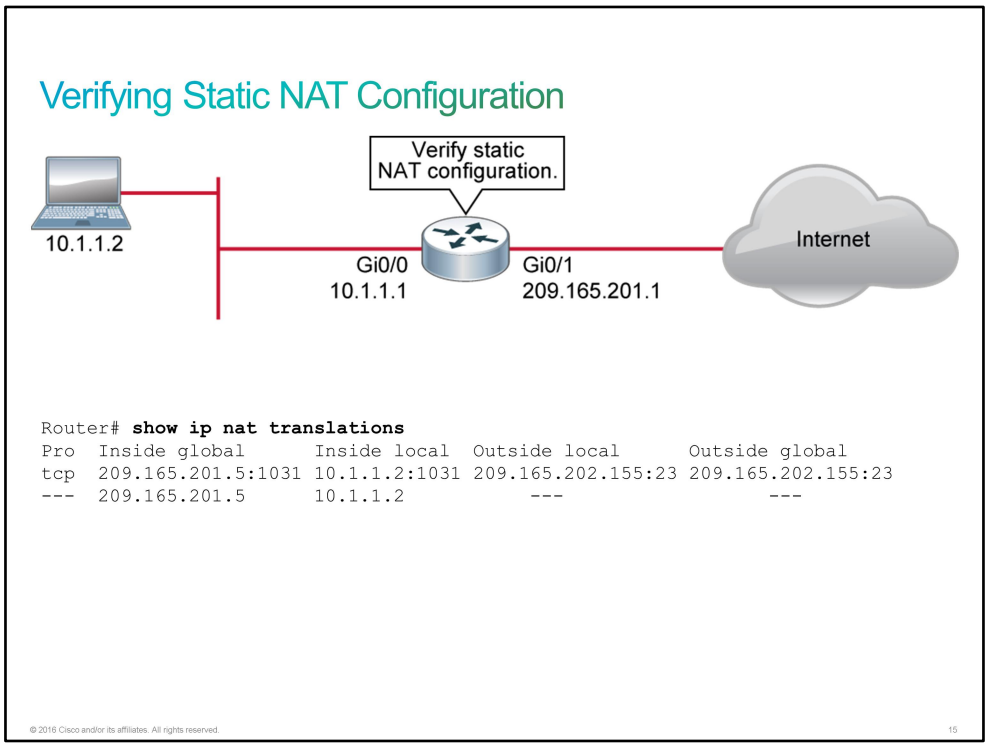
Configuring static NAT translations is a simple task. You need to define the addresses to translate and then configure NAT on the appropriate interfaces. Packets arriving on an inside interface from the identified IP address are subject to translation. Packets arriving on an outside interface that are addressed to the identified IP address are also subject to translation.

The figure shows examples of commands for the steps. You enter static translations directly into the configuration. Unlike dynamic translations, these translations are always in the NAT table.

Command	Description
interface *interface*	Specifies an interface and enters the interface configuration mode
ip address *address subnet_mask*	Sets the IP address and mask of the device
ip nat inside	Marks the interface as connected to the inside network

Command	Description
ip nat outside	Marks the interface as connected to the outside network
ip nat inside source static *inside_address outside_address*	Establishes a static translation between an inside local and inside global address

Verifying Static NAT Configuration



Command	Description
show ip nat translations	Displays active NAT translations

For more details about the **ip nat inside**, **ip nat pool**, **show ip nat translations**, and related commands, check the *Cisco IOS IP Addressing Services Command Reference* at http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr/command/ipaddr-cr-book.html.

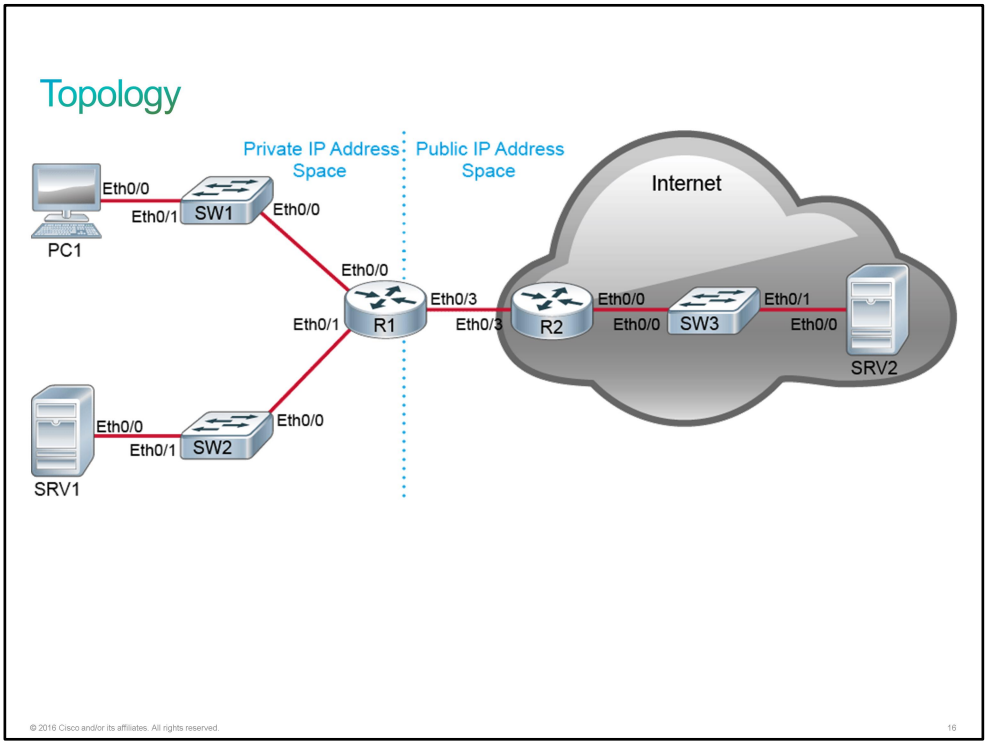
Discovery 13: Configure Static NAT

Introduction

This discovery lab will guide you through the aspects of connecting a small network to the Internet. NAT is a very important concept for Internet connectivity. The private IP addresses that are used on most internal networks are not routable on the public Internet. Because they are not routable, the private IP addresses must be translated to assigned public IP addresses at the border to the Internet.

The lab is prepared with the devices that are represented in the topology diagram. All the devices have their basic configurations in place, including hostnames and IP addresses. Router R1 receives the default route from R2 via DHCP, but NAT has not been implemented. Implementing NAT will be your job during this discovery lab. You will implement a static NAT translation for SRV1. Static NAT, which can maintain persistent IP addresses for servers, facilitates inbound connectivity.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1

Device	Characteristic	Value
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
SRV1	Hostname	SRV1
SRV1	IP address	10.10.2.20/24
SRV1	Default gateway	10.10.2.1
SRV2	Hostname	SRV2
SRV2	IP address	203.0.113.30/24
SRV2	Default gateway	203.0.113.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.2.4/24
SW2	Default gateway	10.10.2.1
SW2	Ethernet0/0 description	Link to R2
SW2	Ethernet0/1 description	Link to PC2
SW3	Hostname	SW3
SW3	VLAN 1 IP address	203.0.113.4/24
SW3	Default gateway	203.0.113.1
SW3	Ethernet0/0 description	Link to R3
SW3	Ethernet0/1 description	Link to SRV2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1

Device	Characteristic	Value
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/1 description	Link to SW2
R1	Ethernet0/1 IP address	10.10.2.1/24
R1	Ethernet0/3 description	Link to R2
R1	Ethernet0/3 IP address	198.51.100.2/24
R2	Hostname	R2
R2	Ethernet0/0 description	Link to SW3
R2	Ethernet0/0 IP address	203.0.113.1/24
R2	Ethernet0/3 description	Link to R1
R2	Ethernet0/3 IP address	198.51.100.1/24

PC and SRVs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Global IP Address Networks

Address Block	Host Starting Address	Host Ending Address	Broadcast Address	Subnet Mask
192.0.2.0/24	192.0.2.1	192.0.2.254	192.0.2.255	255.255.255.0
198.51.100.0/24	198.51.100.1	198.51.100.254	198.51.100.255	255.255.255.0
203.0.113.0/24	203.0.113.1	203.0.113.254	203.0.113.255	255.255.255.0

Task 1: Configure Static NAT

Activity

Step 1 While R1 does have access to the public IP address space, systems within the private IP address space of the topology do not. Verify this fact. Access the console of PC1 and attempt to ping SRV2. This process should fail.

On PC1, enter the following command:


```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

For a ping operation to be successful, bidirectional connectivity must exist. In this case, the problem is not getting the echo requests from PC1 to SRV2. Instead, it is a failure in getting the echo replies from SRV2 back to PC1. Because NAT has not been configured, SRV2 is receiving IP packets from the private IP address of PC1 (10.10.1.10). Routers on the Internet are not aware of the private IP address space within the networks that connect to the Internet. When SRV2 generates an echo reply to 10.10.1.10 and sends that reply to R2 for forwarding, R2 does not have a route to use to forward the reply, so the reply is dropped.

Step 2 Configure R1 interfaces for NAT. Ethernet0/0 and Ethernet0/1 are on the inside, and Ethernet0/3 is on the outside.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# int e0/0
R1(config-if)# ip nat inside
*Dec  3 20:19:13.670: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0,
changed state to up
R1(config-if)# int e0/1
R1(config-if)# ip nat inside
R1(config-if)# int e0/3
R1(config-if)# ip nat outside
```

There will be a significant pause in response to the first interface NAT command because R1 will have to initiate an internal NVI to support NAT.

Step 3 On R1 router, configure access list number 10, to identify addresses within 10.10.0.0/16 as NAT-eligible.

On R1, enter the following command:

```
R1(config)# access-list 10 permit 10.10.0.0 0.0.255.255
```

Step 4 To the R1 configuration, add a NAT statement that enables PAT. It translates the addresses that are permitted by access list 10 using the IP address that is assigned to the interface Ethernet0/3. Leave the configuration mode when you are done.

On R1, enter the following commands:

```
R1(config)# ip nat inside source list 10 interface e0/3 overload
R1(config)# end
R1#
```

Step 5 Return to the console of PC1 and reattempt the ping operation to SRV2 (203.0.113.30). This time, it should succeed.

On PC1, enter the following command:

```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/3 ms
```

Step 6 In this step and the next few steps, you will verify the status of the translation that is in place. Establish a Telnet session from PC1 to SRV2. Enter the username "admin" with the password "Cisco123."

On PC1, enter the following command:

```
PC1# telnet 203.0.113.30
Trying 203.0.113.30 ... Open


User Access Verification

Username: admin
Password: Cisco123
SRV2>
```

Step 7 You are now connected to a vty line of SRV2 from PC1. Using this interface, view the status of the IP sockets on SRV2, noting the foreign IP address that SRV2 sees.

On SRV2, enter the following command:

```
SRV2> show control-plane host open-ports
Active internet connections (servers and established)
Prot               Local Address             Foreign Address
Service    State
 tcp                        *:23                          *:0
Telnet   LISTEN
 tcp                        *:23        198.51.100.101:12959
Telnet ESTABLIS
```

SRV2 sees 198.51.100.101 as the source IP address of the connection that is coming in from PC1. This address is the IP address on Ethernet0/3 that R1 obtained via DHCP. PAT is in effect.

Step 8 Leave the connection to SRV2 from PC1 running. Access the console of SRV1 and establish a second connection to SRV2 from the private IP address space.

On SRV1, enter the following commands:

```
SRV1# telnet 203.0.113.30
Trying 203.0.113.30 ... Open

User Access Verification

Username: admin
Password: Cisco123
SRV2>
```

Step 9 Using the connection to SRV2 from SRV1, again review the IP socket status on SRV2.

On SRV2, enter the following command:


```
SRV2> show control-plane host open-ports
Active internet connections (servers and established)
Prot               Local Address             Foreign Address
Service    State
 tcp                        *:23                         *:0
Telnet   LISTEN
 tcp                        *:23        198.51.100.101:21299
Telnet ESTABLIS
 tcp                        *:23        198.51.100.101:34023
Telnet ESTABLIS
```

There are now two established Telnet sessions to SRV2. One is from PC1 and the other is from SRV1. But, from the perspective of SRV2, both connections are coming from 198.51.100.101. The two connections are uniquely identified by their source ports.

Step 10 Leaving both connections to SRV2 running, access the console of R1. Display the translation table on R1.

On R1, enter the following command:

```
R1# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 198.51.100.101:21299 10.10.1.10:21299 203.0.113.30:23    203.0.113.30:23
tcp 198.51.100.101:34023 10.10.2.20:34023 203.0.113.30:23    203.0.113.30:23
```

R1 is also using the inside source port to uniquely identify the two translation sessions.

The source ports are dynamically generated so that the ports that are shown in the example will not match those ports that you see in the lab environment. But the source ports in the R1 translation table should match those in the open-ports status for SRV2.

Using the depicted example, when R1 receives a packet from 203.0.113.30 with a source port of 23 that is destined for 198.51.100.101 and a destination port of 21299, R1 knows to translate the destination address to 10.10.1.10 and forward the packet to PC1. On the other hand, if the destination port of a similar inbound packet is 34023, R1 will translate the destination address to 10.10.2.20 and forward the packet to SRV2.

Step 11 View the running translation statistics on R1.

On R1, enter the following command:

```
R1# show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:12:47 ago
Outside interfaces:
  Ethernet0/3
Inside interfaces:
  Ethernet0/0, Ethernet0/1
Hits: 389  Misses: 0
CEF Translated packets: 389, CEF Punted packets: 0
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 10 interface Ethernet0/3 refcount 2

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

The **show ip nat statistics** command displays information on the current configuration of NAT (interface assignment, ACL specification, and so on), active translation statistics, and historic translation statistics.

Step 12 One at a time, access the consoles of PC1 and SRV1. Terminate their Telnet sessions to SRV2.

Terminate PC1 Telnet session to SRV2:

```
SRV2> exit

[Connection to 203.0.113.30 closed by foreign host]
PC1#
```

Terminate SRV1 Telnet session to SRV2:

```
SRV2> exit

[Connection to 203.0.113.30 closed by foreign host]
SRV1#
```

Step 13 At this point, you will start to migrate from a DHCP and PAT-based configuration to a configuration that uses a static IP address and NAT. On R1, set the IP address of Ethernet0/3 to 198.51.100.2/24.

On R1, enter the following command:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config-if)# int e0/3
R1(config-if)# ip address 198.51.100.2 255.255.255.0
R1(config-if)# exit
R1(config)#
```

Step 14 From the configuration, use the **do** command to execute the **show ip interface brief** command to verify the configuration on Ethernet0/3.

On R1, enter the following command:


```
R1(config)# do show ip int brief
Interface                  IP-Address      OK? Method Status
Protocol
Ethernet0/0                10.10.1.1       YES NVRAM  up                    up
Ethernet0/1                10.10.2.1       YES NVRAM  up                    up
Ethernet0/2                unassigned      YES NVRAM  administratively down
down
Ethernet0/3                198.51.100.2    YES manual up                    up
Serial1/0                  unassigned      YES NVRAM  administratively down
down
Serial1/1                  unassigned      YES NVRAM  administratively down
down
Serial1/2                  unassigned      YES NVRAM  administratively down
down
Serial1/3                  unassigned      YES NVRAM  administratively down
down
NVI0                       10.10.1.1       YES unset  up                    up
```

Step 15 Statically configure R1 to use the R2 interface as its default route.

On R1, enter the following command:

```
R1(config)# ip route 0.0.0.0 0.0.0.0 198.51.100.1
```

Step 16 Remain in the configuration mode and verify that the default route is now in the routing table.

On R1, enter the following command:

```
R1(config)# do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 198.51.100.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 198.51.100.1
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.10.1.0/24 is directly connected, Ethernet0/0
L        10.10.1.1/32 is directly connected, Ethernet0/0
C        10.10.2.0/24 is directly connected, Ethernet0/1
L        10.10.2.1/32 is directly connected, Ethernet0/1
      198.51.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        198.51.100.0/24 is directly connected, Ethernet0/3
L        198.51.100.2/32 is directly connected, Ethernet0/3
```

Step 17 Leave the inside and outside NAT configurations on the R1 interfaces, but remove the PAT configuration statement from the running configuration of R1.

On R1, enter the following command:

```
R1(config)# no ip nat inside source list 10 interface Ethernet0/3 overload
```

Step 18 Add a static NAT configuration entry that translates the SRV1 IP address (10.10.2.20) to 198.51.100.20. Then leave the configuration mode.

On R1, enter the following command:

```
R1(config)# ip nat inside source static 10.10.2.20 198.51.100.20
R1(config)# end
R1#
```

Step 19 Display the translation table on R1.

On R1, enter the following command:

```
R1# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 198.51.100.20      10.10.2.20         ---                ---
```

Static translations continuously remain in the translation table, regardless of their use.

Step 20 Access the console of SRV1 and establish a Telnet session to SRV2.

On SRV1, enter the following command:

```
SRV1# telnet 203.0.113.30
Trying 203.0.113.30 ... Open

User Access Verification

Username: admin
Password: Cisco123
SRV2>
```

Step 21 Return to the console of R1 and display the translation table while the session from SRV1 to SRV2 is open.

On R1, enter the following command:

```
R1# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 198.51.100.20:23024 10.10.2.20:23024  203.0.113.30:23    203.0.113.30:23
--- 198.51.100.20      10.10.2.20         ---                ---
```

This example shows two entries in the translation table.

The first entry is an extended entry because it embodies more details than just an IP address mapping to an IP address. In this case, it specifies the protocol (TCP) and also the ports in use on both systems.

The second entry is a simple entry. It simply maps one IP address to another.

The extended entry is due to the use of the static translation for the Telnet session from SRV1 to SRV2. It details the characteristics of that session.

The simple entry is the persistent entry that is associated with the configured static translation.

Step 22 The most common use for static NAT translations is to provide a persistent IP address that the systems in the public IP address space can use to communicate with specific systems in the private IP address space. Demonstrate this function. Access the console of SRV2 and establish a Telnet connection back to SRV1.

On SRV2, enter the following command:

```
SRV2# telnet 198.51.100.20
Trying 198.51.100.20 ... Open


User Access Verification

Username: admin
Password: Cisco123
SRV1>
```

Step 23 With the two Telnet connections running, return to the console of R1 and view the translation table.

On R1, enter the following command:

```
R1# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 198.51.100.20:23   10.10.2.20:23      203.0.113.30:46401 203.0.113.30:46401
tcp 198.51.100.20:29158 10.10.2.20:29158  203.0.113.30:23    203.0.113.30:23
--- 198.51.100.20      10.10.2.20         ---                ---
```

There is the one simple entry that is associated with the configured static translation, and two extended entries, each associated with an active session.

Step 24 Access the console of SRV2 and terminate the Telnet session to SRV1.

On SRV2, enter the following command:

```
SRV1> exit

[Connection to 198.51.100.20 closed by foreign host]
SRV2#
```

Step 25 Access the console of SRV1 and terminate the Telnet session to SRV2.

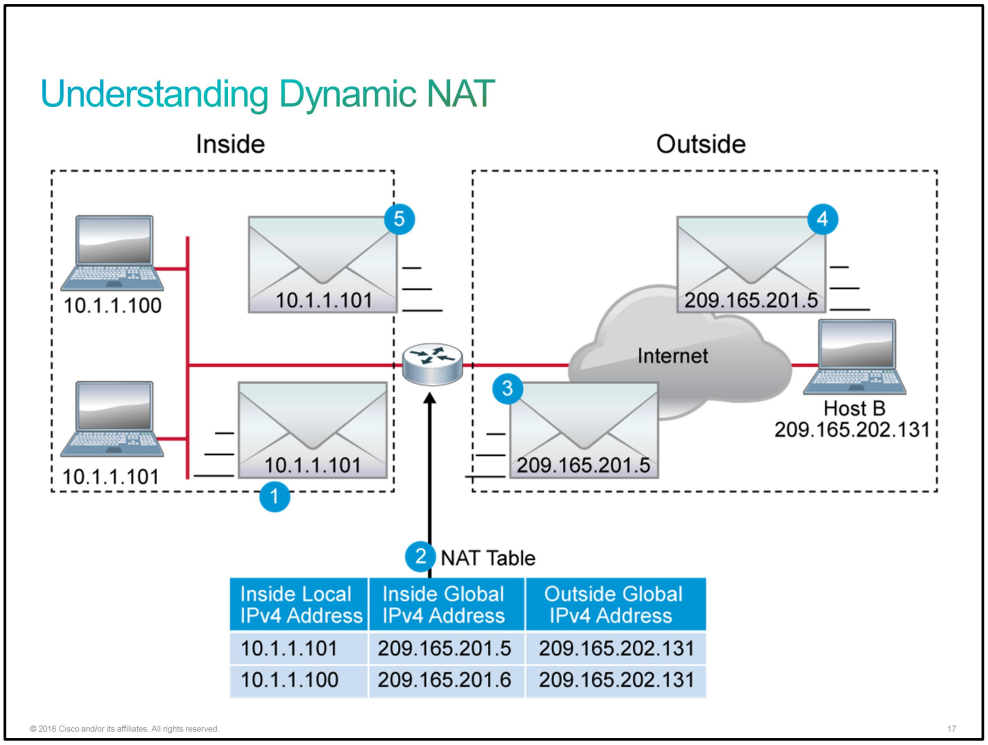
On SRV1, enter the following command:

```
SRV2> exit

[Connection to 203.0.113.30 closed by foreign host]
SRV1#
```

This is the end of the discovery lab.

Understanding Dynamic NAT

While static NAT provides a permanent mapping between an internal address and a specific public address, dynamic NAT maps private IP addresses to public addresses. These public IP addresses come from a NAT pool. Dynamic NAT configuration differs from static NAT, but it also has some similarities. Like static NAT, it requires the configuration to identify each interface as an inside or outside interface. However, rather than creating a static map to a single IP address, a pool of inside global addresses is used.



The figure illustrates a router that is translating a source address inside a network into a source address outside the network. The following are the steps for translating an inside source address:

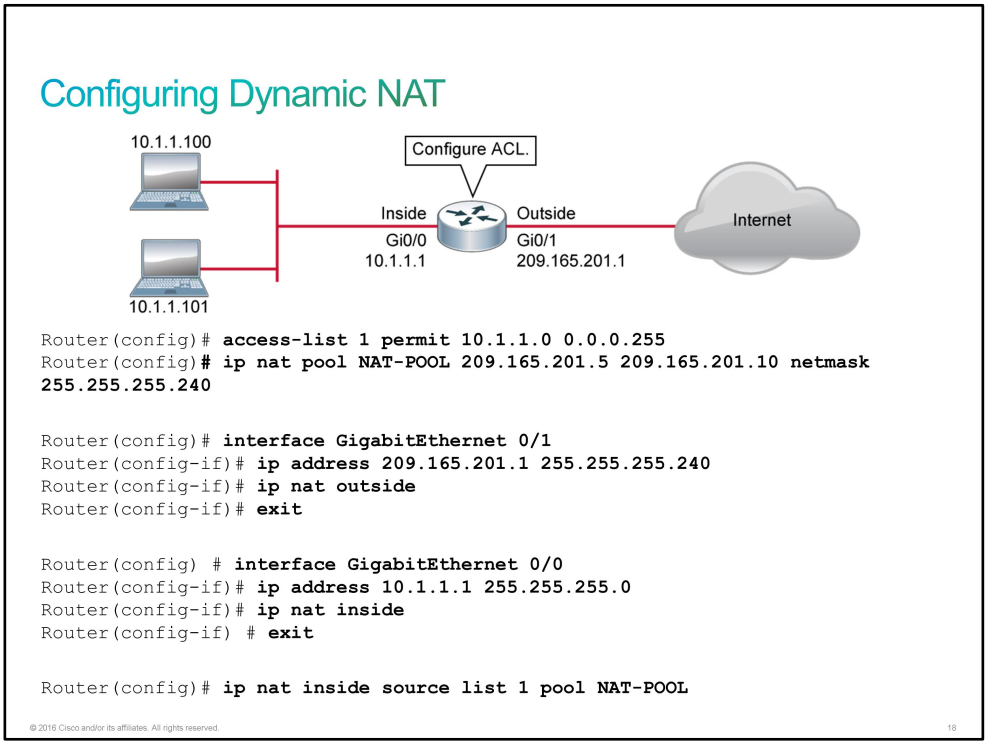
1. The users at hosts 10.1.1.100 and 10.1.1.101 want to open a connection to Host B (IP address 209.165.202.131).
2. The first packet that the router receives from host 10.1.1.101 causes the router to check its NAT table. If no static translation entry exists, the router determines that the source address 10.1.1.101 must be translated dynamically. The router then selects a legal global address from the dynamic address pool and creates a translation entry (in this example, 209.165.201.5). This type of entry is called a *simple entry*. For the second host, 10.1.1.100, the router selects a legal global address from the dynamic address pool and creates a second translation entry (in this example, 209.165.201.6).
3. The router replaces the inside local source address of host 10.1.1.101 with the translated inside global address and forwards the packet.
4. Host B receives the packet and responds to host 209.165.201.5, using the inside global IPv4 destination address 209.165.201.5. When Host B receives the second packet, it responds to host 209.165.201.6, using the inside global IPv4 destination address 209.165.201.6.

5. When the router receives the packet with the inside global IPv4 address of 209.165.201.5, the router performs a NAT table lookup using the inside global address as a key. The router then translates the address back to the inside local address of host 10.1.1.101 and forwards the packet to host 10.1.1.101. When the router receives the packet with the inside global IPv4 address of 209.165.201.6, the router performs a NAT table lookup using the inside global address as a key. The router then translates the address back to the inside local address of host 10.1.1.100 and forwards the packet to host 10.1.1.100.
6. Hosts 10.1.1.100 and 10.1.1.101 receive the packets and continue the conversation. The router performs Steps 2 through 5 for each packet.

Configuring and Verifying Dynamic NAT

Configuring Dynamic NAT

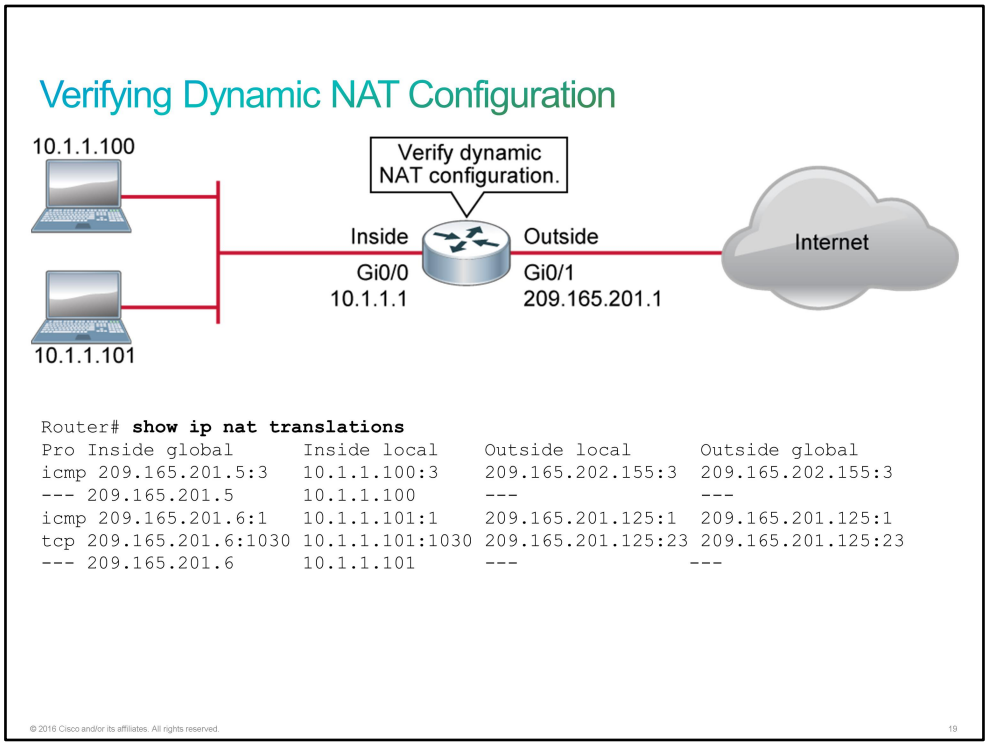
Command	Description
interface *interface*	Specifies an interface and enters the interface configuration mode
ip nat pool *pool_name start_ip end_ip* **netmask** *netmask*	Defines an IP address pool
ip nat inside source list *acl_number* **pool** *pool_name*	Establishes a dynamic source translation by specifying the ACL and the address pool
ip address *address subnet_mask*	Sets the IP address and mask
ip nat inside	Marks the interface as connected to the inside network
ip nat outside	Marks the interface as connected to the outside network
access-list *acl_number* **permit** *ip_address netmask*	Creates an access list that defines the inside local addresses that are eligible to be translated



Note	The ACL must permit only those addresses that need to be translated. Remember that there is an implicit **deny any** statement at the end of each ACL. An ACL that is too permissive can lead to unpredictable results. Using **permit any** can result in NAT consuming too much router resources, which can cause network problems.

Verifying Dynamic NAT Configuration

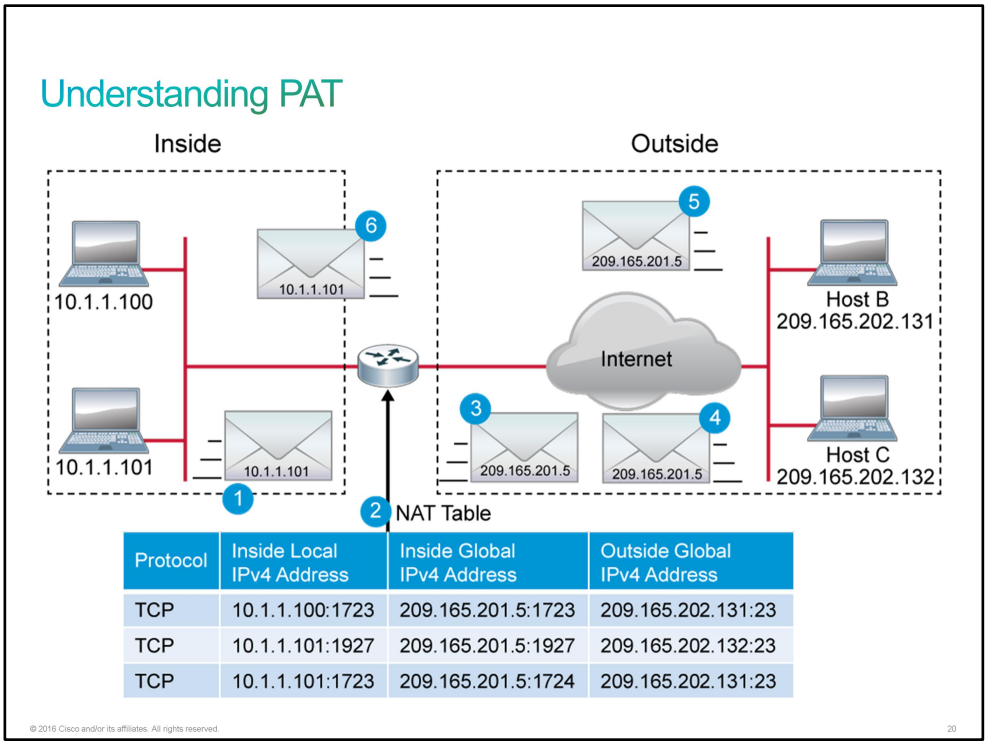
Command	Description
show ip nat translations	Displays active NAT translations



For more details about the **ip nat inside**, **ip nat pool**, **show ip nat translations** and related commands, check the *Cisco IOS IP Addressing Services Command Reference* at http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr/command/ipaddr-cr-book.html.

Understanding PAT

One of the main forms of NAT is PAT, which is also referred to as *overload* in Cisco IOS configuration. Several inside local addresses can be translated using NAT into just one or a few inside global addresses by using PAT. Most home routers operate in this manner. Your ISP assigns one address to your router, yet several members of your family can simultaneously surf the Internet.



With NAT overload, multiple addresses can be mapped to one or a few addresses because a TCP or UDP port number tracks each private address. When a client opens a TCP/IP session, the NAT router assigns a port number to its source address. NAT overload ensures that clients use a different TCP or UDP port number for each client 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 the client to which the router routes the packets. It also validates that the incoming packets were requested, which adds a degree of security to the session.

- PAT uses unique source port numbers on the inside global IPv4 address to distinguish between translations. Because the port number is encoded in 16 bits, the total number of internal addresses that NAT can translate into one external address is, theoretically, as many as 65,536.
- PAT attempts to preserve the original source port. If the source port is already allocated, PAT attempts to find the first available port number. It starts from the beginning of the appropriate port group, 0 to 511, 512 to 1023, or 1024 to 65535. If PAT does not find an available port from the appropriate port group and if more than one external IPv4 address is configured, PAT moves to the next IPv4 address and tries to allocate the original source port again. PAT continues trying to allocate the original source port until it runs out of available ports and external IPv4 addresses.

NAT generally translates IP addresses only as a 1:1 correspondence between publicly exposed IP addresses and privately held IP addresses. NAT overload modifies the private IP address and potentially the port number of the sender. NAT overload chooses the port numbers that hosts see on the public network.

NAT routes incoming packets to their inside destination by referring to the incoming destination IP address given by the host on the public network. With NAT overload, there is generally only one publicly exposed IP address (or a very few). Incoming packets from the public network are routed to their destinations on the private network by referring to a table in the NAT overload device that tracks public and private port pairs. This mechanism is called *connection tracking*.

The figure illustrates a PAT operation when one inside global address represents multiple inside local addresses. The TCP port numbers act as differentiators. Both Hosts B and C think that they are talking to a single host at the address 209.165.201.5. They are actually talking to different hosts, and the port number is the differentiator. In fact, many inside hosts could share the inside global IPv4 address by using many port numbers.

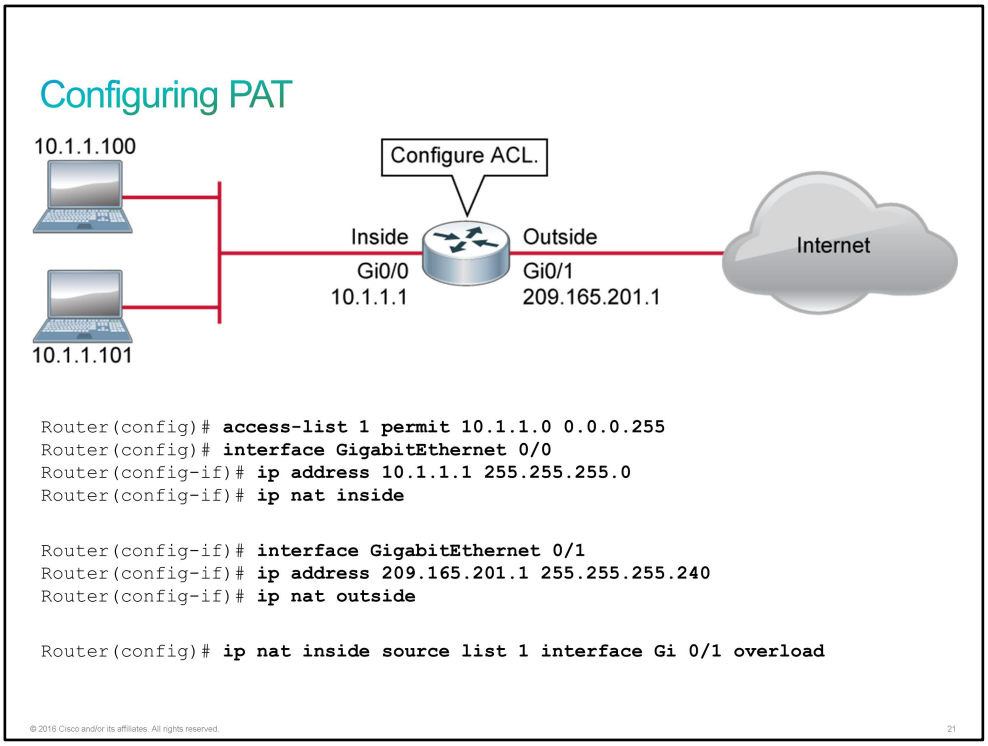
The router performs this process when it overloads inside global addresses:

1. The user at host 10.1.1.100 opens a connection to Host B. A second user at host 10.1.1.101 opens a connection to Hosts B and C.
2. The first packet that the router receives from host 10.1.1.100 causes the router to check its NAT table. If no translation entry exists, the router determines that address 10.1.1.100 must be translated and sets up a translation of the inside local address 10.1.1.100 into an inside global address. If overloading is enabled and another translation is active, the router reuses the inside global address from that translation and saves enough information to be able to translate back. This type of entry is called an extended entry.
3. The router replaces the inside local source address 10.1.1.100 with the selected inside global address 209.165.201.5 and forwards the packet.
4. Host B receives the packet and responds to host 10.1.1.100, using the inside global IPv4 address 209.165.201.5. Host C receives a packet with the same inside global address, even though the packet originated from host 10.1.1.101.
5. When the router receives the packet with the inside global IPv4 address, the router performs a NAT table lookup. Using the inside global address and port and outside global address and port as a key, the router translates the address back into the correct inside local address, 10.1.1.100, and forwards the packet to host 10.1.1.100.
6. Host 10.1.1.100 receives the packet and continues the conversation. The router performs Steps 2 through 5 for each packet.

Configuring and Verifying PAT

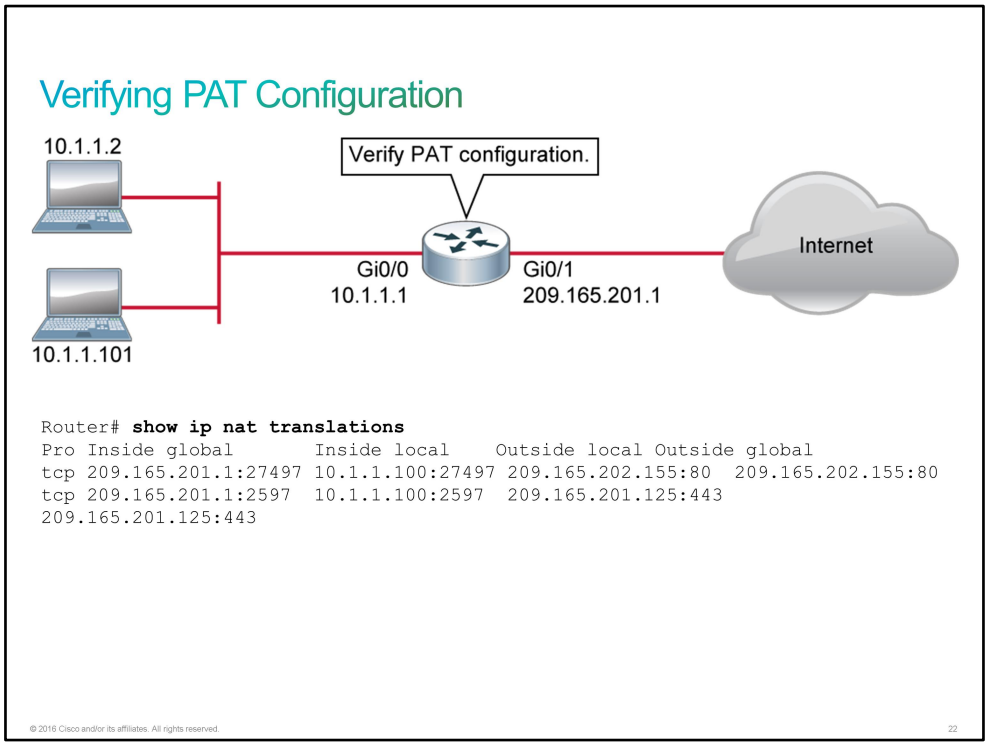
Configuring PAT

Command	Description
interface *interface*	Specifies an interface and enters the interface configuration mode
ip address *address subnet_mask*	Sets the IP address and mask
ip nat inside	Marks the interface as connected to the inside network
ip nat outside	Marks the interface as connected to the outside network
ip nat inside source list *access-list-number* **interface** *interface* **overload**	Establishes dynamic source translation, specifying the ACL
access-list *acl_number* **permit** *ip_address netmask*	Creates an ACL that defines the inside local addresses that are eligible to be translated



Verifying PAT Configuration

Command	Description
show ip nat translations	Displays active NAT translations



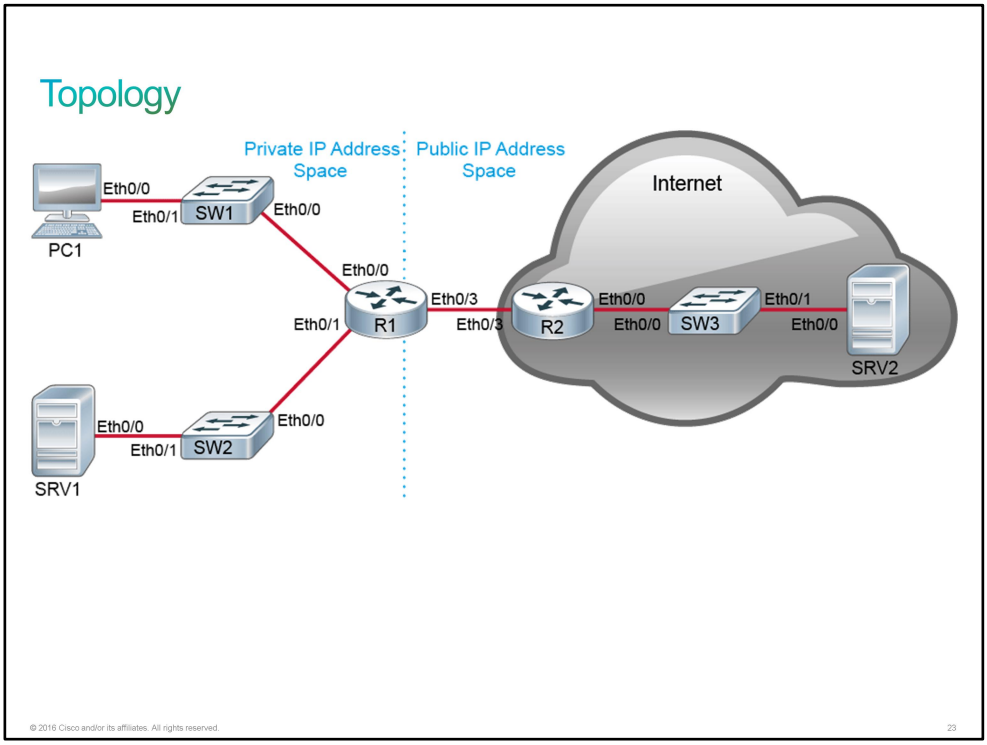
For more details about the **ip nat inside**, **ip nat pool**, and **show ip nat translations** and related commands, check the *Cisco IOS IP Addressing Services Command Reference* at http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr/command/ipaddr-cr-book.html.

Discovery 14: Configure Dynamic NAT and PAT

Introduction

In this discover lab, you will implement a dynamic NAT pool that other systems on the internal network can share for outbound connectivity.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
SRV1	Hostname	SRV1
SRV1	IP address	10.10.2.20/24

Device	Characteristic	Value
SRV1	Default gateway	10.10.2.1
SRV2	Hostname	SRV2
SRV2	IP address	203.0.113.30/24
SRV2	Default gateway	203.0.113.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.2.4/24
SW2	Default gateway	10.10.2.1
SW2	Ethernet0/0 description	Link to R2
SW2	Ethernet0/1 description	Link to PC2
SW3	Hostname	SW3
SW3	VLAN 1 IP address	203.0.113.4/24
SW3	Default gateway	203.0.113.1
SW3	Ethernet0/0 description	Link to R3
SW3	Ethernet0/1 description	Link to SRV2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/1 description	Link to SW2
R1	Ethernet0/1 IP address	10.10.2.1/24
R1	Ethernet0/3 description	Link to R2

Device	Characteristic	Value
R1	Ethernet0/3 IP address	198.51.100.2/24
R2	Hostname	R2
R2	Ethernet0/0 description	Link to SW3
R2	Ethernet0/0 IP address	203.0.113.1/24
R2	Ethernet0/3 description	Link to R1
R2	Ethernet0/3 IP address	198.51.100.1/24

PC and SRVs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Global IP Address Networks

Address Block	Host Starting Address	Host Ending Address	Broadcast Address	Subnet Mask
192.0.2.0/24	192.0.2.1	192.0.2.254	192.0.2.255	255.255.255.0
198.51.100.0/24	198.51.100.1	198.51.100.254	198.51.100.255	255.255.255.0
203.0.113.0/24	203.0.113.1	203.0.113.254	203.0.113.255	255.255.255.0

Task 1: Configure Dynamic NAT and PAT

Activity

Step 1 The static NAT configuration is in place. It is now time to explore dynamic NAT translation by using a pool of IP addresses. First, access the console of PC1 and verify that it cannot ping SRV2.

On PC1, enter the following command:

```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The failure of the ping process is not a failure on the delivery of the echo requests from PC1 to SRV2. It is a failure on the return of the echo replies from SRV2 to PC1. Because there is no translation that is configured for PC1, SRV2 receives the echo requests that come from 10.10.1.10. When SRV2 tries to reply to 10.10.1.10 from within the public IP address space, R2 does not have a route that it can to use to get to the private IP address space.

Step 2 On R1, define a pool of NAT addresses named "NatPool" by specifying the address range from 198.51.100.100 to 198.51.100.149.

On R1, enter the following command:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip nat pool NatPool 198.51.100.100 198.51.100.149 netmask
255.255.255.0
```

Step 3 Verify that access list 10 is still in place, permitting addresses from 10.10.0.0/16.

On R1, enter the following command:

```
R1(config)# do show access-list 10
Standard IP access list 10
    10 permit 10.10.0.0, wildcard bits 0.0.255.255
```

On R1, verify which interfaces are NAT inside and NAT outside.

```
R1(config)# do show ip nat statistics
Total active translations: 1 (1 static, 0 dynamic; 0 extended)
Peak translations: 1, occurred 00:04:10 ago
Outside interfaces:
  Ethernet0/3
Inside interfaces:
  Ethernet0/0, Ethernet0/1
Hits: 0  Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

Step 4 Define a dynamic translation rule that specifies access list 10 as the source and that uses addresses from the pool NatPool. Then leave the configuration mode.

On R1, enter the following commands:

```
R1(config)# ip nat inside source list 10 pool NatPool
R1(config)# end
R1#
```

Step 5 Verify that there is now bidirectional connectivity between PC1 and SRV2. Access the console of PC1 and send a ping to SRV2.

On PC1, enter the following command:

```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Step 6 Access the console of R1 and view the current translation table.

On R1, enter the following command:

```
R1# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 198.51.100.100:4  10.10.1.10:4       203.0.113.30:4     203.0.113.30:4
--- 198.51.100.100     10.10.1.10         ---                ---
--- 198.51.100.20      10.10.2.20         ---                ---
```

If you proceeded quickly enough, three translations will be in the table.

The extended translation that is associated with the ICMP session is short-lived and may have timed out. If it did, you can re-send the ping from PC1 and display the translation table again.

There is a simple entry in the table that is associated with the assignment of an address from the pool to PC1. By default, dynamic translations that are assigned from a NAT pool have a 24-hour inactivity timeout. So the translation for PC1 to 198.51.100.100 will persist as long as it is used at least once per day.

The third entry that is translating 10.10.2.20 to 198.51.100.20 is the static entry.

Step 7 One at a time, access the consoles of PC1, SW1, and SW2. From PC1, execute a Telnet session to SRV2. From SW1 and SW2, send pings to SRV2.

On PC1, enter the following command:

```
PC1# telnet 203.0.113.30
Trying 203.0.113.30 ... Open


User Access Verification

Username: admin
Password: Cisco123
SRV2>
```

On SW1, enter the following command:

```
SW1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/202/1008 ms
```

On SW2, enter the following command:


```
SW2# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/201/1003 ms
```

Step 8 Return to the console of R1 and view the translation table.

On R1, enter the following command:

```
R1# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 198.51.100.101:0  10.10.1.4:0        203.0.113.30:0     203.0.113.30:0
--- 198.51.100.101     10.10.1.4          ---                ---
tcp 198.51.100.100:24829 10.10.1.10:24829 203.0.113.30:23    203.0.113.30:23
--- 198.51.100.100     10.10.1.10         ---                ---
icmp 198.51.100.102:0  10.10.2.4:0        203.0.113.30:0     203.0.113.30:0
--- 198.51.100.102     10.10.2.4          ---                ---
--- 198.51.100.20      10.10.2.20         ---                ---
```

The extended ICMP entries that are associated with the ping activity are short-lived. You can always try to resend the ping and re-display the translation table.

SW1 (10.10.1.4) and SW2 (10.10.2.4) have been assigned IP addresses from the NAT pool. Again, there is a 24-hour inactivity timeout on these dynamic entries by default.

Step 9 Display the translation statistics on R1.

On R1, enter the following command:

```
R1# sh ip nat statistics
Total active translations: 5 (1 static, 4 dynamic; 1 extended)
Peak translations: 7, occurred 00:03:10 ago
Outside interfaces:
  Ethernet0/3
Inside interfaces:
  Ethernet0/0, Ethernet0/1
Hits: 112  Misses: 0
CEF Translated packets: 112, CEF Punted packets: 0
Expired translations: 7
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 10 pool NatPool refcount 4
 pool NatPool: netmask 255.255.255.0
        start 198.51.100.100 end 198.51.100.149
        type generic, total addresses 50, allocated 3 (6%), misses 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

The statistics that are displayed in the lab environment will likely differ from the example. But, in any case, statistics include the current status such as the current active translation count, historical statistics such as the largest number of translations seen on R1, and configuration information such as the details of the NAT pools.

Step 10 Return to the console of PC1 and terminate the Telnet session to SRV2.

On PC1, enter the following command:

```
SRV2> exit

[Connection to 203.0.113.30 closed by foreign host]
PC1#
```

Step 11 Return to the console of R1 and clear all dynamic translations from the translation table.

On R1, enter the following command:

```
R1# clear ip nat translation *
```

Step 12 Display the translation table, verifying the removal of the dynamic entries.

On R1, enter the following command:

```
R1# show ip nat translation
Pro Inside global      Inside local       Outside local      Outside global
--- 198.51.100.20      10.10.2.20         ---                ---
```

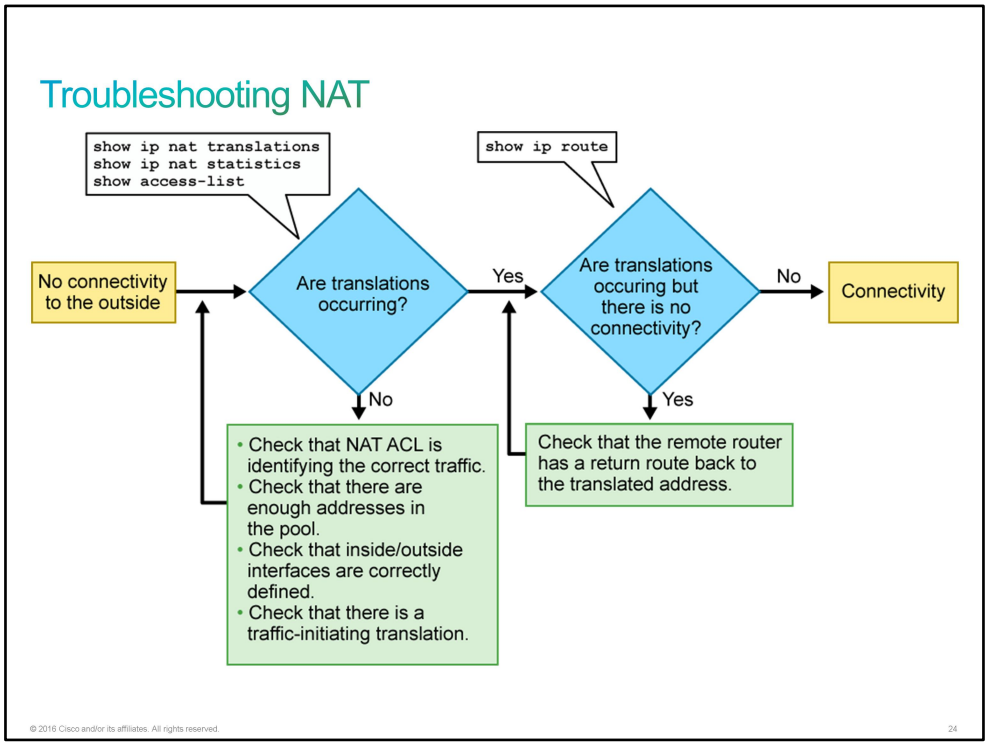
The dynamic entries have been removed, but the statically configured entry for SRV1 remains.

Feel free to continue with independent exploration of NAT concepts within the lab environment.

This is the end of the discovery lab.

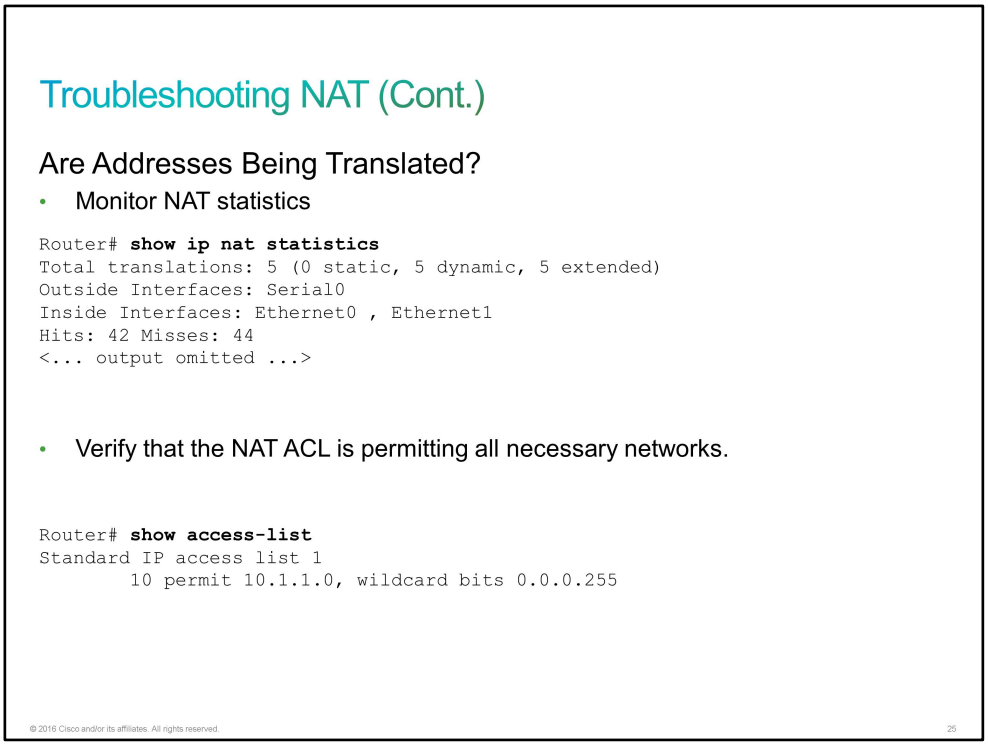
Troubleshooting NAT

When you have IPv4 connectivity problems in a NAT environment, it is often difficult to determine the cause of the problem. NAT is often blamed, when in reality there is an underlying problem. When you are trying to determine the cause of an IPv4 connectivity problem, it helps to eliminate NAT as the problem.



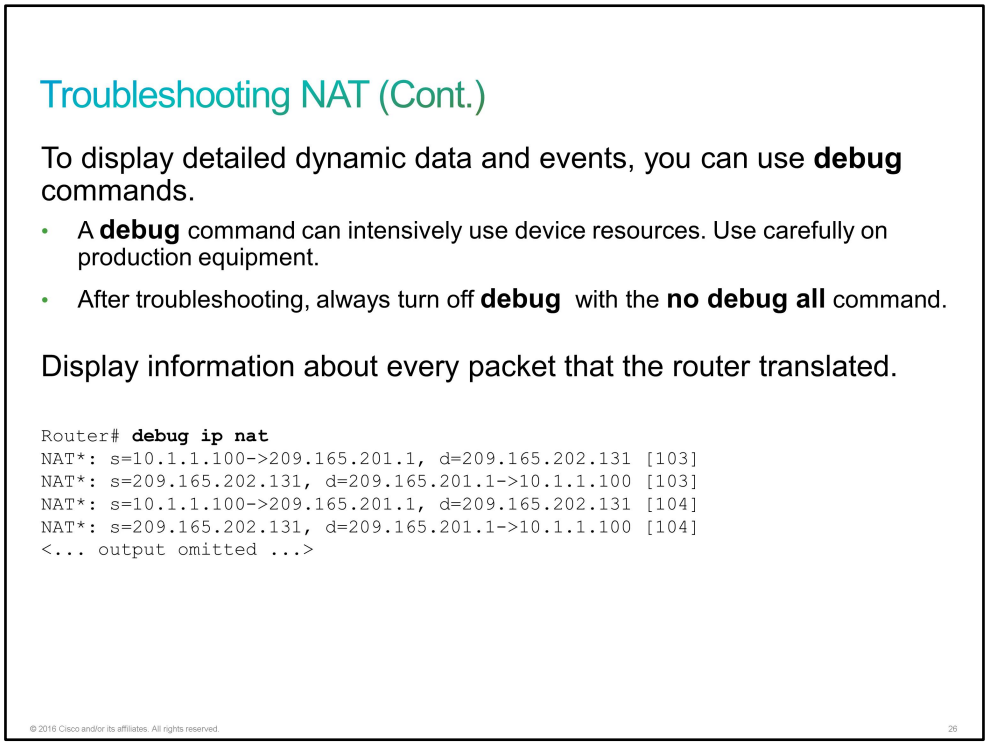
Follow these steps to verify that NAT is operating as expected:

1. Verify that translations are occurring:
   - Use the **show ip nat translations** command to determine if translations exist in the translation table.
   - Verify that the translation is actually occurring by using the **show ip nat statistics** and **debug ip nat** commands.
   - Use the **show access-list** command to verify that the ACL that is associated with the NAT command is permitting all necessary networks.
   - Use the **show ip nat statistics** command to verify that the router interfaces are appropriately defined as NAT inside or NAT outside.
   - If some devices have connectivity but others do not, the NAT pool might be out of addresses.
2. If translations are occurring but there is no connectivity, use the **show ip route** command to verify that there is a return route to the translated address.



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 numbers that have been allocated. However, in a more complex NAT environment, with several translations taking place, this **show** command may not clearly identify the issue. It may be necessary to run **debug** commands on the router.

Note	You can use the **clear ip nat translation *** command to clear all dynamic address translation entries. By default, translation entries time out after 24 hours. When testing the NAT configuration, it can be useful to clear translations.



Note	The **debug** command, especially the **debug all** command, should be used sparingly. These commands can disrupt router operations. The **debug** commands are useful when configuring or troubleshooting a network. However, they can make intensive use of CPU and memory resources. It is recommended that you run as few debug processes as necessary and disable them immediately when they are no longer needed. On production networks, you should use the **debug** commands with caution because they can affect the performance of the device.

The **debug ip nat** command displays information about every packet that the router translates, which helps you to verify NAT operation. The **debug ip nat detailed** command generates a description of each packet that is 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 you need to troubleshoot the NAT problem. Always remember to turn off debugging when finished.

The example shows a sample **debug ip nat** output. In the output, you can see that the inside host 10.1.1.100 initiated traffic to the outside host 209.165.202.131 and has been translated to the address 209.165.201.1.

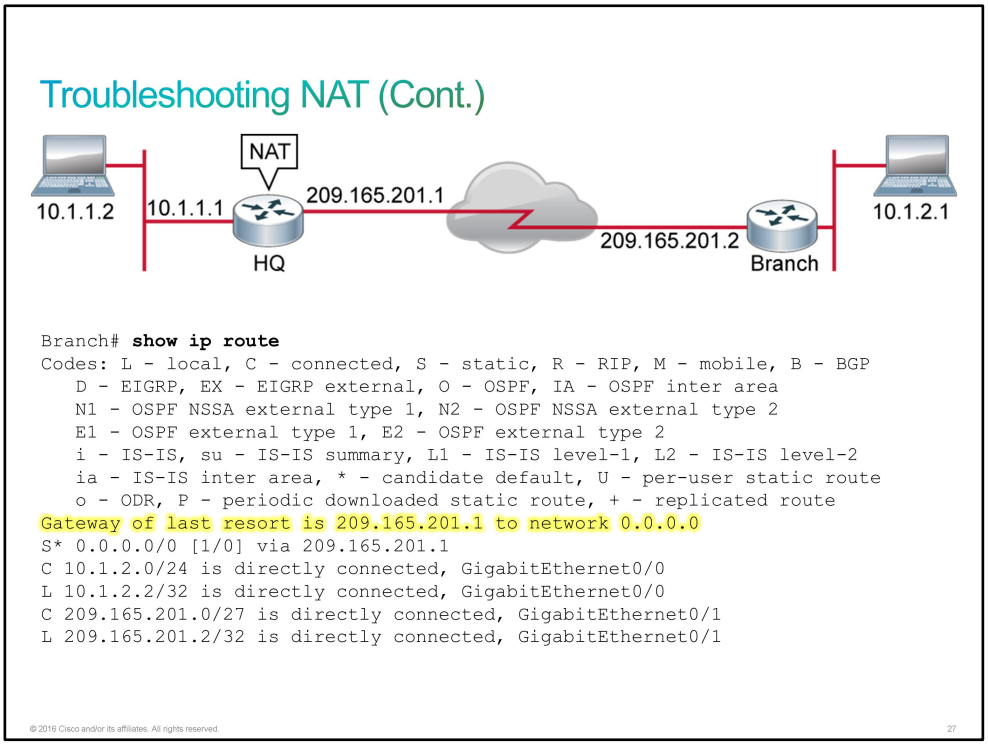
For decoding the **debug** output, note what the following symbols and values indicate:

- *****: 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=**: Refers **t**o the source IP address.

- **a.b.c.d->w.x.y.z**: Indicates that source address a.b.c.d is translated to w.x.y.z.
- **d=**: Refers to the destination IP address.
- **[xxxx]**: The value in brackets is the IP identification number. This information may be useful for debugging because it enables correlation with other packet traces from protocol analyzers.

Finally, you should make sure that the ACL that the NAT command references is permitting all the necessary networks. Notice that ACLs use wildcard masks and not subnet masks.

If translations are occurring, but there is no connectivity, verify that the remote router has a route to the translated address.



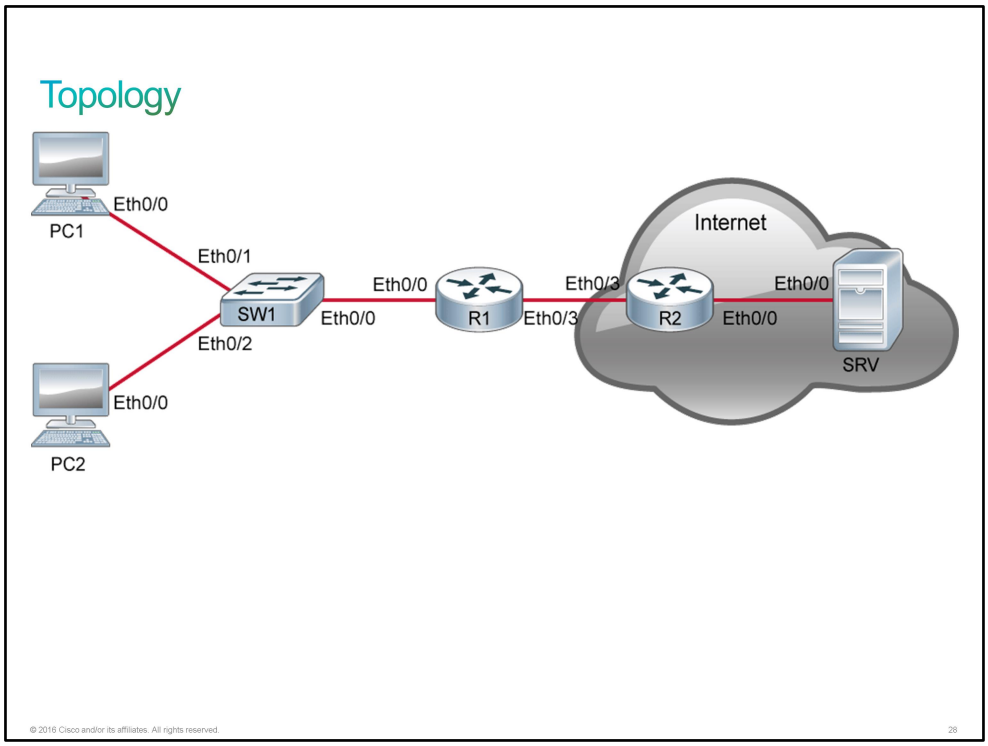
If translations are occurring but a ping to the remote network still fails, the issue might be a missing route back to the translated address. This problem can arise in NAT between a headquarters and branch office. It is usually not an issue when connecting to an ISP, because the service provider takes care of routing all the necessary traffic back to the customer.

Discovery 15: Troubleshoot NAT

Introduction

In this discovery lab, you will use different **show** commands to troubleshoot common NAT-related issues.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1

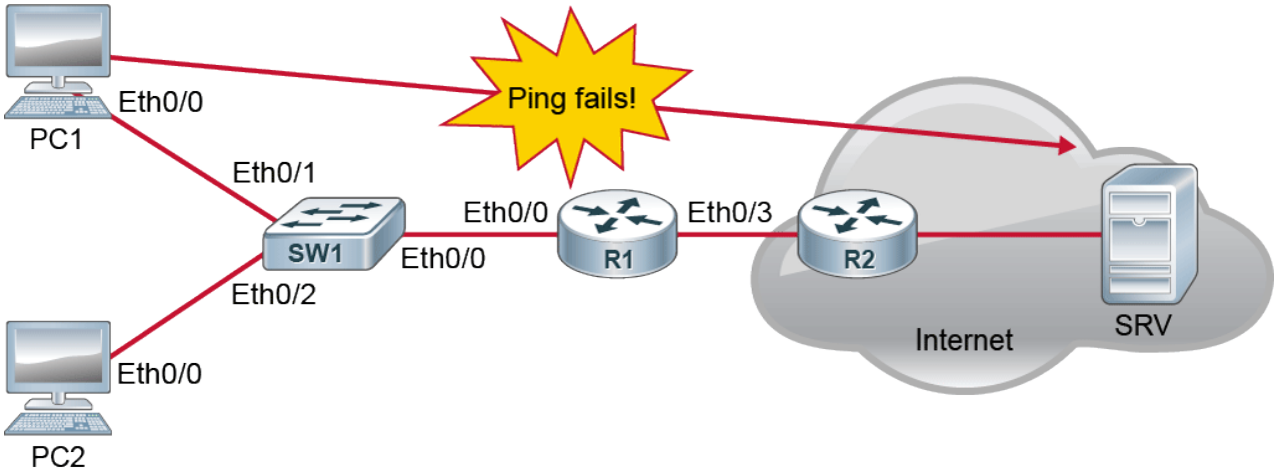
Device	Characteristic	Value
SRV	Hostname	SRV
SRV	IP address	203.0.113.30/24
SRV	Default gateway	203.0.113.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.4/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to R1
SW1	Ethernet0/1 description	Link to PC1
SW1	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW1
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Ethernet0/3 description	Link to R2
R1	Ethernet0/3 IP address	198.51.100.2/24
R2	Hostname	R2
R2	Ethernet0/0 description	Link to SRV
R2	Ethernet0/0 IP address	203.0.113.1/24
R2	Ethernet0/3 description	Link to R1
R2	Ethernet0/3 IP address	198.51.100.1/24

PC and SRV in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Troubleshoot NAT

Activity

Step 1 PC1 and SRV are unable to ping after a new NAT configuration is put in place.



The figure shows that PC1 (10.10.1.10) cannot ping SRV (203.0.113.30). R1 router has a default gateway set to 198.51.100.1.

Ping from PC1 to SRV will fail.

```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Verify relevant part of configuration on the R1 router.

```
R1# show running-config
<... output omitted ...>
interface Ethernet0/0
 description Link to SW1
 ip address 10.10.1.1 255.255.255.0
 ip nat outside
<... output omitted ...>
!
interface Ethernet0/3
 description Link to R2
 ip address 198.51.100.2 255.255.255.0
 ip nat inside
!
ip nat inside source list 20 interface Ethernet0/3 overload
ip route 0.0.0.0 0.0.0.0 198.51.100.1
!
access-list 20 permit 0.0.0.0 255.255.255.0
<... output omitted ...>
```

Step 2 To troubleshoot the problem, use the **show ip nat translation** command to see if any translations are currently in the table.

On R1, enter the following command:

```
R1# show ip nat translations
R1#
```

Translations are not occurring.

Step 3 Next, you must determine whether any translations have ever taken place and identify the interfaces between which translation should be occurring. You use the **show ip nat statistics** command.

On R1, enter the following command:

```
R1# show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Peak translations: 0
Outside interfaces:
  Ethernet0/0
Inside interfaces:
  Ethernet0/3
Hits: 0  Misses: 0
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 20 interface Ethernet0/3 refcount 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

Step 4 The NAT counters are at 0, verifying that no translation has occurred. The R1 router interfaces are incorrectly defined as NAT inside and NAT outside. Fix the R1 router configuration.

On R1, enter the following command:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# inter Eth0/0
R1(config-if)# no ip nat outside
R1(config-if)# ip nat inside
R1(config-if)# inter Eth0/3
R1(config-if)# no ip nat inside
R1(config-if)# ip nat outside
```

Verify connectivity between PC1 and SRV. Ping from PC1 to SRV will fail again.

```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Step 5 Verify that the access list is correct.

On R1, enter the following command:

```
R1# show access-list
Standard IP access list 20
    10 permit 0.0.0.0, wildcard bits 255.255.255.0
```

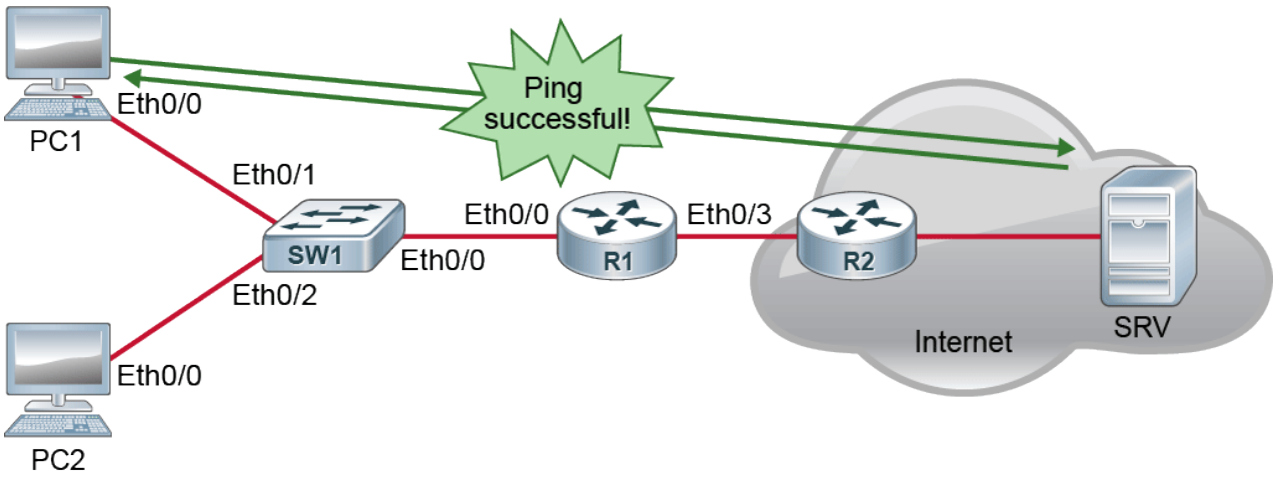
Access list has wrong wildcard mask. Wildcard mask is matching on the fourth octet only. You will need to invert wildcard mask and define the correct network part of the access list.

Step 6 On the R1 router fix access list.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# no access-list 20
R1(config)# access-list 20 permit 10.10.1.0 0.0.0.255
```

Step 7 After you have corrected the wildcard mask, you generate another ping from PC1 to SRV. The connectivity test is now a success. Verify that translations are occurring and you have connectivity to the remote network.



On PC1, enter the following command:

```
PC1# ping 203.0.113.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

On R1, enter the following command:

```
R1# sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
icmp 198.51.100.2:12   10.10.1.10:12      203.0.113.30:12    203.0.113.30:12
```

This is the end of the discovery lab.

Challenge

1. Which of the following is the customer side of the demarcation point?

   A. CPE
   B. CTE

2. The following are methods that are used to connect small offices to the internet. Which of them would you use if you had an environment filled with EMI and RFI?

   A. Copper
   B. Optic Fiber
   C. Wifi

3. **show ip nat translations** command displays which interfaces are enabled for NAT configuration on a router. True or False?

   A. True
   B. False

4. Which of the following is eliminated with the use of NAT? (Choose two.)

   A. Need to readdress all hosts that require external access
   B. IP Address conservation
   C. Revealing of private addresses outside of the network
   D. Performance problems and switching delays

5. Which of the following is the IPv4 address of an outside host as it appears to the inside network?

   A. Inside local address
   B. Inside global address
   C. Outside global address
   D. Outside local address

6. What is the difference between static NAT and dynamic NAT?

   A. Static NAT maps one-to-one and dynamic NAT maps one-to-many
   B. Static NAT maps one-to-many and dynamic NAT maps many-to-one
   C. Static NAT maps one-to-one and dynamic NAT maps many-to-many

7. Which Translation technology would most likely be used at home, especially for connecting devices such as tablets, phones, and PC's through the DSL internet connection?

   A. static NAT
   B. dynamic NAT
   C. PAT

Answer Key

Challenge

1. A
2. B
3. B
4. A, C
5. D
6. C
7. C

Module 5: Network Device Management and Security

Introduction

This module describes the steps that are required to secure local and remote access to network devices. It discusses general recommendations on how to improve device hardening. It describes how to configure syslog and how to safely run debugs on Cisco IOS. This module also describes the different stages of the router bootup process, Cisco IOS File System, and how to manage Cisco IOS images or configuration files. The universality of Cisco IOS images and the idea behind licensing are explained, and students are also shown how to verify the current license and install a new one.

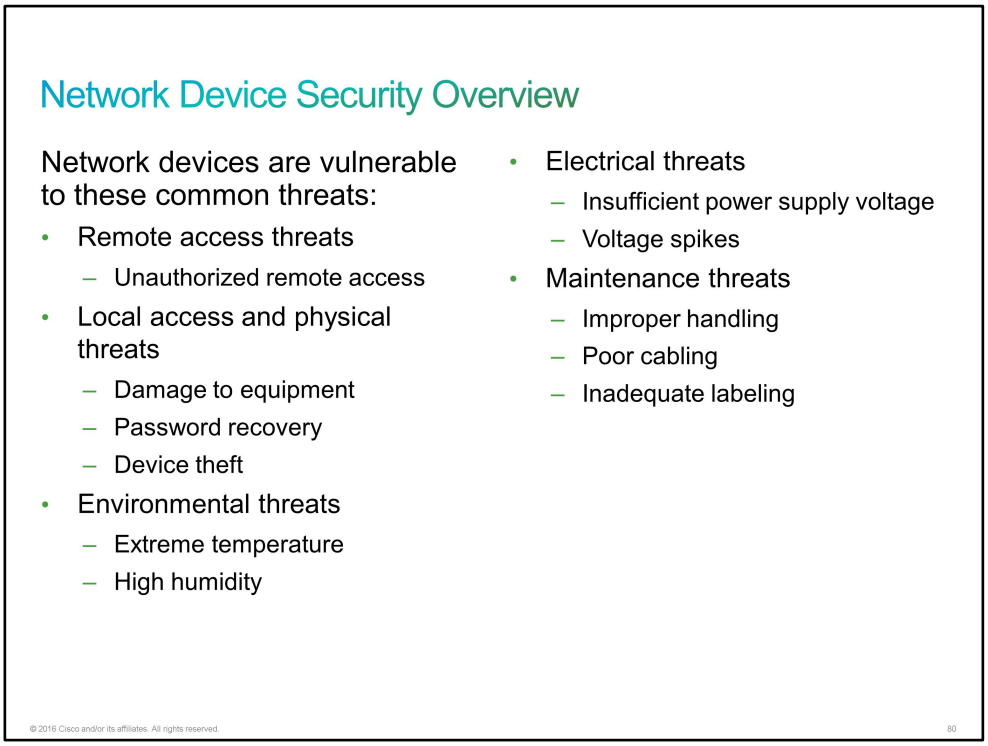
Lesson 1: Securing Administrative Access

Introduction

Your boss sends you to your customer to verify potential security threats. On the customer network devices, you will secure access to a privileged level. The customer may want to know the difference between enabling a password and enabling a secret. The customer may ask you how to secure access to the console line and how to secure remote access by enabling and limiting access to SSH. You will also explain how to protect vty with a standard numbered access control list.

Network Device Security Overview

Many forms of security threats have emerged because of the rapid growth of the Internet. Viruses, Trojan horse attacks, malicious hackers, and even the employees of an organization are potential security hazards to corporate networks. These threats have the potential to steal and destroy sensitive corporate data, tie up valuable resources, and inflict major damage due to network downtime. This situation may lead to a cost crisis and cripple the company financially. Security breaches are also encountered more frequently in home or private networks. Everyone has a reason to be concerned.



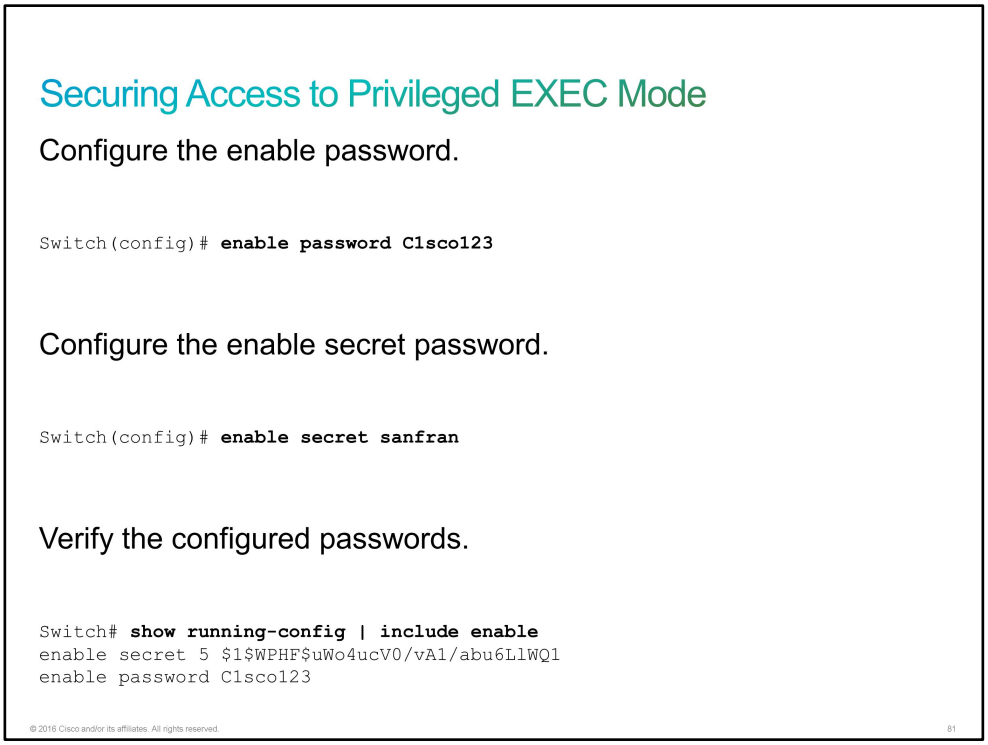
Common threats to network device security and mitigation strategies can be summarized as follows:

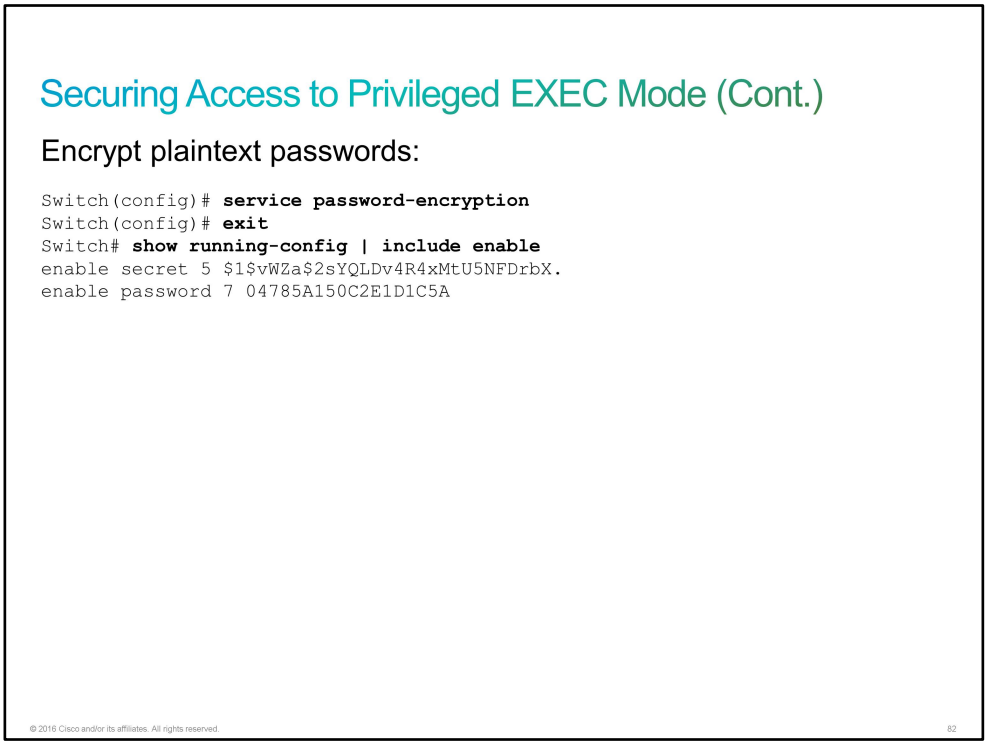
- **Remote access threats:** Unauthorized remote access is a threat when security is weak in remote access configuration. Mitigation techniques for this type of threat include configuring strong authentication and encryption for remote access policy and rules, configuration of login banners, use of ACLs, and VPN access.
- **Local access and physical threats:** These threats include physical damage to network device hardware, password recovery that is allowed by weak physical security policies, and device theft. Mitigation techniques for this type of threat include locking the wiring closet and allowing access only to authorized personnel. It also includes blocking physical access through a dropped ceiling, raised floor, window, duct work, or other possible point of entry. Use electronic access control, and log all entry attempts. Monitor facilities with security cameras.
- **Environmental threats:** Temperature extremes (heat or cold) or humidity extremes (too wet or too dry) can present a threat. Mitigation techniques for this type of threat include creating the proper operating environment through temperature control, humidity control, positive air flow, remote environmental alarms, and recording, and monitoring.

- **Electrical threats:** Voltage spikes, insufficient supply voltage (brownouts), unconditioned power (noise), and total power loss are potential electrical threats. Mitigation techniques for this type of threat include limiting potential electrical supply problems by installing UPS systems and generator sets, following a preventative maintenance plan, installing redundant power supplies, and using remote alarms and monitoring.
- **Maintenance threats:** These threats include improper handling of important electronic components, lack of critical spare parts, poor cabling, and inadequate labeling. Mitigation techniques for this type of threat include using neat cable runs, labeling critical cables and components, stocking critical spares, and controlling access to console ports.

Securing Access to Privileged EXEC Mode

You can secure a router or a switch by using passwords to restrict access. Using passwords and assigning privilege levels is a way to provide terminal access control in a network. It is a form of management plane hardening. You can establish passwords on individual lines, such as the console, and to the privileged EXEC mode. Passwords are case-sensitive.





Note	The passwords that the figure shows are for instructional purposes only. Passwords that are used in an actual implementation should meet the requirements of strong passwords.

The **enable password** global command restricts access to the privileged EXEC mode. You can assign an encrypted form of the enable password, which is called the enable secret password, by entering the **enable secret** *password* command at the global configuration mode prompt with the desired password. When you configure the enable secret password, it is used instead of the enable password rather than in addition to it.

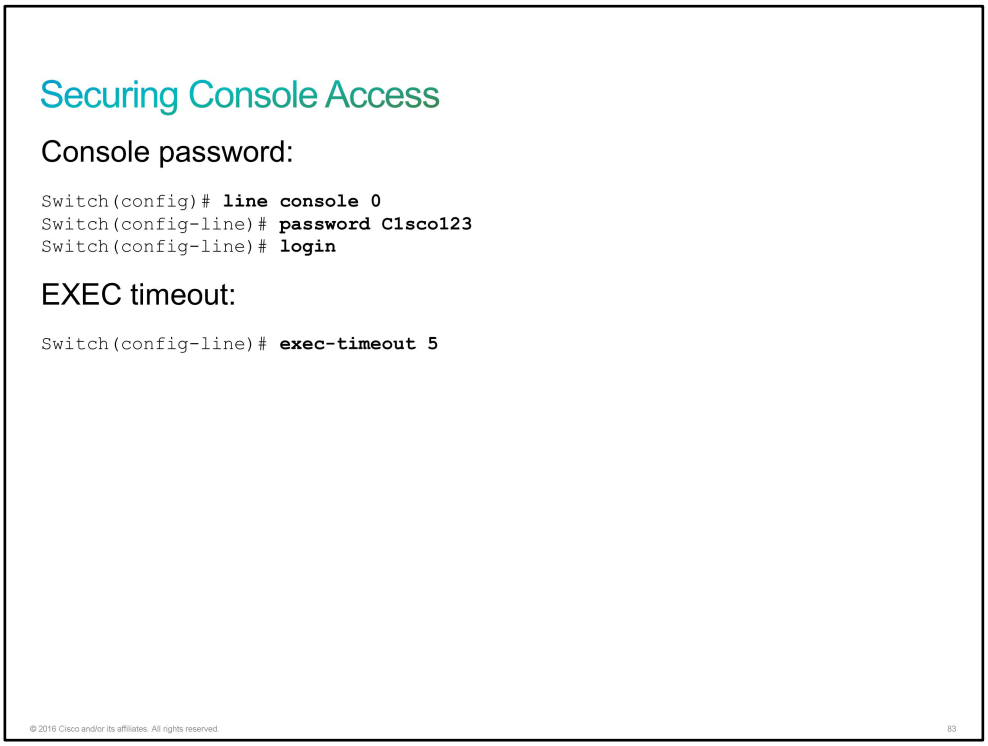
You can also add a further layer of security, which is particularly useful for passwords that cross the network or are stored on a TFTP server. Cisco provides a feature that allows the use of encrypted passwords. To set password encryption, enter the **service password-encryption** command in the global configuration mode.

Passwords that are displayed or set after you configure the **service password-encryption** command will be encrypted. Service password encryption uses type-7 encryption, which is not very secure. There are several tools and web pages available that convert an encrypted password into a plaintext string.

On the other hand, the **enable secret** command uses the MD5-type encryption that, to this point, has not been broken. It is recommended that you always use the **enable secret** *password* instead of the **enable password** command.

Securing Console Access

Use the **line console 0** command followed by the **password** and **login** subcommands to require login and establish a login password on a console terminal. By default, logging in is not enabled on the console.

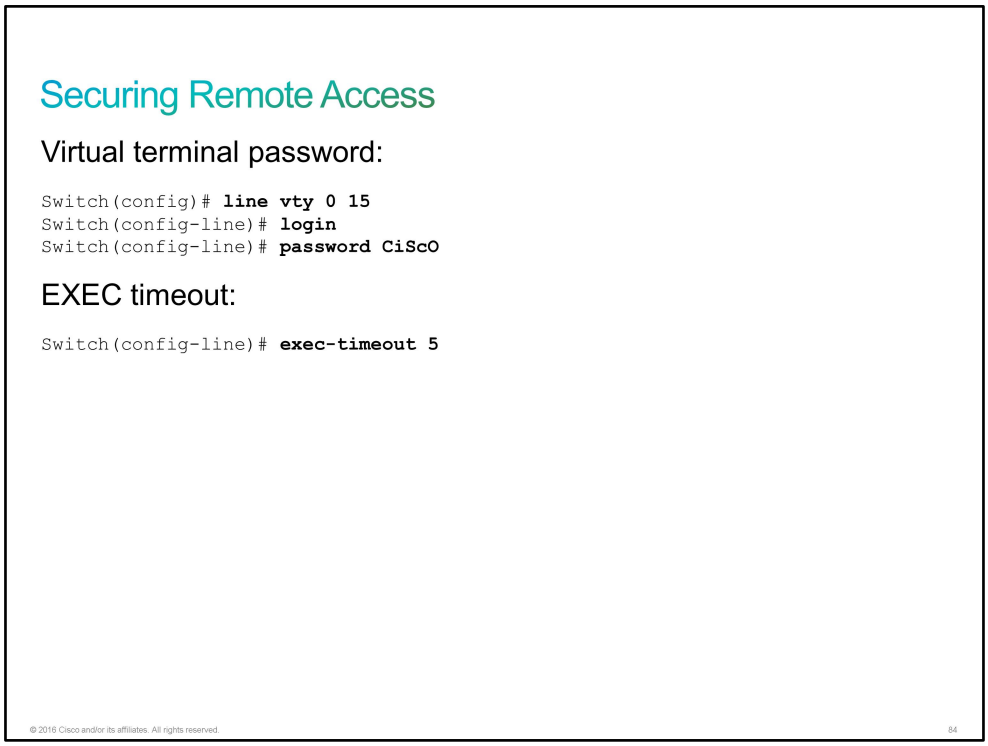


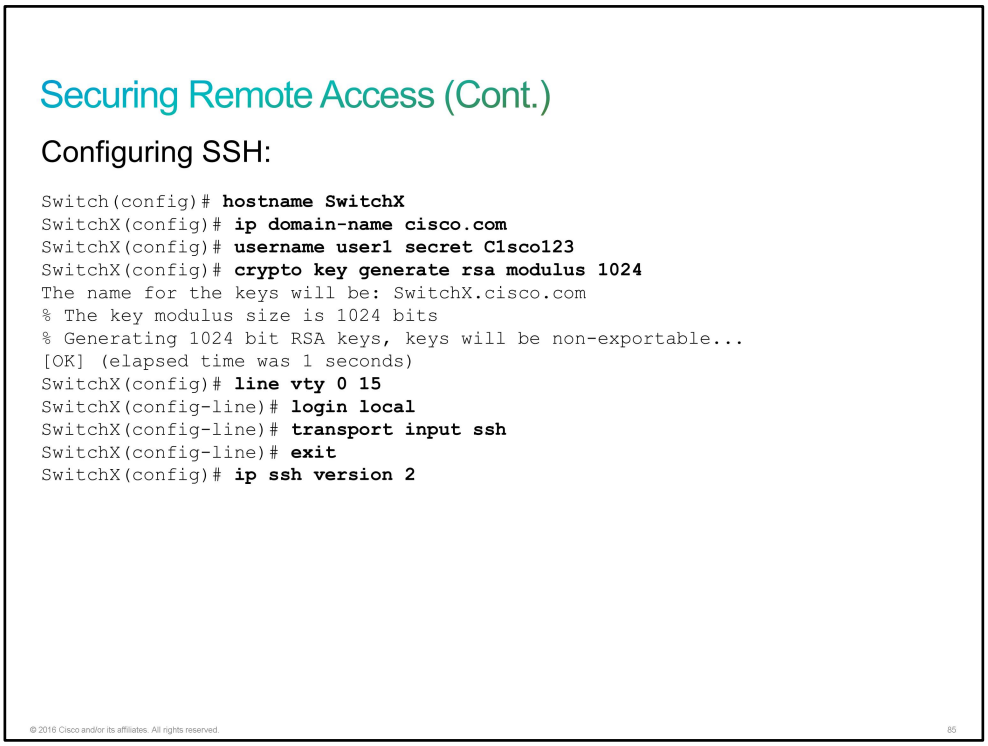
Note	Enter the **service password-encryption** command in the global configuration mode to encrypt the console password. Although this encryption is weak and can be easily decrypted, it is still better than a cleartext password. At least you are protected against exposing the password to casual observers.

The **exec-timeout** command prevents users from remaining connected to a console port when they leave a station. In the example, when no user input is detected on the console for 5 minutes, the user that is connected to the console port is automatically disconnected.

Securing Remote Access

You can establish an SSH connection to the SSH-enabled device using an SSH client on your PC, such as PuTTY. When you establish a connection for the first time from a specific computer, you are presented with a security alert window that indicates that the server host key is not cached in the PuTTY cache. By adding a key to the cache, you will avoid seeing this security alert window every time that you establish an SSH connection from this computer.





The **line vty 0 15** command, followed by the **login** and **password** subcommands, requires login and establishes a login password on incoming Telnet sessions.

You can use the **login local** command to enable password checking on each user by using the username and secret password that are specified with the **username** global configuration command. The **username** command establishes username authentication with encrypted passwords.

The **exec-timeout** command prevents users from remaining connected to a vty port when they leave a station. In the example, when no user input is detected on a vty line for 5 minutes, the vty session is automatically disconnected.

To configure SSH on a Cisco switch or router, you need to complete the following steps:

1. Use the **hostname** command to configure the hostname of the device so that it is not *Switch* (on a Cisco switch) or *Router* (on a Cisco router).
2. Configure the DNS domain with the **ip domain name** command. The domain name is required to be able to generate certificate keys.
3. Generate RSA keys that the user will use in authentication—use the **crypto key generate rsa** command.
4. Configure the user credentials that the user will use for authentication. By specifying the **login local** command for vty lines, you are essentially telling the network device to use locally defined credentials for authentication. Configure locally defined credentials using the **username** *username* **secret** *password* command.
5. (Optional) You can also limit access to a device to users that use SSH and block Telnet with the **transport input ssh** vty mode command. If you want to support login banners and enhanced security encryption algorithms, force SSH version 2 on your device with the **ssh version 2** command in the global configuration mode.

Securing Remote Access (Cont.)

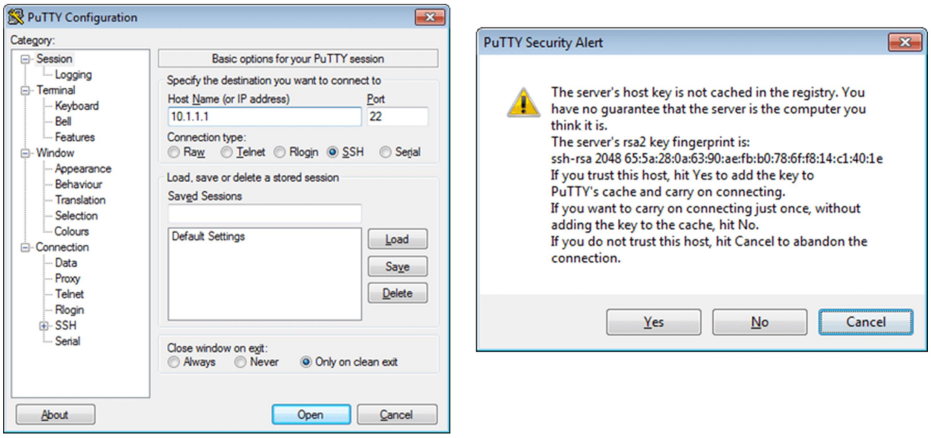
Verify that SSH is enabled:

```
Switch# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

Check the SSH connection to the device:

```
Switch# show ssh
Connection  Version  Encryption     State             Username
0           2.0      3DES           Session started   cisco
```


Securing Remote Access (Cont.)



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, SSHv2 is enabled.

To check the SSH connection to the device, use the **show ssh** command.

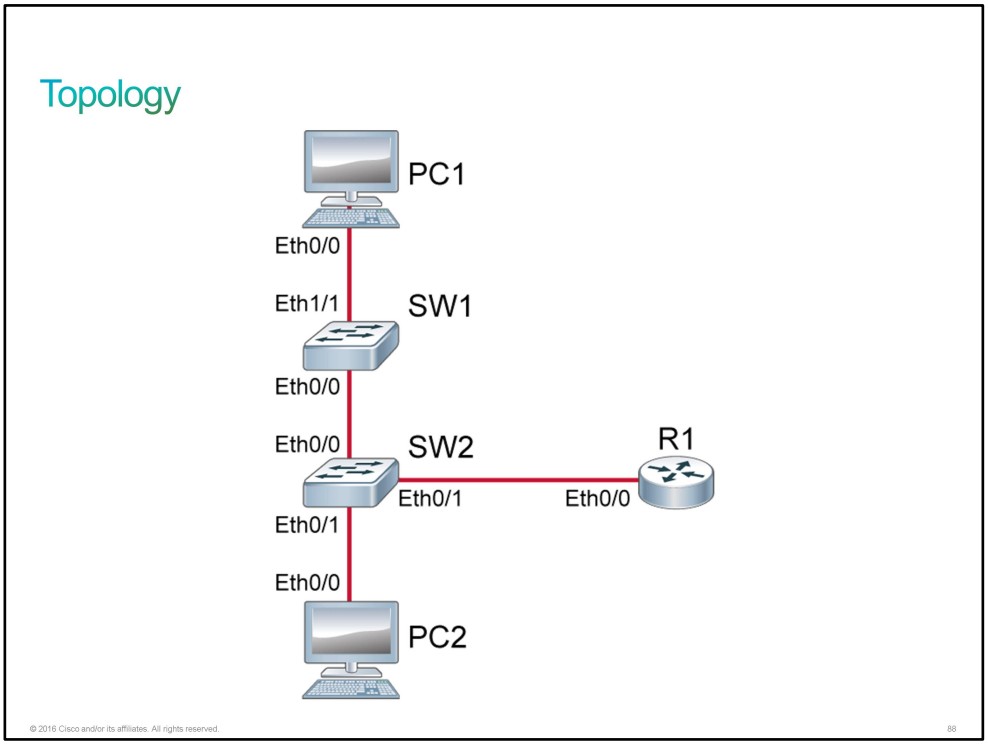
Discovery 22: Enhance Security of Initial Configuration

Introduction

This discovery lab will guide you through the various aspects of securing administrative access to Cisco IOS devices. You will secure access to the privileged EXEC, and see the difference between enable password and enable secret. You will also secure access to the console port. You will enable remote access to the vty lines via Telnet and SSH. You will set SSH as the only acceptable remote access protocol.

The devices are configured as represented in the topology diagram, including their IP addresses. This discovery lab will focus on R1. You will use other devices as sources of remote access connections.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24

Device	Characteristic	Value
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Secure Access to Privileged EXEC Mode

Activity

Step 1 On R1, access the privileged EXEC with the **enable** command and the global configuration with the **configure terminal** command.

On R1, enter the following commands:

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

The most commonly used commands are abbreviated in this guided discovery. For example, **en** for **enable** and **conf t** for **configure terminal**. If there is any confusion, you can attempt tab completion of commands to see the full commands during the discovery execution. For example, **en<tab>** would expand to **enable** and **conf<tab> t<tab>** would expand to **configure terminal**.

Step 2 Set the enable password to "Password123" and leave the configuration mode.

On R1, enter the following commands:

```
R1(config)# enable password Password123
R1(config)# end
R1#
```

Step 3 The enable password will now protect access to the privileged EXEC. Verify this fact by leaving the privileged EXEC with the **disable** command, and then use **enable** again, and authenticate with the password "Password123."

On R1, enter the following commands:

```
R1# disable
R1> enable
Password: Password123
R1#
```

Step 4 View the enable password in the running configuration.

On R1, enter the following command:

```
R1# sh run | inc enable
enable password Password123
```

By default, the enable password is stored in the configuration as clear text.

Step 5 Configure an enable secret, setting it to "Secret123."

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# enable secret Secret123
R1(config)# end
R1#
```

Step 6 When both are present, the enable secret takes precedence over the enable password. Verify that this fact is correct.

On R1, enter the following commands:

```
R1# disable
R1> enable
Password: Password123
Password: Secret123
R1#
```

The enable password was not accepted to access privileged EXEC. The enable secret was required.

Step 7 View the enable password and the enable secret in the configuration.

On R1, enter the following command:

```
R1# sh run | inc enable
enable secret 4 9h/bNbZRK8Hm9J2ONmwUdf0KoztPJewuR2NseOBKzM6
enable password Password123
```

The enable secret is always stored in a protected fashion in the configuration file. Cisco IOS on routers supports several encryption types. On production routers, you will most likely find type 8 or type 9. Using type 4 is not recommended due to security risks.

Step 8 Enable the **service password-encryption** in the configuration mode. Then revisit how the enable credentials appear in the running configuration.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# service password-encryption
R1(config)# end
R1# sh run | inc enable
enable secret 4 9h/bNbZRK8Hm9J2ONmwUdf0KoztPJewuR2NseOBKzM6
enable password 7 03345A1815182E5E4A584B56
```

The **enable password** is now protected using the Cisco IOS type 7 encryption algorithm. Understand that type 7 encryption is better than nothing, but it is nowhere near as strong as type 5 MD5. The service password encryption will also protect other cleartext passwords that may appear in the configuration file.

Task 2: Secure Console and Remote Access

Activity

Step 1 Enable a password on the console of R1 (line console 0) by using the **login** command with the **password** command. Set the password to "Console123." Also, set the exec-timeout value to 5 minutes.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line con 0
R1(config-line)# login
% Login disabled on line 0, until 'password' is set
R1(config-line)# password Console123
R1(config-line)# exec-timeout 5
R1(config-line)# end
R1#
```

The default value for exec-timeout is 10 minutes. Setting the time longer may be a convenience for the administrator. Setting the time shorter improves security by limiting the time that a session stays up if the administrator physically walks away from the terminal.

Step 2 View the configuration that is now in place on "line con 0."

On R1, enter the following command:

```
R1# sh run | section line
line con 0
 exec-timeout 5 0
 password 7 080243401A16091243595F
 logging synchronous
 login
line aux 0
line vty 0 4
 login
 transport input all
```

Service password-encryption continues to encrypt new passwords as they are defined.

Step 3 Verify the console password by logging out completely from the Cisco IOS CLI session on R1 and then logging back in. Continue by using the **enable** command to access the privileged EXEC.

On R1, enter the following commands:

```
R1# logout

R1 con0 is now available

Press RETURN to get started.
<Enter>

User Access Verification

Password: Console123
R1> en
Password: Secret123
R1#
```

Step 4 In a similar fashion, add a password to the five vty lines (line vty 0 4), setting the credential that is required for remote access to the CLI of R1. Also, for demonstration purposes, set the exec-timeout to the very small value of 0 minutes and 30 seconds.

On R1, enter the following commands:


```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line vty 0 4
R1(config-line)# login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password' is set
R1(config-line)# password VTYPass
R1(config-line)# exec-timeout 0 30
R1(config-line)# end
R1#
```

Step 5 Verify that you can now access the CLI of R1 via Telnet from other systems. Access the console of PC1 and telnet to 10.10.1.1.

On PC1, enter the following commands:

```
PC1> telnet 10.10.1.1
Trying 10.10.1.1 ... Open

User Access Verification

Password: VTYPass
R1>
```

The prompt changed from PC1 to R1. You are currently accessing the console of PC1, but using PC1 to remotely access the CLI of R1.

Step 6 Verify that you can use this remote connection to access the R1 privileged EXEC with the **enable** command and the "Secret123" enable secret. Then remain idle for 30 seconds and verify the functioning of the exec-timeout.

On PC1, enter the following commands:

```
R1> en
Password: Secret123
R1# <wait 30 seconds for exec-timeout>
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

After the exec-timeout expired, the telnet session was closed. You have returned to the PC1 CLI.

Step 7 Return to the console of R1.

If it has been longer than the 5 minute exec-timeout set on line con 0, you will have to reauthenticate using "Console123" as the console password and "Secret123" as the enable password.

Step 8 You will now increase the sophistication of the login process. Instead of using simply a password for remote access, you will require a username and a password. The first step is to define a username in the configuration. Enter the configuration mode, and then use the **?** to display the options that are available as you configure a username.

On R1, enter the following commands:

```
R1> en
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# username ?
  WORD  User name
```

The next element of the command line is to specify the username as a freeform WORD.

Step 9 Continue by specifying "admin" as the username followed by the **?** to display the next set of options.

On R1, enter the following command:

```
R1(config)# username admin ?
  aaa                 AAA directive
  access-class        Restrict access by access-class
  autocommand         Automatically issue a command after the user logs in
  callback-dialstring Callback dialstring
  callback-line       Associate a specific line with this callback
  callback-rotary     Associate a rotary group with this callback
  dnis                Do not require password when obtained via DNIS
  nocallback-verify   Do not require authentication after callback
  noescape            Prevent the user from using an escape character
  nohangup            Do not disconnect after an automatic command
  nopassword          No password is required for the user to log in
  one-time            Specify that the username/password is valid for only one
                      time
  password            Specify the password for the user
  privilege           Set user privilege level
  secret              Specify the secret for the user
  user-maxlinks       Limit the user's number of inbound links
  view                Set view name
  <cr>
```

There are several options available, but for this purpose, focus only on "password" and "secret." Understand that the differences regarding the **username** command are the same as they are with the enable password and the enable secret.

Step 10 Continue by specifying secret as the credential storage option and use the **?** to display the next set of options.

On R1, enter the following command:

```
R1(config)# username admin secret ?
  0     Specifies an UNENCRYPTED secret will follow
  4     Specifies a SHA256 ENCRYPTED secret will follow
  5     Specifies a MD5 ENCRYPTED secret will follow
  LINE  The UNENCRYPTED (cleartext) user secret
```

You can specify a 4 followed by an SHA-256-protected secret or a 5 followed by an MD5-protected secret. These options allow you to copy the protected secret from one configuration to another. There is also the option to specify a 0 followed by the clear text secret. Specifying the 0 is optional and generally not used. In this case, you do not have a protected secret to work with. You will simply enter the cleartext secret next.

Step 11 Complete the definition of the username "admin" with the "Cisco123" secret.

On R1, enter the following command:


```
R1(config)# username admin secret Cisco123
```

Step 12 Remain in configuration mode. Use the **do** command to execute the privileged EXEC **show running-config** command from within configuration mode. Send the output through the include filter specifying the "user" string.

On R1, enter the following command:

```
R1(config)# do show run | inc user
username admin secret 4 vwcGVdcUZcRMCyxaH2U9Y/PTujsnQWPSbt.LFG8lhTw
```

The username "admin" is now defined and its password is stored in the configuration as a Cisco IOS type 4, SHA-256-protected secret.

Step 13 Currently, the vty lines have the **login** command set without an argument. In this state, authentication is done using the password that is defined on the line itself. If the **login** command is enhanced with the **local** argument, then authentication will be accomplished using usernames stored in the local running configuration. Enter the vty line configuration mode and configure the **login local** command. Then leave the configuration mode.

On R1, enter the following commands:

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# end
R1(config-line)# no password
R1#
```

Step 14 Remote access authentication should now require a valid username and password. Access the console of PC1 and attempt to telnet to R1 (10.10.1.1) to verify this fact.

On PC1, enter the following commands:

```
PC1> telnet 10.10.1.1
Trying 10.10.1.1 ... Open

User Access Verification

Username: admin
Password: Cisco123
R1>
```

Step 15 Recall that the exec-timeout on the vty lines is set to 0 minutes and 30 seconds. Quickly continue to use the remote connection to modify the configuration. As long as less than 30 seconds pass between keystrokes, the connection should remain open. If you do get logged out, you can return to R1 to enter the configuration changes.

On PC1 console while telneted to R1, enter the following commands:

```
R1> en
Password: Secret123
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line vty 0 4
R1(config-line)# exec-timeout 10
R1(config-line)# end
R1#
```

You may be accustomed to seeing a "SYS-5-CONFIG_I" syslog message displayed when leaving the configuration mode. That message did not appear here. By default, syslog messages are displayed on the console (line con 0) only. In this case, you have a session to a vty line. You can see the syslog message on the R1 console screen.

Changing the configuration via a remote access session is itself a demonstration of the importance of securing administrative access. Without proper security, network attackers could access your network devices and take control of your network.

Step 16 From this remote connection, review the configuration of the vty lines with a filtered **show running-config** command.

On PC1 console while telneted to R1, enter the following commands:

```
R1# show run | section line
line con 0
 exec-timeout 5 0
 password 7 080243401A16091243595F
 logging synchronous
 login
line aux 0
line vty 0 4
 login local
 transport input all
```

There is no longer an exec-timeout command on line vty 0 4. Setting the timeout value back to its default value of 10 minutes causes the command to be hidden in the running configuration.

Step 17 Close the remote access connection using either the **logout** or **exit** command.

On PC1 console while telneted to R1, enter the following commands:

```
R1# logout
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

From the user or privileged EXEC, you can use the **logout** and **exit** commands interchangeably to terminate remote access connections.

Task 3: Enable SSH

Activity

Step 1 On R1, enable SSH. The prerequisite of SSH on Cisco IOS vty lines is having an RSA public/private key pair. The prerequisite to defining the key pair is to have a hostname and a

domain name defined. R1 already has a hostname configured. Configure "icnd.lab" as the domain name, then generate a 1024-bit RSA public/private key pair.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip domain-name icnd.lab
R1(config)# crypto key generate rsa
The name for the keys will be: R1.icnd.lab
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 0 seconds)

R1(config)#
*Nov 12 12:43:54.804: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

Shortly after you generate the RSA keypair, SSH is automatically enabled on the router.

Step 2 There are some security issues that are associated with SSHv1. Limit the options to SSHv2 only, then leave configuration mode.

On R1, enter the following commands:

```
R1(config)# ip ssh version 2
R1(config)# end
R1#
```

Step 3 Both Telnet and SSH should now be options for remote access to R1. Access the console of PC1 to verify this fact. First telnet from PC1 to R1. Disconnect when you are connected.

On PC1, enter the following command:

```
PC1> telnet 10.10.1.1
Trying 10.10.1.1 ... Open

User Access Verification
Username: admin
Password: Cisco123

R1> exit
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

Step 4 Next, test an SSH connection from PC1 to R1.

On PC1, enter the following command:

```
PC1> ssh -l admin 10.10.1.1
Password: Cisco123
R1>
```

The "–l" is a dash and a lower-case "L" letter, not a dash with the number 1. Think lower-case "L" for "login."

Step 5 From this remote access connection, examine the vty configuration to understand why both Telnet and SSH are allowed. It is because "transport input all" is specified on the "line vty 0 4" configuration.

On R1, enter the following command:

```
R1> en
Password: Secret123
R1# show run | section line
line con 0
 exec-timeout 5 0
 password 7 080243401A16091243595F
 logging synchronous
 login
line aux 0
line vty 0 4
 password 7 0125323D6B0A151C
 login local
 transport input all
```

Step 6 Because SSH is superior to Telnet from a security perspective, change the transport input option from **all** to **ssh** under "line vty 0 4."

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line vty 0 4
R1(config-line)# transport input ssh
R1(config-line)# end
```

Step 7 Terminate this SSH session, to return to the local console of PC1.

On R1, enter the following command:

```
R1# logout
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

Step 8 Attempt a Telnet session from PC1 to R1. Because the transport input is now set to SSH only, the Telnet attempt should be rejected.

On PC1, enter the following command:

```
PC1> telnet 10.10.1.1
Trying 10.10.1.1 ...
% Connection refused by remote host
```

Step 9 Verify that SSH is still valid for new remote access sessions. Try to connect with SSH one more time from PC1 to R1. Terminate the session after it successfully initiates.

On PC1, enter the following command:

```
PC1> ssh -l admin 10.10.1.1
Password: Cisco123
R1> logout
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

You have explored many options for securing administrative access on Cisco IOS devices during this discovery lab. You used the enable password and the enable secret to protect the privileged EXEC. You used service password encryption to provide simple protection to cleartext passwords. You implemented a simple password protection on the console and the vty lines. You then went further with the vty lines, requiring a username and password for access, and configuring SSH. Feel free to continue exploring these concepts independently within the lab environment.

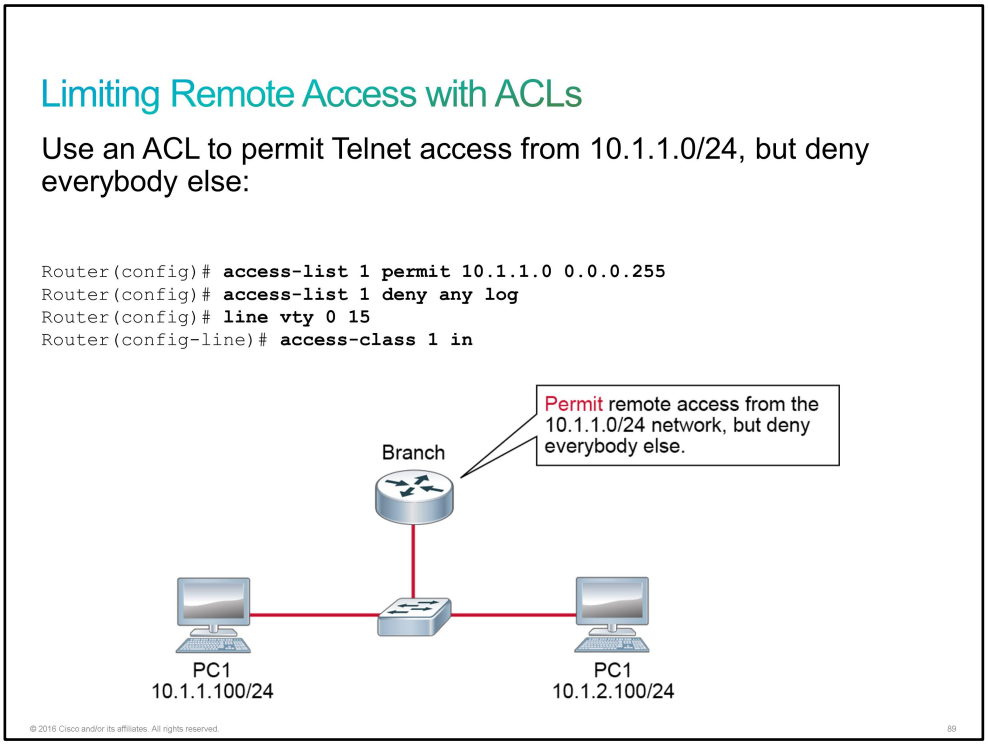
This is the end of the discovery lab.

Limiting Remote Access with ACLs

You can limit access to vty lines to specific IP addresses or subnets in order to control remote administration of network devices. Remote administration is commonly run over a Telnet or an SSH connection, where the SSH connection is an encrypted communication channel between the administrator workstation and the device.

Usually, there are two steps that you must complete to limit remote access with ACLs:

1. **Configure an ACL:** The following example shows an ACL being configured with two lines. The first line permits Telnet access from the network addresses in the 10.1.1.0/24 subnet. The second line is not mandatory because there is an implicit deny statement at the end of every ACL. However, creating an explicit deny statement and appending the **log** keyword allows you to monitor attempts of unauthorized sources trying to access the device.



2. **Apply the ACL to the lines:** The **access-class** command applies the ACL on vty lines. Using the **in** keyword after the name of the ACL tells the router to limit vty connections that are coming into the network device.

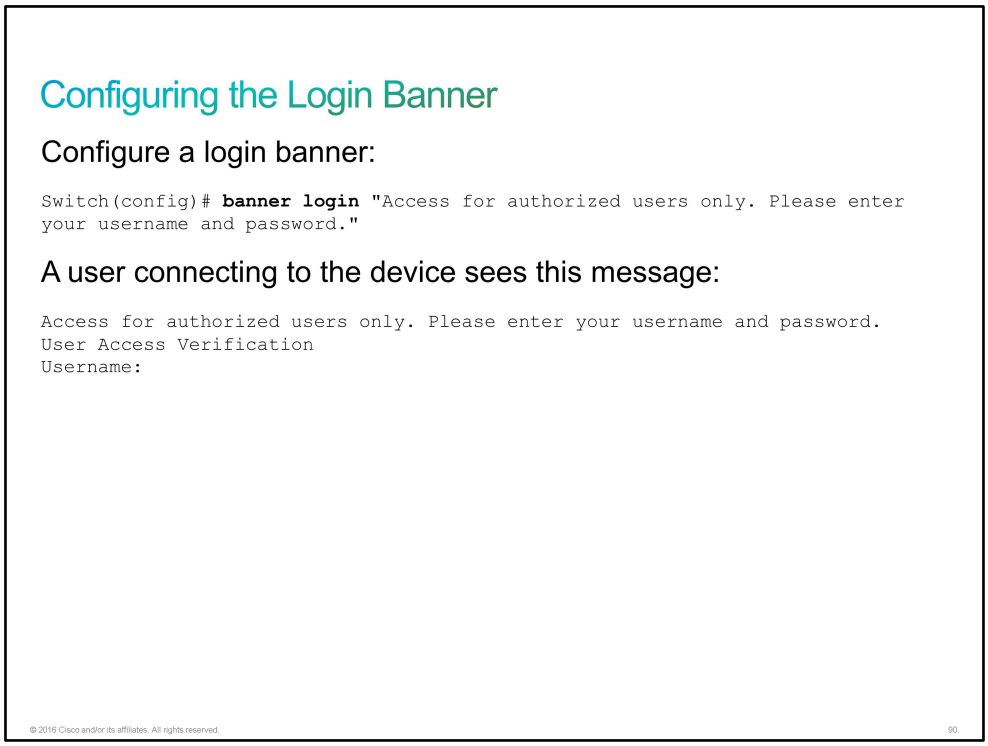
Apply the ACL on vty lines:

```
Router(config)# line vty 0 15
Router(config-line)# access-class 1 in
```

The example uses 16 vty lines (range 0 to 15). In the configuration output, it will appear in two vty line ranges, first from 0 to 4 and second from 5 to 15. If there is no need for more than 5 vty lines, you may configure only first range.

Configuring the Login Banner

You can define a customized login banner to be displayed before the username and password login prompts.



To configure a login banner, use the **banner login** command in global configuration mode. Enclose the banner text in quotation marks or use a delimiter that is different from any character appearing in the banner string.

Note	Use caution when you create the text that is used in the login banner. Words like "welcome" may imply that access is not restricted and may allow hackers some legal defense of their actions.

To define and enable an MOTD banner, use the **banner motd** command in global configuration mode.

This MOTD banner is displayed to all terminals that are connected and is useful for sending messages that affect all users (such as impending system shutdowns).

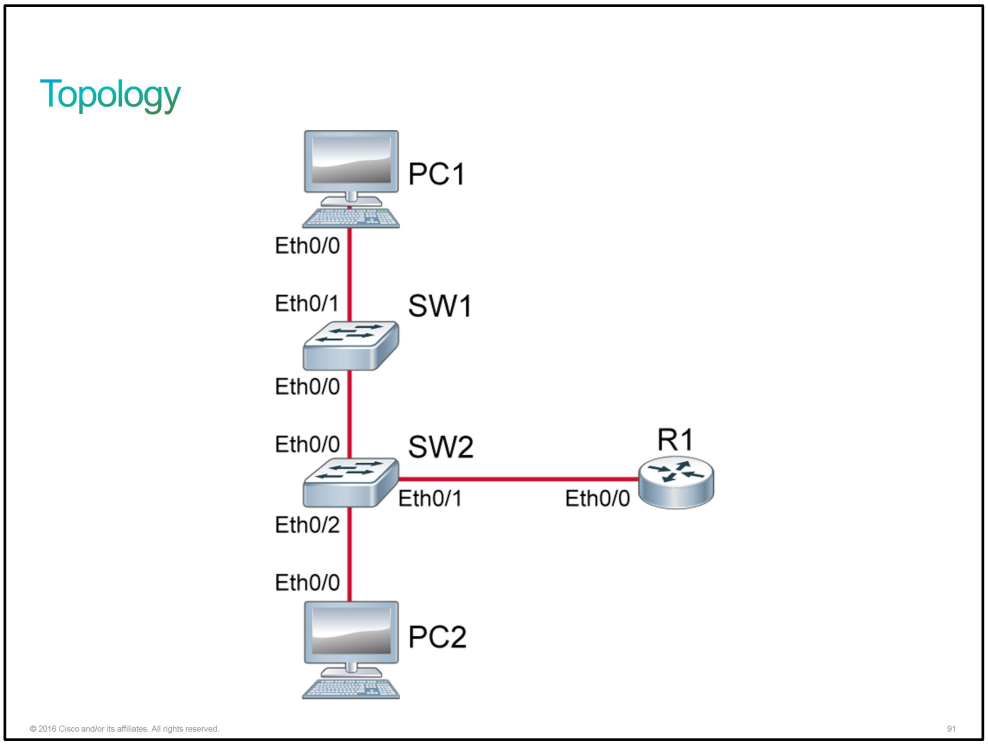
Discovery 23: Limit Remote Access Connectivity

Introduction

This discovery lab will guide you through the remote access limitation by using an ACL. You will implement login and exec banners.

The devices are configured as represented in the topology diagram, including IP addresses. This discovery lab will focus on R1. Other devices will be used as sources of remote access connections.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2

Device	Characteristic	Value
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24
R1	Enable password	Password123
R1	Enable secret	Secret123
R1	Console password	Console123
R1	Console Login Mode	line
R1	VTY 0 4 Line Password	VTYPass
R1	VTY 0 4 Login Mode	local
R1	Username / Secret	admin / Cisco123

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Limit Remote Access with ACLs

Activity

Step 1 Explore the use of ACLs to control the source IP addresses that are allowed to establish remote access sessions to a Cisco IOS device. Before defining new access lists, you should know which access lists exist to prevent accidental editing of an access list that is already defined. On R1, view the access lists that are in place. To access R1, use the console password "Console123" and the "Secret123" enable secret.

On R1, enter the following command:

```
R1 con0 is now available

Press RETURN to get started.
<Enter>

User Access Verification

Password: Console123
R1> en
Password: Secret123
R1# show access-list
R1#
```

On R1, no access lists are configured.

Step 2 Enter the global configuration mode and define a new access list number 1 that permits PC1 (10.10.1.10) and PC2 (10.10.1.20) and explicitly denies all other addresses with the log option enabled.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# access-list 1 permit 10.10.1.10
R1(config)# access-list 1 permit 10.10.1.20
R1(config)# access-list 1 deny any log
```

Step 3 Assign the access list 1 to the vty lines in the inbound direction using the **access-class** command, then leave the configuration mode.

On R1, enter the following commands:

```
R1(config)# line vty 0 4
R1(config-line)# access-class 1 in
R1(config-line)# end
R1#
```

Step 4 One at a time, access the consoles of PC1, PC2, and SW1 and attempt SSH sessions to the R1 (10.10.1.1). The sessions should be successful from PC1 and PC2, but not from SW1.

On PC1, enter the following command:

```
PC1> ssh -l admin 10.10.1.1
Password: Cisco123
R1> logout
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

On PC2, enter the following command:

```
PC2> ssh -l admin 10.10.1.1
Password: Cisco123
R1> logout
[Connection to 10.10.1.1 closed by foreign host]
PC2>
```

On SW1, enter the following command:

```
SW1> ssh -l admin 10.10.1.1
% Connection refused by remote host
SW1>
```

Step 5 Return to the console of R1. You should find a syslog message that is associated with the access attempt from SW1 that the explicit deny statement at the end of the access list denied.

Syslog message displayed on the R1 console:

```
R1#
*Nov 12 13:03:18.445: %SEC-6-IPACCESSLOGNP: list 1 denied 0 10.10.1.2 ->
0.0.0.0, 1 packet
```

Step 6 On the R1, show access list 1 and verify the match counters on each line.

On R1, enter the following command:

```
R1# show access-list 1
Standard IP access list 1
    10 permit 10.10.1.10 (2 matches)
    20 permit 10.10.1.20 (2 matches)
    30 deny   any log (1 match)
```

When an ACL is applied to the vty lines with the **access-class** command, each successful connection will increment the match counter on the associated permit statement by two while each rejected connection will only increment the match counter on the associated deny statement by one.

Task 2: Configure the Login and EXEC Banners

Activity

Step 1 Another access control option that you will explore during this discovery is the use of banner messages. The login banner is displayed before the user logs in, and the EXEC banner is displayed after a successful login. Start by configuring a login banner on R1.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# banner login "
Enter TEXT message.  End with the character '"'.
Access for authorized users only.
Enter your valid credentials for access:
"
```

Step 2 Also, configure an EXEC banner on R1, and then leave the configuration mode.

On R1, enter the following commands:

```
R1(config)# banner exec "
Enter TEXT message.  End with the character '"'.
AUTHORIZED ACCESS ONLY!
If you are not authorized, LOGOUT IMMEDIATELY
"
R1(config)# end
R1#
```

Step 3 Access the console of PC1 and execute an SSH connection to R1. You should see the login banner before entering the password and you should see the EXEC banner after authentication and before the first user EXEC prompt is displayed.

On PC1, enter the following commands:

```
PC1> ssh -l admin 10.10.1.1
Access for authorized users only.
Enter your valid credentials for access:

Password: Cisco123

AUTHORIZED ACCESS ONLY!
If you are not authorized, LOGOUT IMMEDIATELY
R1> logout
[Connection to 10.10.1.1 closed by foreign host]
PC1>
```

You have explored how to limit remote access connections during this discovery. You have limited authorized remote access systems with ACLs. You finished with a simple demonstration of the login and EXEC banners. Feel free to continue exploring these concepts independently within the lab environment.

This is the end of the discovery lab.

Challenge

1. High levels of humidity posing a danger to devices operating properly can be considered to be which of the following types of threats?

   A. Remote Access Threats
   B. Local Access and Physical Threats
   C. Environmental Threats
   D. Electrical Threats
   E. Maintenance Threats

2. Which of the following commands will encrypt plain text passwords on Routers?

   A. **password encryption**
   B. **service password-encryption**
   C. **service encryption**
   D. **enable secret**

3. Which of the following commands enables you to configure the parameters for console access?

   A. **line console 0**
   B. **line console**
   C. **login console 0**
   D. **login console**

4. Which of the following is the correct command to generate RSA keys that the user will use during authentication, when connecting over SSH to a router?

   A. **crypto key generate rsa**
   B. **crypto generate key rsa**
   C. **crypto rsa generate key**
   D. **crypto generate rsa key**

5. Choose the valid configuration to restrict remote users by applying ACL to vty lines.

   A. **router(config)# line vty 0 15**
      router(config-line)# access-group 1 in
   B. **router(config)# line vty 0 15**
      router(config-line)# access-list 1 in
   C. **router(config)# line vty 0 15**
      router(config-line)# access-class 1 in
   D. **router(config)# line vty 0 15**
      router(config-line)# ip access-group 1 in

6. Which of the following banners should be used to show information that should be hidden from unauthorized users?

   A. MOTD
   B. Login
   C. EXEC

7. Making sure that the cable runs are neat, is mitigation for which kind of threat?

   A. Remote Access Threats
   B. Environmental Threats
   C. Electrical Threats
   D. Maintenance Threats

Answer Key

Challenge

1. C
2. B
3. A
4. A
5. C
6. C
7. D

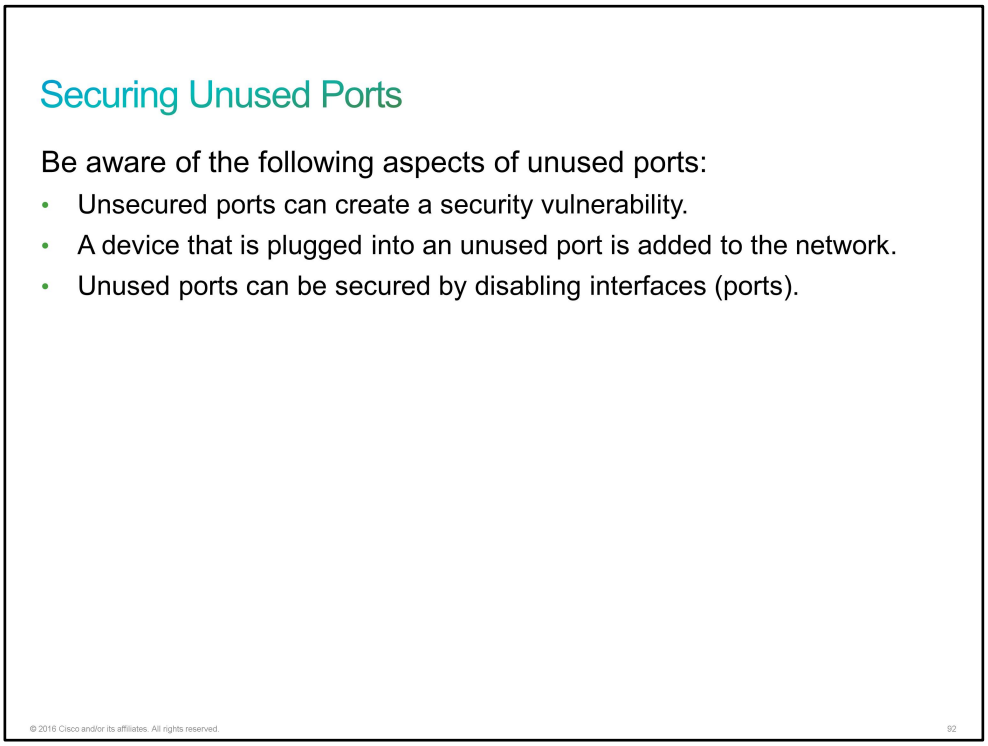
Lesson 2: Implementing Device Hardening

Introduction

Your boss sends you to your customer to secure unused ports. When discussing how to secure ports, you will introduce the **interface range** command. You will discuss the lack of control over utilized ports, and present port security as a possible solution. You will also explain the need to disable unused services. Your customer wants to implement the correct system time, so you will introduce NTP and demonstrate an NTP configuration example.

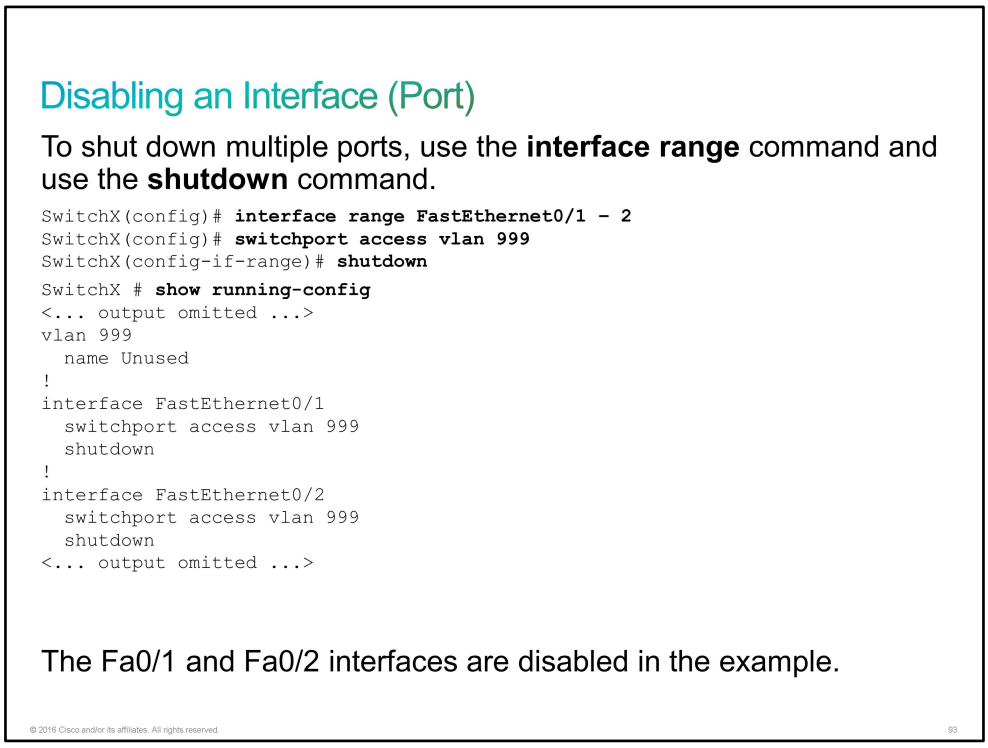
Securing Unused Ports

Unused ports on a switch can be a security risk. A hacker can plug a switch into an unused port and become part of the network. Therefore, unsecured ports can create a security hole.



Disabling an Interface (Port)

A simple method that many administrators use to help secure their network from unauthorized access is to disable all unused ports on a network switch.



Imagine, for example, that the Cisco switch has 24 ports. If there are 3 Fast Ethernet connections in use, the practicing good security demand is that you disable the 21 unused ports.

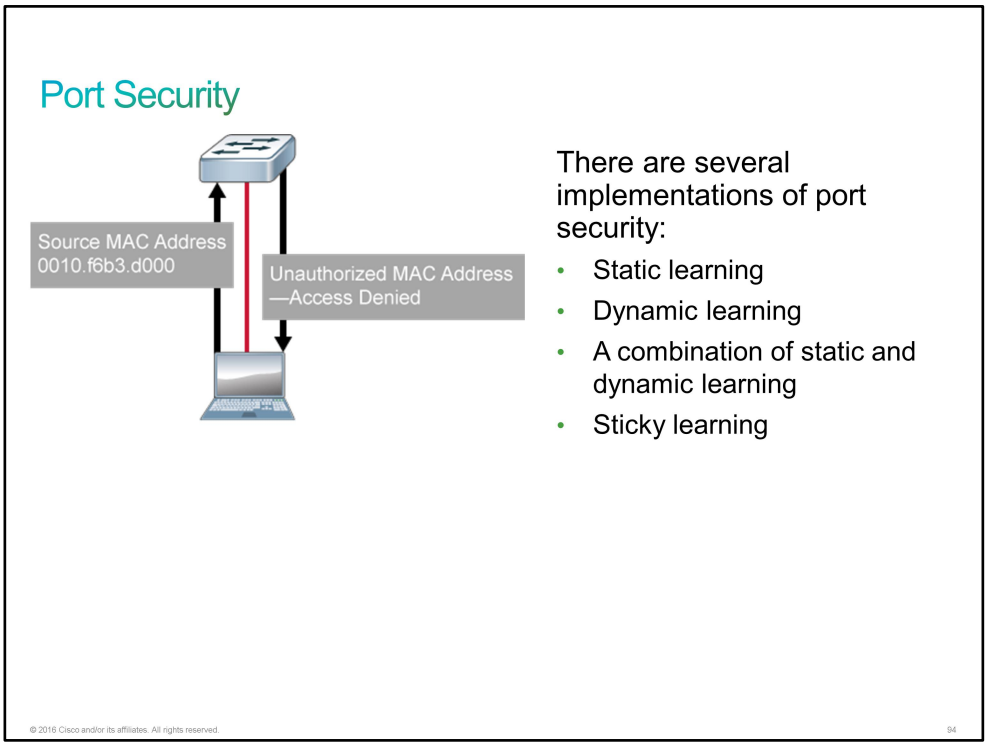
It is simple to disable multiple ports on a switch. Navigate to each unused port and issue the Cisco IOS **shutdown** command. An alternate way to shut down multiple ports is to use the **interface range** command. If a port needs to be activated, you can manually enter the **no shutdown** command on this interface.

The process of enabling and disabling ports can become a tedious task, but the enhanced security on your network is well worth the effort.

To make configuration more secure add unused ports into an unused vlan. Use **vlan** command, to configure new VLAN and use **switchport access vlan** interface command, to add port into VLAN.

Port Security

Now that you know about protecting unused ports, you need to learn how to protect the ports that are in use. You can use the port security feature of Cisco IOS Software to restrict access to a switch port based on MAC addresses. A port that is configured with port security accepts frames only from secure MAC addresses. You can configure a device to learn these addresses dynamically, or you can configure them statically.



There are several implementations of port security:

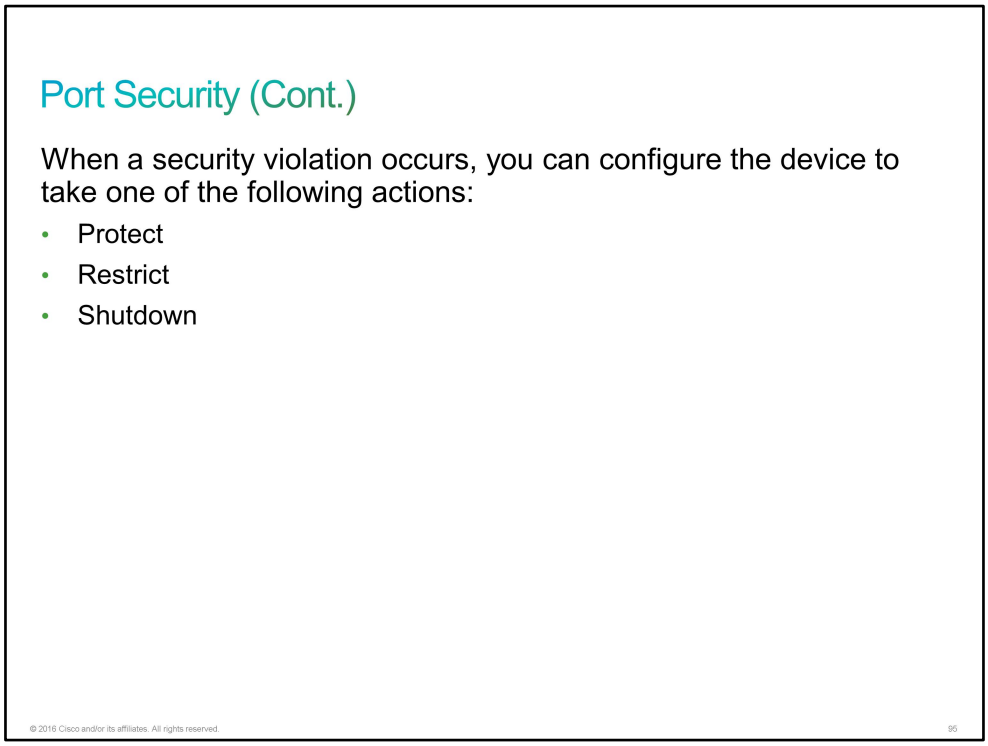
- **Static learning:** You can statically configure specific MAC addresses that are permitted to use a port. The source MAC addresses that you do not specifically permit are not allowed to source frames to the port.
- **Dynamic learning:** You can specify how many MAC addresses are permitted to use a port at one time. Use the dynamic approach when you care only about how many MAC addresses are permitted to use the port, rather than which MAC addresses are permitted. If a port on which dynamic learning is configured has a link-down condition, all dynamically learned addresses are removed. Following bootup, a reload, or a link-down condition, port security does not populate the address table with dynamically learned MAC addresses until the port receives ingress traffic. Depending on how you configure the device, dynamically learned addresses age out after a certain period and new addresses are learned, up to the maximum that you have defined.

- **A combination of static and dynamic learning:** You can specify some of the permitted MAC addresses and let the switch learn the rest of the permitted MAC addresses. For example, you could limit the number of MAC addresses to four and statically configure two of the MAC addresses. The switch would then dynamically learn the next two MAC addresses that it received on that port. The two statically configured addresses would not age out, but the two dynamically learned addresses could, depending on your configuration.
- **Sticky learning:** When you configure sticky learning on an interface, the interface converts dynamically learned addresses to "sticky secure" addresses. This feature adds the dynamically learned addresses to the running configuration as if they were statically configured. If you save the running configuration to NVRAM, port security with sticky MAC addresses saves dynamically learned MAC addresses in the startup configuration file, and the port does not have to learn addresses from ingress traffic after a bootup or a restart. Sticky secure addresses do not age out.

When a frame arrives on a port for which port security is configured, its source MAC address is checked against the secure MAC address table. If the source MAC address matches an entry in the table for that port, the device forwards the frame to be processed. Otherwise, the device does not forward the frame.

When port security is configured on a port, the following situations are considered security violations:

- The maximum number of secure MAC addresses has been added to the address table, and a host whose MAC address is not in the address table attempts to access the interface.
- A host with a secure MAC address that is configured or learned on one secure port attempts to access another secure port in the same VLAN.



You can configure the device to take one of the following actions when a security violation occurs:

- **Protect:** The protect violation mode drops packets with unknown source addresses until you remove enough secure MAC addresses to drop below the maximum value.
- **Restrict:** The restrict violation mode also drops packets with unknown source addresses until you remove enough secure MAC addresses to drop below the maximum value. However, it also generates a log message and causes the security violation counter to increment.
- **Shutdown:** The shutdown violation mode puts the interface into an error-disabled state immediately. The entire port is shut down. Also, in this mode, the system generates a log message, sends an SNMP trap, and increments the violation counter. To make the interface usable, you must use a manual intervention or the error-disabled recovery. Shutdown is the default violation mode.

  When the port security violation mode is set to shutdown, the port with the security violation goes to the error-disabled state. You receive this notification on the device:

```
Sep 20 12:44:54.966: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/5,
putting Fa0/5 in err-disable state
Sep 20 12:44:54.966: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred,
caused by MAC
address 000c.292b.4c75 on port FastEthernet0/5.
Sep 20 12:44:55.973: %LINEPROTO-5-PPDOWN: Line protocol on Interface FastEthernet0/5,
changed state
 to down
Sep 20 12:44:56.971: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to down
```

  To make the interface operational again, you need to disable the interface administratively and then enable it again, as shown here:

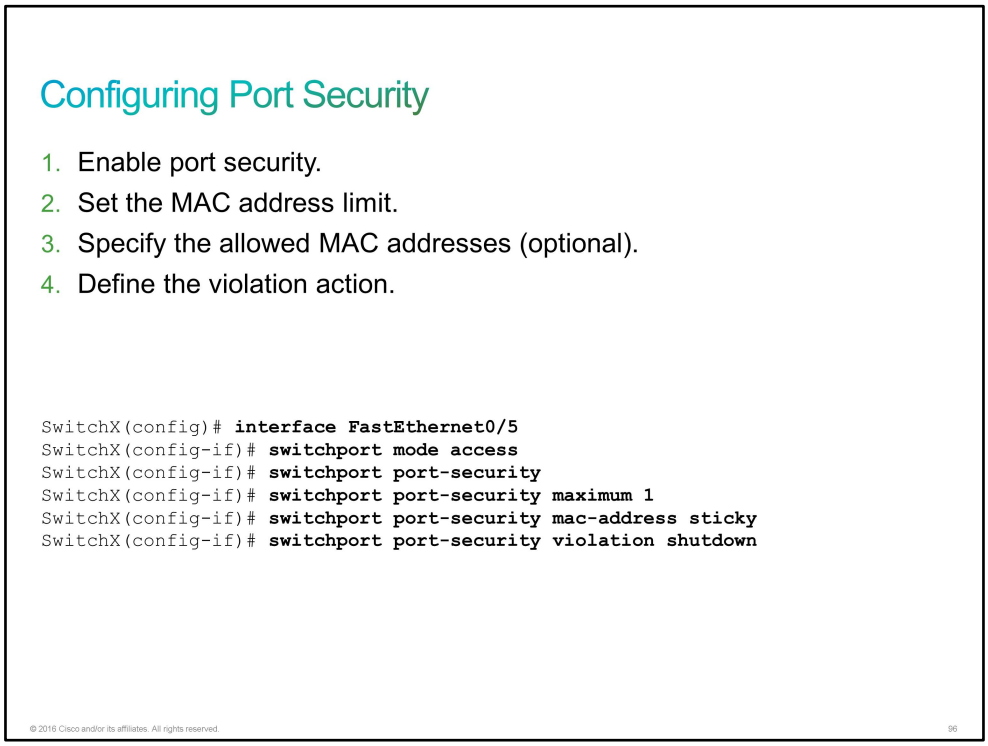
```
SwitchX(config)# interface FastEthernet 0/5
SwitchX(config-if)# shutdown
Sep 20 12:57:28.532: %LINK-5-CHANGED: Interface FastEthernet0/5, changed state to
administratively down
SwitchX(config-if)# no shutdown
Sep 20 12:57:48.186: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
Sep 20 12:57:49.193: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5,
changed state
 to up
```

You can specify how secure MAC address aging occurs on a port by configuring either absolute or inactivity aging. When you configure absolute aging, all the dynamically learned secure addresses age out when the aging time expires. When you configure inactivity aging, the aging time defines the period of inactivity after which all the dynamically learned secure addresses age out. You can also specify the aging time.

You can configure port security only on static access ports or trunk ports. You cannot configure port security on an interface in the default mode (dynamic auto).

Configuring Port Security

To configure port security to limit and identify the MAC addresses of stations that are allowed to access the port, do as follows:



The figure shows how to enable sticky port security on the FastEthernet0/5 port of SwitchX.

Port security limits the number of valid MAC addresses that are allowed on a port. When you assign secure MAC addresses to a secure port, the port does not forward packets with source addresses outside the group of defined addresses.

If you limit the number of secure MAC addresses to one and assign a single MAC address to this port, only the workstation with this particular secure MAC address can successfully connect to this switch port.

If you configure a port as a secure port and the maximum number of secure MAC addresses is reached, a security violation occurs when the MAC address of a workstation that is attempting to access the port is different from any of the identified secure MAC addresses.

Note	Before port security can be activated, you must set the port mode to "access" or "trunk" using the **switchport mode access** | **trunk** command.

Use the **switchport port-security** interface command *without* keywords to enable port security on an interface. Use the **switchport port-security** interface command *with* keywords to configure a secure MAC address, a maximum number of secure MAC addresses, or the violation mode. Use the **no** form of this command to disable port security or to set the parameters to their default states.

You can also configure the maximum number of secure MAC addresses. In this figure, you can see the Cisco IOS command syntax that you use to set the maximum number of MAC addresses to one (**switchport port-security maximum 1**).

You can add secure addresses to the address table after setting the maximum number of secure MAC addresses that are allowed on a port in these ways:

- Manually configure all the addresses (**switchport port-security mac-address 0008.eeee.eeee**).
- Allow the port to dynamically configure all the addresses (**switchport port-security mac-address sticky**).
- Configure several MAC addresses and allow the rest of the addresses to be dynamically configured.

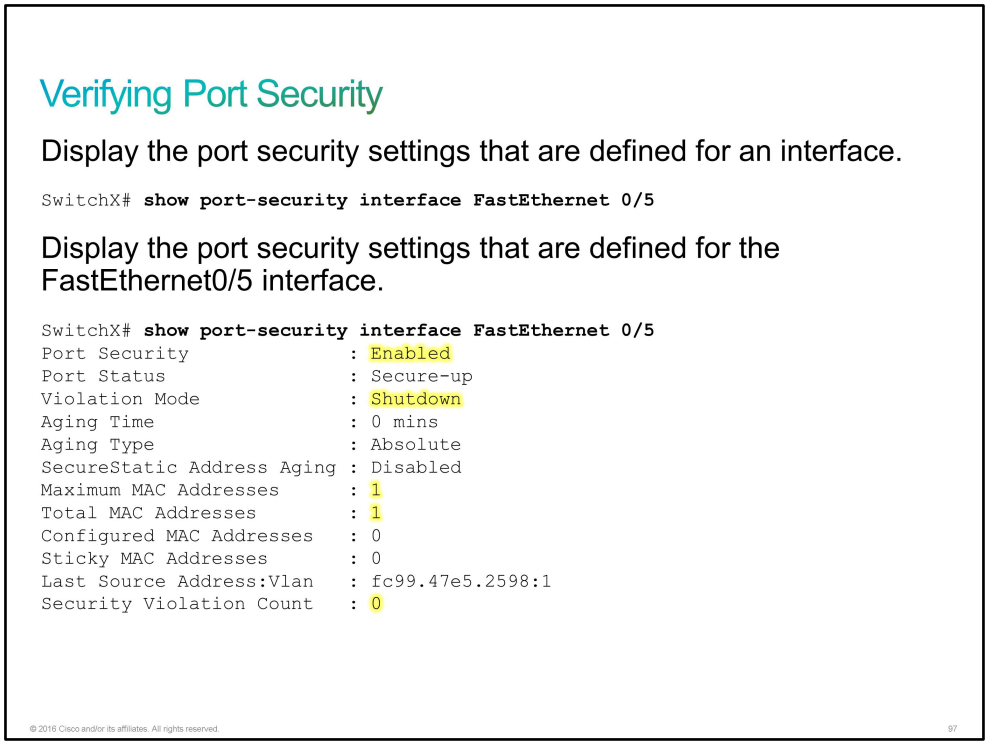
You can configure an interface to convert the dynamic MAC addresses to sticky secure MAC addresses and add them to the running configuration by enabling sticky learning. To enable sticky learning, enter the **switchport port-security mac-address sticky** interface configuration command. When you enter this command, the interface converts all the dynamic secure MAC addresses, including the MAC addresses that the device dynamically learned before you enabled sticky learning, to sticky secure MAC addresses.

The violation mode is set to shutdown (**switchport port-security violation shutdown**). This violation mode, which is the default mode, puts the interface into the error-disabled state immediately and shuts down the entire port.

Note	As mentioned, the other two violation modes are protect and restrict. These modes drop frames from the address that is not allowed, but unlike the shutdown mode, they do not put the interface into the error-disable state.
Note	Port security is disabled by default.

Verifying Port Security

After you have configured port security for your switch, verify that it has been configured correctly.



You must check each interface to verify that you have set the port security correctly. You must also verify that you have configured static MAC addresses correctly. Use the **show port-security interface** privileged EXEC command to display the port security settings that are defined for an interface.

The output displays this information (from the top down):

- Whether the port security feature is enabled
- The violation mode
- The maximum allowed number of secure MAC addresses for each interface
- The number of secure MAC addresses on the interface
- The number of security violations that have occurred

Port Security Verification (Cont.)

Display the port security violation for the FastEthernet0/5 interface.

```
SwitchX#show port-security interface FastEthernet 0/5
Port Security                       : Enabled
Port Status                         : Secure-shutdown
Violation Mode                      : Shutdown
Aging Time                          : 0 mins
Aging Type                          : Absolute
SecureStatic Address Aging          : Disabled
Maximum MAC Addresses               : 1
Total MAC Addresses                 : 1
Configured MAC Addresses            : 0
Sticky MAC Addresses                : 1
Last Source Address:Vlan            : 001a.2fe7.3089:1
Security Violation Count            : 1
```


Port Security Verification (Cont.)

Verify the status of the interface.

```
SwitchX# show interface status
Port      Name            Status      Vlan     Duplex  Speed Type
Fa0/1                     connected   1        a-full  a-100 10/100BaseTX
Fa0/2                     notconnect  1          auto   auto 10/100BaseTX
Fa0/3                     notconnect  1          auto   auto 10/100BaseTX
Fa0/4                     notconnect  1          auto   auto 10/100BaseTX
Fa0/5                     err-disabled 1          auto   auto 10/100BaseTX
<output omitted>
```


When MAC addresses are assigned to a secure port, the port does not forward frames with source MAC addresses outside the group of defined addresses. When a port that is configured with port security receives a frame, the source MAC address of the frame is compared to the list of secure source addresses that were manually configured or autoconfigured (learned) on the port. If a MAC address of a device that is attached to the port differs from the list of secure addresses, the port either shuts down until it is administratively re-enabled (default mode) or drops incoming frames from the insecure host (the **restrict** option). The behavior of the port depends on how it is configured to respond to a security violation.

The output in the figure shows that a security violation has occurred, and the port is in the secure-shutdown state.

Because the port security violation mode is set to **shutdown**, the port with the security violation (source MAC addresses outside the group of defined addresses) goes to the error-disabled state. You receive this notification on the switch:

```
Sep 20 06:44:54.966: %PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/5,
putting Fa0/5 in err-disable state
Sep 20 06:44:54.966: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred,
caused by MAC
address 000c.292b.4c75 on port FastEthernet0/5.
Sep 20 06:44:55.973: %LINEPROTO-5-PPDOWN: Line protocol on Interface FastEthernet0/5,
changed state
 to down
Sep 20 06:44:56.971: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to down
```

To verify the status of the interface, use the **show interface status** command.

To make the interface operational again, you need to disable the interface administratively and then enable it again:

```
SwitchX(config)# interface FastEthernet 0/5
SwitchX(config-if)# shutdown
Sep 20 06:57:28.532: %LINK-5-CHANGED: Interface FastEthernet0/5, changed state to
administratively down
SwitchX(config-if)# no shutdown
Sep 20 06:57:48.186: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
Sep 20 06:57:49.193: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5,
changed state
 to up
```

Port Security Verification (Cont.)

Display the secure MAC addresses for all ports.

```
SwitchX# show port-security address
          Secure Mac Address Table
------------------------------------------------------------------------
Vlan Mac Address     Type                Ports   Remaining Age (mins)
---- -----------     ----                -----   -------------
  1  0008.dddd.eeee    SecureConfigured      Fa0/5        -
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 1024
```

Display the port security settings for the switch.

```
SwitchX# show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
                (Count)       (Count)       (Count)
------------------------------------------------------------------------
      Fa0/5        1             1            0         Shutdown
------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 1024
```


Use the **show port-security address** command to display the secure MAC addresses for all ports. Use the **show port-security** command *without* keywords to display the port security settings for the switch.

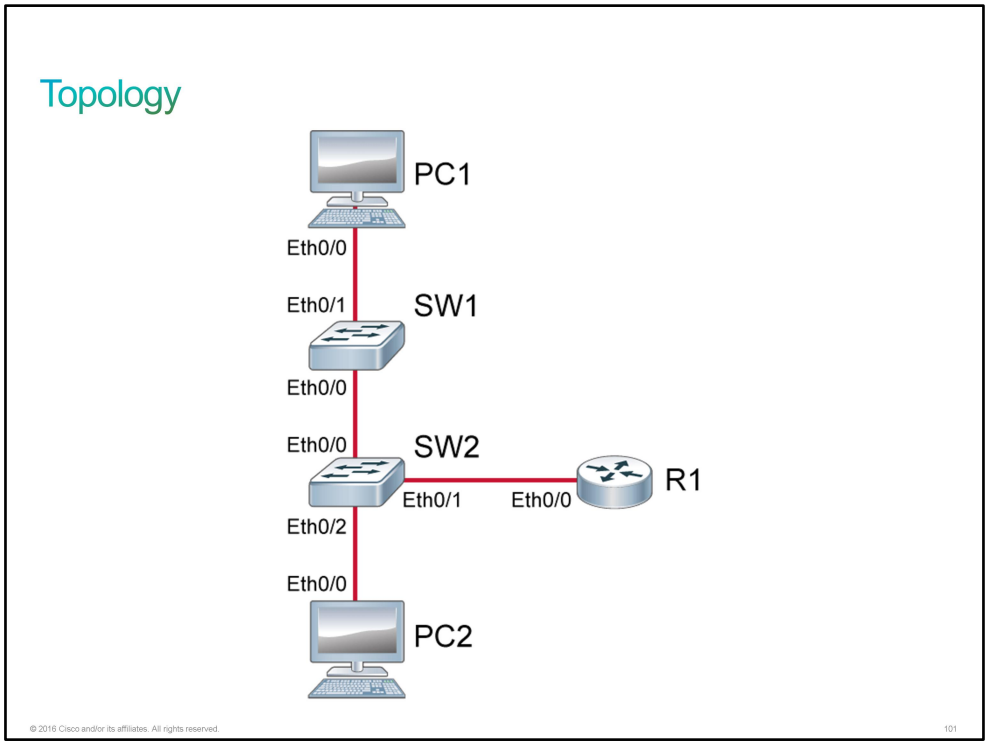
Discovery 24: Configure and Verify Port Security

Introduction

Port security restricts a switch port to a specific set of MAC addresses. You should configure it on all ports that connect to end devices.

In this discovery lab, you will configure and verify port security. You will also set error-disabled port automatic recovery.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1
PC2	Hostname	PC2

Device	Characteristic	Value
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Configure and Verify Port Security

Activity

Step 1 On SW1, configure port security with sticky learning on the Ethernet0/1 interface.

On SW1, enter the following commands:


```
SW1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)# interface Ethernet 0/1
SW1(config-if)# switchport port-security mac-address sticky
SW1(config-if)# switchport mode access
SW1(config-if)# switchport port-security
SW1(config-if)# end
SW1# copy running-config startup-config
Destination filename [startup-config]? <Enter>
Building configuration...
Compressed configuration from 1075 bytes to 720 bytes[OK]
```

When SW1 learns the MAC address of the PC1, you have to save the running configuration so that the learned MAC address stays in the configuration even if the switch reboots.

Step 2 On SW1, verify the port security status.

On SW1, enter the following command:

```
SW1# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
-----------------------------------------------------------------------------
      Et0/1              1            1                  0         Shutdown
-----------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 4096
```

Step 3 On SW1, verify which MAC address is learned on the Ethernet0/1.

On SW1, enter the following command:

```
SW1# show port-security interface Ethernet 0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : aabb.cc00.0200:1
Security Violation Count   : 0
```

MAC address in your output may be different.

Also verify that the same MAC address is listed in the configuration of the SW1.

```
SW1# sh run int e0/1
Building configuration...

Current configuration : 222 bytes
!
interface Ethernet0/1
 description Link to PC1
 switchport mode access
 switchport port-security
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky aabb.cc00.0200
 duplex auto
end
```

Step 4 On PC1, change the MAC address on Ethernet0/0 to eeee.eeee.eeee.

On PC1, enter the following commands:

```
PC1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)# int e 0/0
PC1(config-if)# mac-address eeee.eeee.eeee
PC1(config-if)# end
PC1#
```

When the MAC address on PC1 changes, SW1 will shut down the port toward PC1. You should see the following log messages on the SW1 console:

```
*Jan 29 12:38:18.353: %PM-4-ERR_DISABLE: psecure-violation error detected on
Et0/1, putting Et0/1 in err-disable state
*Jan 29 12:38:18.354: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation
occurred, caused by MAC address eeee.eeee.eeee on port Ethernet0/1.
*Jan 29 12:38:19.356: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/1, changed state to down
*Jan 29 12:38:20.356: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to
down
```

Step 5 On SW1, verify the port security status of Ethernet0/1.

On SW1, enter the following command:

```
SW1# sh port-security int e 0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : eeee.eeee.eeee:1
Security Violation Count   : 1
```

Port Ethernet0/1 is disabled by port security.

Step 6 On PC1, delete the MAC address from Ethernet0/0.

On PC1, enter the following commands:

```
PC1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)# int e 0/0
PC1(config-if)# no mac-address
PC1(config-if)# end
```

PC1 should use the default MAC address as learned by SW1.

```
PC1# sh int e 0/0 | in bia
  Hardware is AmdP2, address is aabb.cc00.0200 (bia aabb.cc00.0200)
```

MAC address in your output may be different.

Step 7 On SW1, verify the port security status of Ethernet0/1.

On SW1, enter the following command:

```
SW1# show port-security int e0/1
Port Security              : Enabled
Port Status                : Secure-shutdown
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : eeee.eeee.eeee:1
Security Violation Count   : 1
```

Port Ethernet0/1 is still disabled by port security.

Step 8 On SW1, enable Ethernet0/1 by shut it down and then bring it back up

On SW1, enter the following command:

```
SW1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)# interface e0/1
SW1(config-if)# shutdown
*Jan 29 12:48:33.655: %LINK-5-CHANGED: Interface Ethernet0/1, changed state to
administratively down
SW1(config-if)# no shutdown
*Jan 29 12:48:39.281: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to
up
*Jan 29 12:48:40.289: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/1, changed state to up
SW1(config-if)# end
```

Step 9 On SW1, verify the port security status of Ethernet0/1 again.

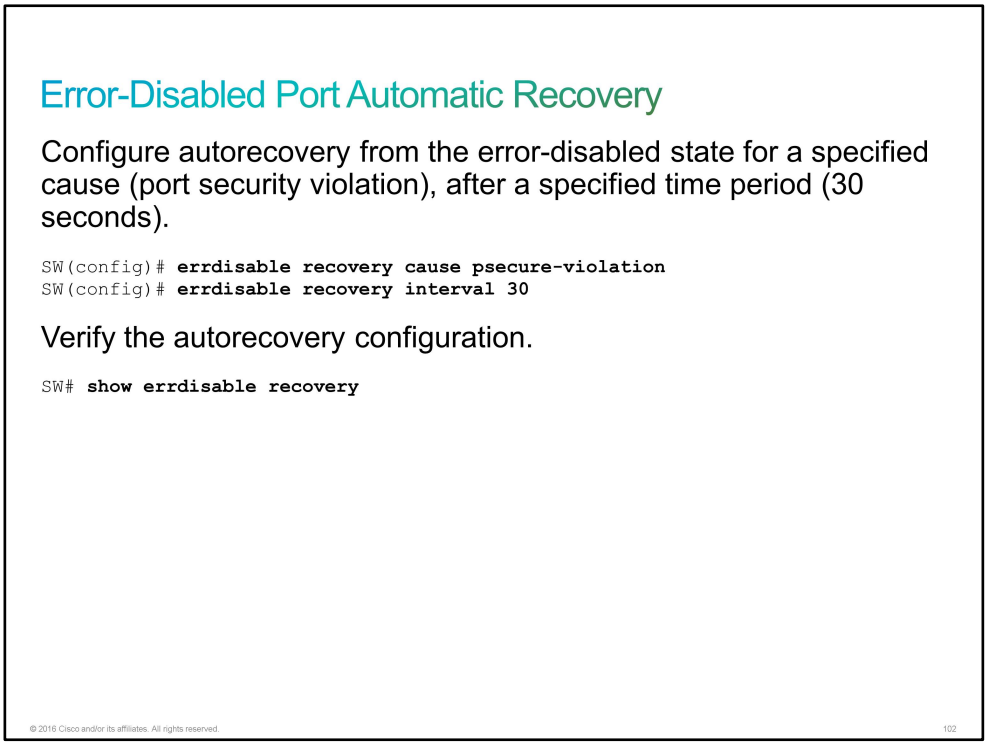
On SW1, enter the following command:

```
SW1# show port-security int e0/1
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Shutdown
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 1
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : aabb.cc00.0200:1
Security Violation Count   : 0
```

Port Ethernet0/1 is operational now.

Error-Disabled Port Automatic Recovery

An error-disabled port will become operational after you shut it down and then bring it back up. To reduce the administrative overhead, an error-disabled port can be automatically re-enabled, after the problem that is causing the error-disabled state is fixed.



Use the **errdisable recovery** command to automatically re-enable the port after a specified time. If the problem that caused the port to change into the error-disabled state is not resolved, the port will stay in the error-disabled state.

```
SW(config)# errdisable recovery cause cause
SW(config)# errdisable recovery interval seconds
```

The default time interval is 300 seconds, and the minimum is 30 seconds.

You can verify where autorecovery is enabled by using the command **show errdisable recovery**. By default, the autorecovery feature is disabled.

Step 10 On SW1, configure the error-disabled recovery cause to **psecure-violation** and set the interval timer to 30 seconds.

On SW1, enter the following commands:

```
SW1# conf t
SW1(config)# errdisable recovery cause psecure-violation
SW1(config)# errdisable recovery interval 30
SW1(config)# end
```

Step 11 On PC1, change the MAC address on Ethernet0/0 to eeee.eeee.eeee again

On PC1, enter the following commands:

```
PC1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)# int e 0/0
PC1(config-if)# mac-address eeee.eeee.eeee
PC1(config-if)# end
PC1#
```

When the MAC address on PC1 changes, SW1 will shut down the port toward PC1.

Step 12 On PC1, delete the MAC address from Ethernet0/0.

On PC1, enter the following commands:

```
PC1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)# int e 0/0
PC1(config-if)# no mac-address
PC1(config-if)# end
```

After 30 seconds, Ethernet0/1 will be recovered. You will see following log message on the SW1 console:

```
*Jan 29 13:02:23.001: %PM-4-ERR_RECOVER: Attempting to recover from psecure-
violation err-disable state on Et0/1
*Jan 29 13:02:25.001: %LINK-3-UPDOWN: Interface Ethernet0/1, changed state to
up
*Jan 29 13:02:26.002: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/1, changed state to up
```

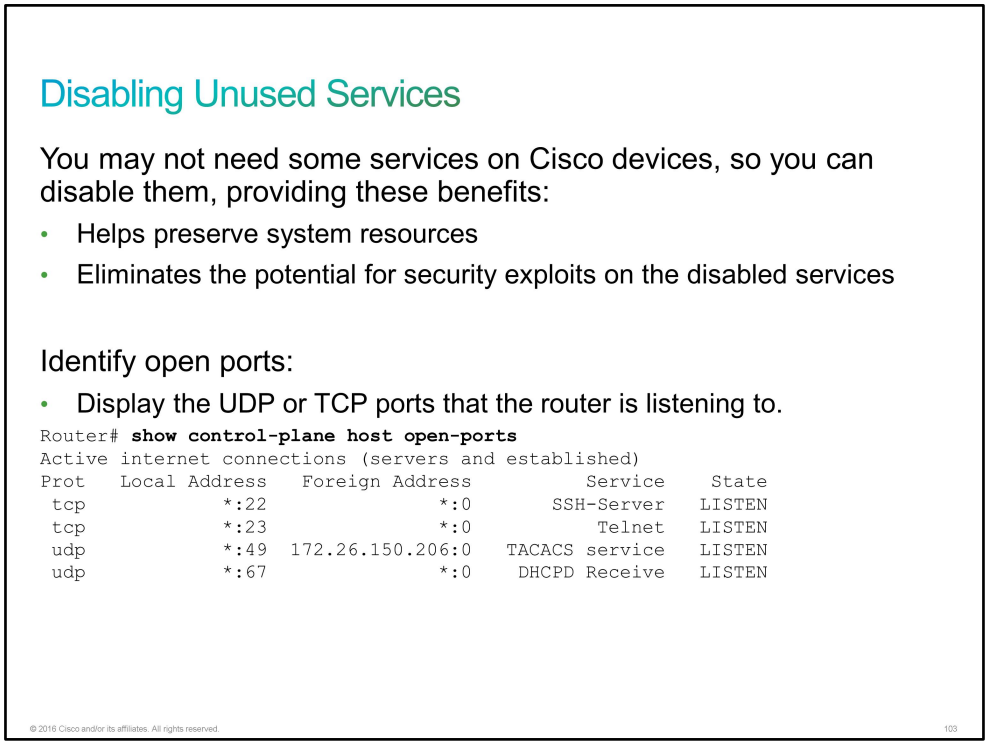
Step 13 From PC1, ping R1 (10.10.1.1) to verify whether Ethernet0/1 on SW1 is operational.

On PC1, enter the following command:

```
PC1# ping 10.10.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/201/1004 ms
```

Disabling Unused Services

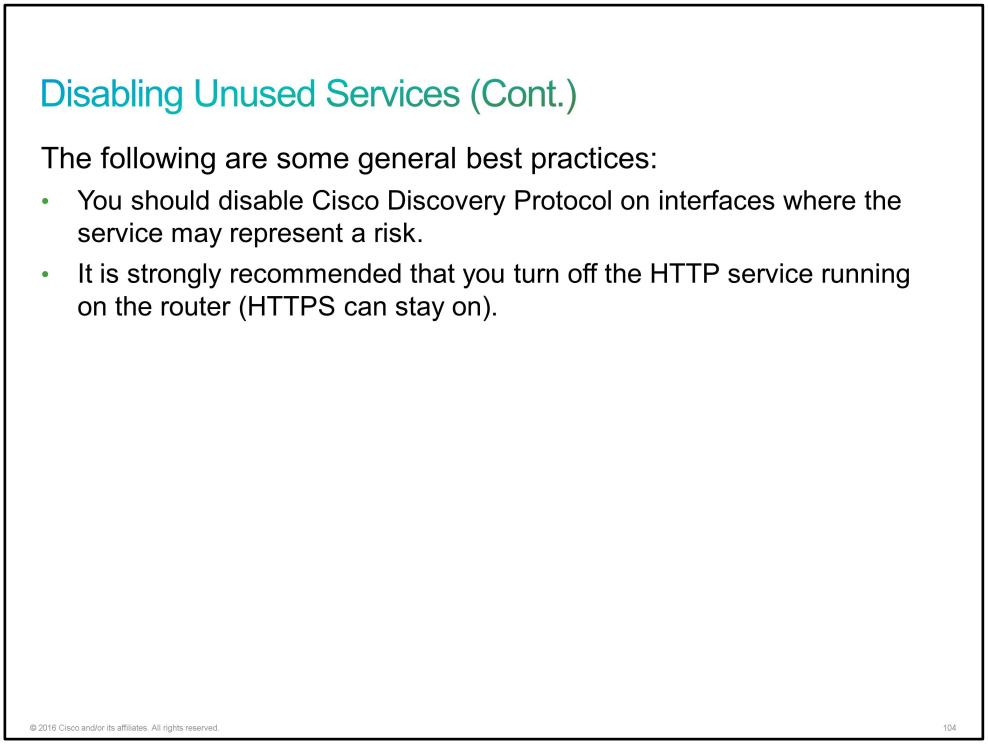
To facilitate deployment, Cisco routers and switches start with a list of services that are turned on and considered to be appropriate for most network environments. However, because not all networks have the same requirements, some of these services may not be needed. Disabling these unnecessary services has two benefits: it helps preserve system resources, and it eliminates the potential for security exploits on the unneeded services.



The general best practice is to identify open ports. Use the **show control-plane host open-ports** command to see which UDP or TCP ports the router is listening to and to determine which services need to be disabled.

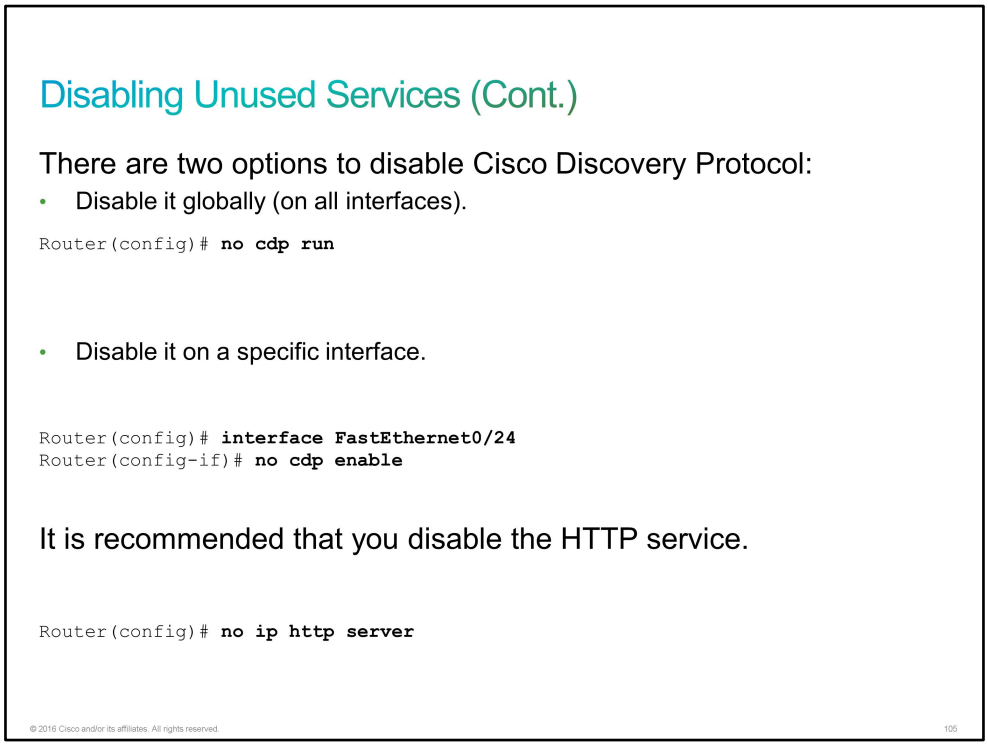
In the example, services that are enabled on the router are SSH, Telnet, TACACS, and DHCP.

Note	As an alternative, Cisco IOS Software provides the AutoSecure function that helps disable these unnecessary services while enabling other security services.



Disable Cisco Discovery Protocol on the interfaces where the service may represent a risk. Examples are external interfaces, such as those at the Internet edge, and data-only ports at the campus and branch access. Cisco Discovery Protocol is enabled by default in Cisco IOS Software Release 15.0 and later.

You can access Cisco routers via a web page, but it is strongly recommended that you turn off the HTTP service that is running on the router.



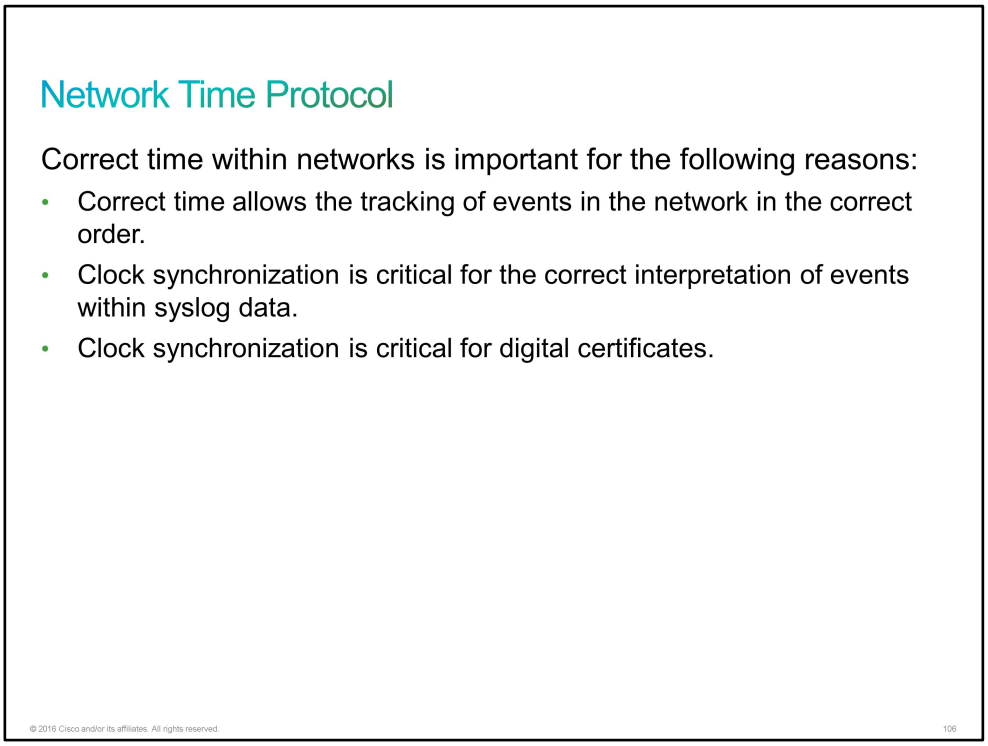
If you prefer not to use the Cisco Discovery Protocol device discovery capability, you can disable it with the **no cdp run** global configuration command. To re-enable Cisco Discovery Protocol after disabling it, use the **cdp run** command in the global configuration mode.

Cisco Discovery Protocol is enabled by default on all supported interfaces to send and receive Cisco Discovery Protocol information. Cisco Discovery Protocol is not on by default on Frame Relay interfaces. You can disable Cisco Discovery Protocol on an interface that supports it with the **no cdp enable** interface configuration command. To re-enable Cisco Discovery Protocol on an interface after disabling it, use the **cdp enable** command in the interface configuration mode.

It is strongly recommended that you turn off the HTTP service that is running on the router. You can use the **no ip http server** global configuration command to disable it. To re-enable the HTTP service after disabling it, use the **ip http server** command in global configuration mode.

Network Time Protocol

Networks use NTP to synchronize the clocks of various devices across a network. Clock synchronization within a network is critical for digital certificates and for the correct interpretation of events within syslog data. A secure method of providing clocking for the network is for network administrators to implement their own private network master clocks that are synchronized to UTC-using satellite or radio. However, if network administrators do not wish to implement their own master clocks because of cost or other reasons, other clock sources are available on the Internet.



Network Time Protocol (Cont.)

NTP provides time synchronization between network devices.

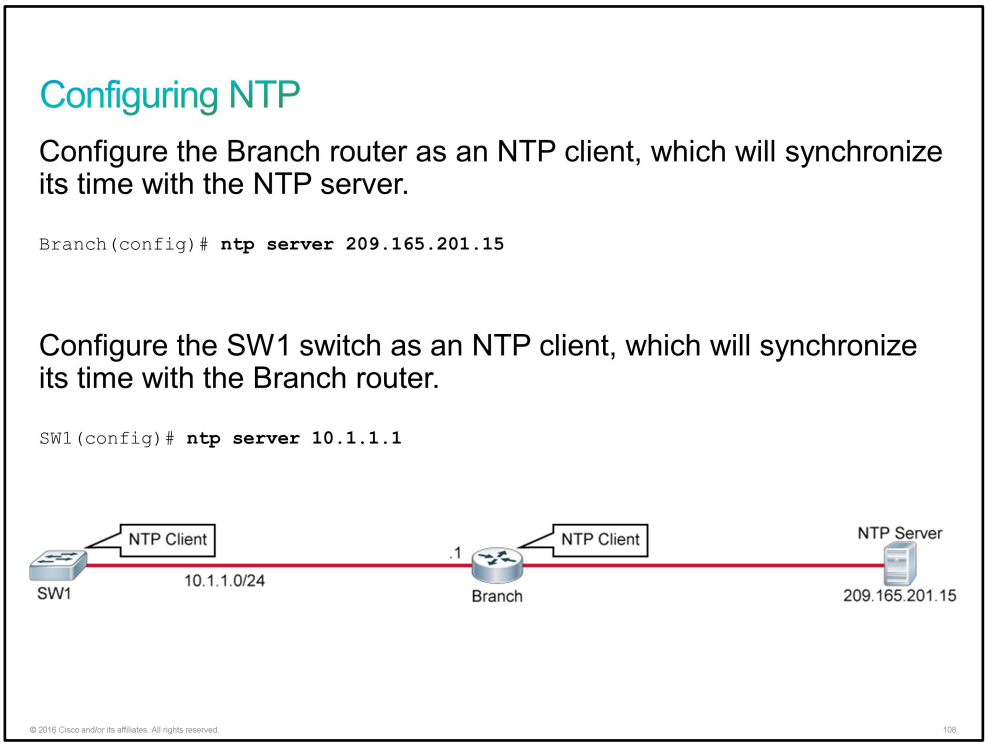
- NTP can get the correct time from an internal or external time source:
  - Local master clock
  - Master clock on the Internet
  - GPS—global positioning system or atomic clock
- A router can act as an NTP server and client. Other devices (NTP clients) synchronize time with the router (NTP server).

NTP is a protocol for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks. NTP allows routers on your 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 will have more consistent time settings.

You can configure a router as an NTP server, to which other devices (NTP clients) synchronize their time settings.

Configuring NTP

To configure NTP on Cisco devices, use following commands:



The figure shows an example configuration scenario. Both the router Branch and switch SW1 are configured as NTP clients using the **ntp server** *ip-address* global configuration command. The IP address of the NTP server is configured.

A Cisco IOS device acting as an NTP client will also respond to received time requests. This factor enables SW1 to sync directly with the router Branch and optimize traffic flows. Alternatively, you could configure the switch SW1 to sync with an external NTP server as well.

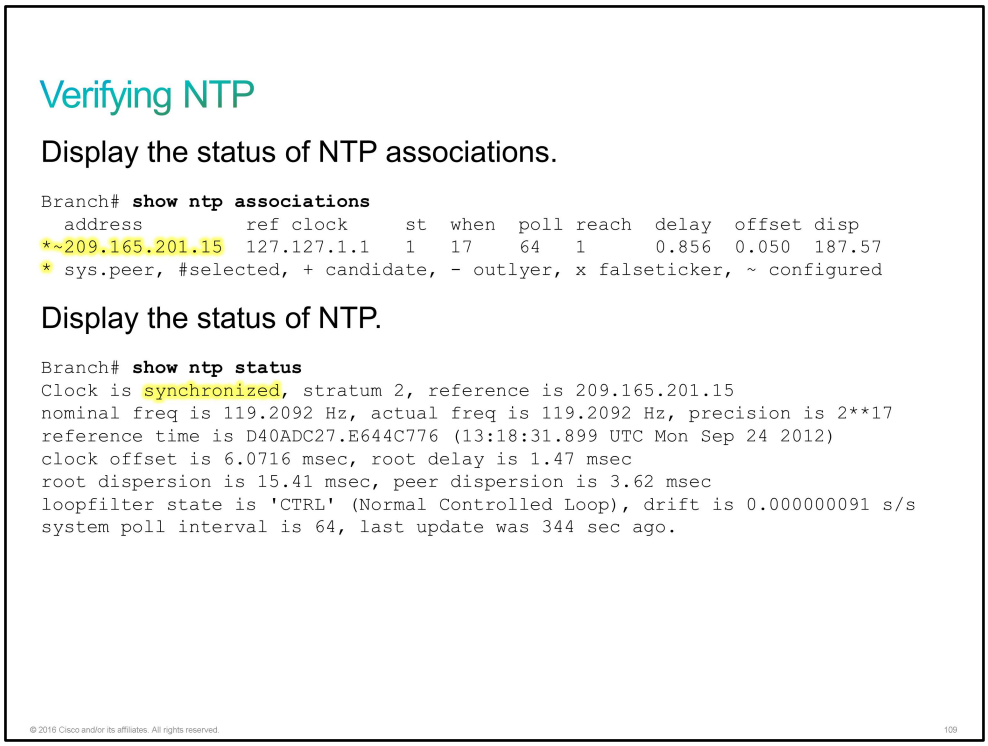
Cisco IOS devices can also act as NTP servers. To configure Cisco IOS Software as an NTP master clock to which peers synchronize themselves, use the **ntp master** command in the global configuration mode: **ntp master** [*stratum*]

Note	Use this command with caution. You can easily override valid time sources using this command, especially if a low stratum number is configured. Configuring multiple devices in the same network with the **ntp master** command can cause instability in keeping time if the devices do not agree on the time.

The *stratum* value is a number from 1 to 15. The lowest stratum value indicates a higher NTP priority. It also indicates the NTP stratum number that the system will claim.

Verifying NTP

To verify NTP on Cisco devices, use the following commands:



To display the status of NTP associations, use the **show ntp associations** command in the privileged EXEC mode.

The output displays these significant fields:

- ***:** The peer that is synchronized to this peer
- **~:** The peer that is is statically configured
- **address:** The address of the peer
- **st:** The stratum setting for the peer

Note	It may take several minutes for an NTP client to synchronize with the NTP server.

To display the status of NTP, use the **show ntp status** command in the user EXEC mode.

The output displays these significant fields:

- **synchronized:** The system that is synchronized to an NTP peer
- **stratum:** The NTP stratum of this system
- **reference:** The address of the peer to which a clock is synchronized

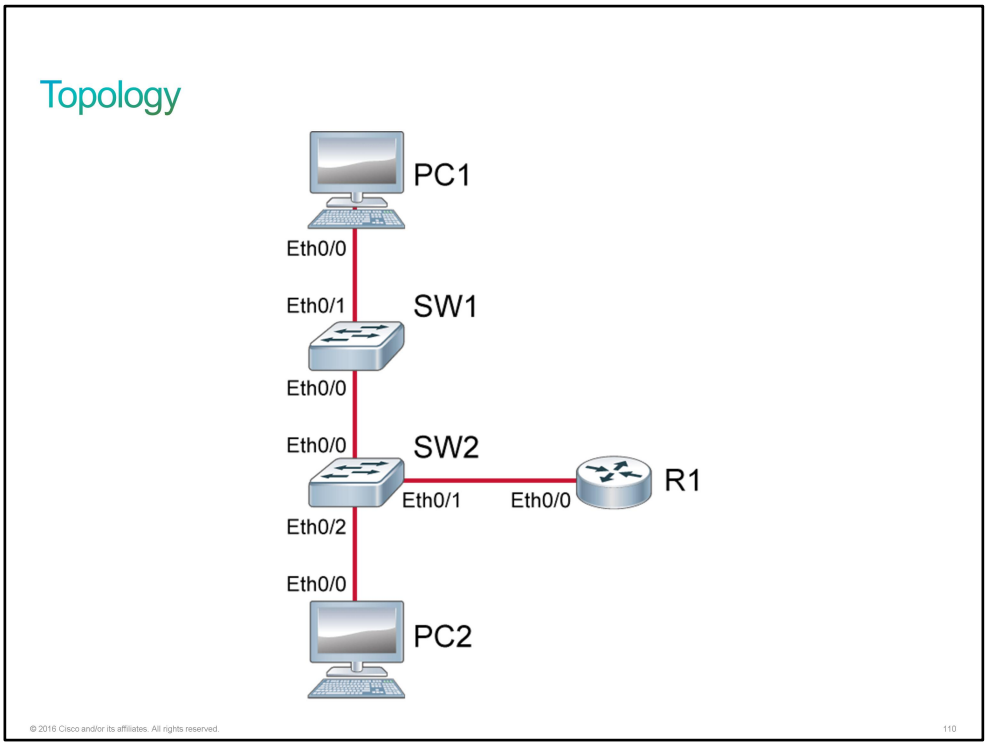
Discovery 25: Configure and Verify NTP

Introduction

Network devices generate syslog messages to convey important information about events within the network. Syslog messages have time stamps that are associated with them. For these time stamps to be of value for security analysis, the clocks on all the network devices must be in sync. NTP is the preferred method to achieve synchronisation.

This discovery lab will guide you through configuring and verifying NTP services on Cisco IOS routers. The lab is prepared as depicted in the topology diagram and the connectivity table.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
PC1	Hostname	PC1
PC1	IP address	10.10.1.10/24
PC1	Default gateway	10.10.1.1

Device	Characteristic	Value
PC2	Hostname	PC2
PC2	IP address	10.10.1.20/24
PC2	Default gateway	10.10.1.1
SW1	Hostname	SW1
SW1	VLAN 1 IP address	10.10.1.2/24
SW1	Default gateway	10.10.1.1
SW1	Ethernet0/0 description	Link to SW2
SW1	Ethernet0/1 description	Link to PC1
SW2	Hostname	SW2
SW2	VLAN 1 IP address	10.10.1.3/24
SW2	Default gateway	10.10.1.1
SW2	Ethernet0/0 description	Link to SW1
SW2	Ethernet0/1 description	Link to R1
SW2	Ethernet0/2 description	Link to PC2
R1	Hostname	R1
R1	Ethernet0/0 description	Link to SW2
R1	Ethernet0/0 IP address	10.10.1.1/24
R1	Loopback 0 IP	10.10.3.1/24

PCs in the virtual lab environment are simulated as routers, so you should use Cisco IOS commands to configure them or make verifications.

Task 1: Configure and Verify NTP

Activity

Step 1 Start by reviewing the clocks on SW1, SW2 and R1. You will find that in the emulated lab environment, the clocks are actually synchronized by default.

On SW1, enter the following command:

```
SW1# sh clock
*00:46:09.379 PST Tue Nov 24 2015
```

On SW2, enter the following command:

```
SW2# sh clock
*00:46:11.251 PST Tue Nov 24 2015
```

On R1, enter the following command:

```
R1# sh clock
*00:46:12.604 PST Tue Nov 24 2015
```

The difference in time is only the time it took you to switch from one console to the next and enter the **show clock** command.

Of course, the times that this output and the following output examples depict will differ from what you can see in the lab environment.

Step 2 Access the console of R1 and configure it as an NTP server by enabling the master clock status.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ntp master
R1(config)# end
R1#
```

Step 3 Configure SW2 to use R1 (10.10.1.1) as its NTP server.

On SW2, enter the following commands:

```
SW2# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW2(config)# ntp server 10.10.1.1
SW2(config)# end
SW2#
```

Step 4 Display the current NTP associations and NTP status on SW2.

On SW2, enter the following commands:


```
SW2# show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~10.10.1.1       127.127.1.1      8     49     64     1  0.000   0.000 189.47
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
SW2# show ntp status
Clock is synchronized, stratum 9, reference is 10.10.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**10
ntp uptime is 6600 (1/100 of seconds), resolution is 4000
reference time is D9FEA2DD.74FDF4F8 (00:48:29.457 PST Tue Nov 24 2015)
clock offset is 0.0000 msec, root delay is 1.00 msec
root dispersion is 4381.02 msec, peer dispersion is 189.47 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s
system poll interval is 64, last update was 64 sec ago.
```

Step 5 One at a time, access the consoles of SW2 and R1 and display their clocks. They should be synchronized. The difference in time is due to the time that you spend switching between consoles and entering the command.

On SW2, enter the following command:

```
SW2# show clock
00:50:58.437 PST Tue Nov 24 2015
```

On R1, enter the following command:

```
R1# sh clock
00:51:00.213 PST Tue Nov 24 2015
```

Step 6 On R1, configure CET time zone.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# clock timezone CET 2
Nov 24 08:51:57.876: %SYS-6-CLOCKUPDATE: System clock has been updated from
00:51:57 PST Tue Nov 24 2015 to 10:51:57 CET Tue Nov 24 2015, configured from
console by console.
R1(config)# end
R1#
```

Step 7 Display the current time on R1 and observe that the time zone has changed.

On R1, enter the following command:

```
R1# show clock
10:53:05.222 CET Tue Nov 24 2015
```

This is the end of the discovery lab.

Challenge

1. Which command can you use to help disable multiple ports in a switch?

   A. **interface range**
   B. **interface**
   C. **shutdown range**
   D. **interface range shutdown**

2. Which of the following converts dynamically learned addresses into secure addresses by modifying the running configuration on the fly?

   A. Static Learning
   B. Dynamic Learning
   C. A combination of static and dynamic learning
   D. Sticky Learning

3. You want an interface to error-disable if traffic on the interface violates port-security parameters. Which of the following would you use?

   A. **switchport port-security shutdown**
   B. **switchport port-security violations on**
   C. **switchport port-security violation err-disabled**
   D. **switchport port-security violation shutdown**

4. Check the following command output. What state is the port in?

   SwitchX# show port-security interface FastEthernet 0/5
   Port Security : Enabled
   Port Status : Secure-up
   Violation Mode : Shutdown
   Aging Time : 0 mins
   Aging Type : Absolute
   SecureStatic Address Aging : Disabled
   Maximum MAC Addresses : 1
   Total MAC Addresses : 1
   Configured MAC Addresses : 0
   Sticky MAC Addresses : 0
   Last Source Address:Vlan : fc99.47e5.2598:1
   Security Violation Count : 0

   A. Forwarding
   B. Err-Disabled
   C. Shutdown
   D. Listening

5. Which of the following commands displays the open ports on a router?

   A. **show open-ports**
   B. **show control-plane host**
   C. **show control-plane host open-ports**
   D. **show ports open host**

6. Why is clock synchronization between network devices important?

   A. To ensure that routing protocols on devices can communicate with each other.
   B. To ensure traffic transiting network devices, do not get dropped.
   C. To ensure no security breaches happen due to an explot called 'clock attack'
   D. To ensure the correct interpretation of events within syslog data.

7. Which command will you use to configure a device as an NTP client?

   A. **ntp client**
   B. **ntp server**
   C. **ntp master**
   D. **ntp source**

Answer Key

Challenge

1. A
2. D
3. D
4. A
5. C
6. D
7. B

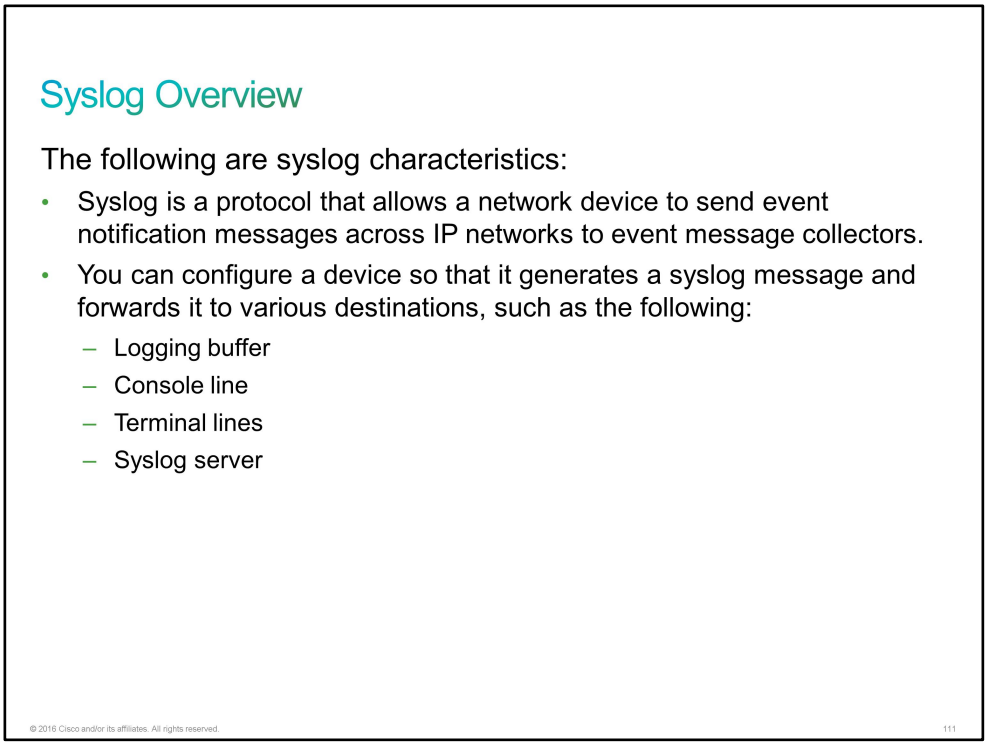
Lesson 3: Configuring System Message Logging

Introduction

Your boss sends you to your customer to enable system logging. You will need to explain to the customer how to configure and verify syslog.

Syslog Overview

Syslog is a protocol that allows a machine to send event notification messages across IP networks to event message collectors. By default, a network device sends the output from system messages and debug privileged EXEC commands to a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a syslog server, depending on your configuration. The process also sends messages to the console. Logging services provide a means to gather logging information for monitoring and troubleshooting, to select the type of logging information that is captured, and to specify the destinations of captured syslog messages.



You can set the severity level of the messages to control the type of messages that the consoles display and each of the destinations. You can time-stamp log messages or set the syslog source address to enhance real-time debugging and management.

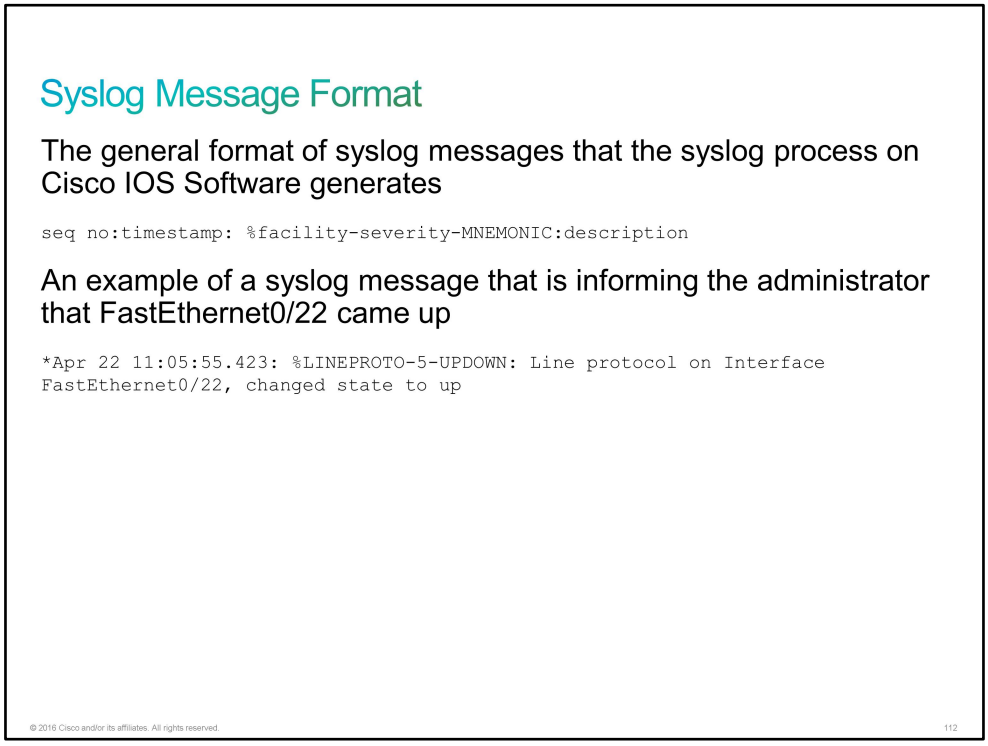
You can access logged system messages by using the device CLI or by saving them to a correctly configured syslog server. The switch or router software saves syslog messages in an internal buffer.

You can 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.

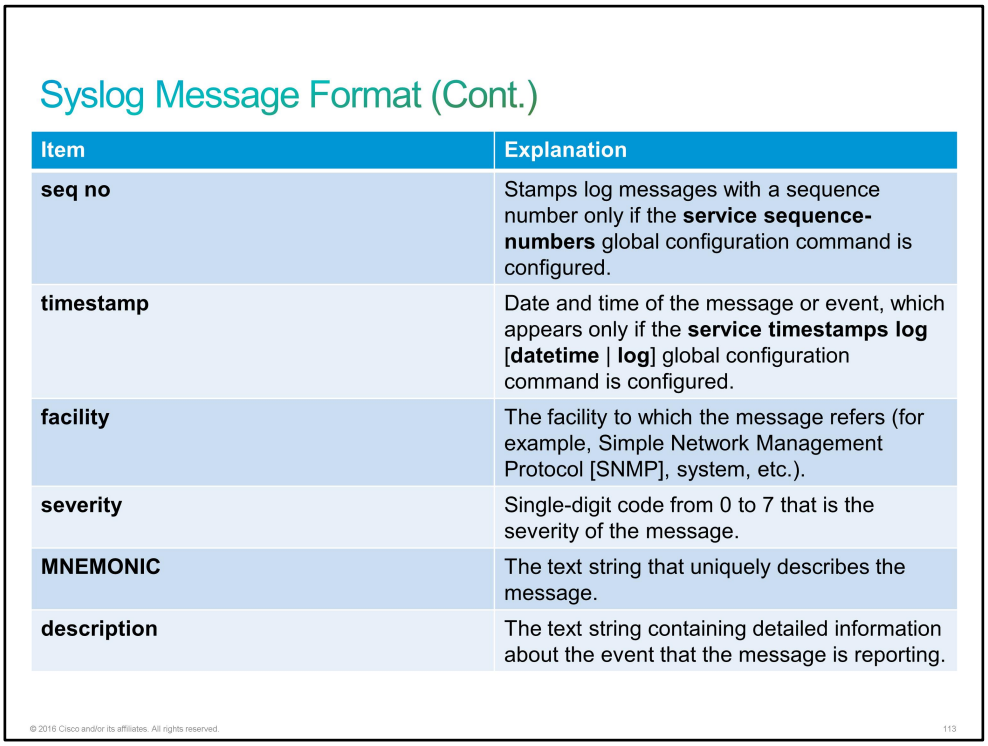
Syslog Message Format

The following is the general format of syslog messages that the syslog process on the Cisco IOS Software generates by default:

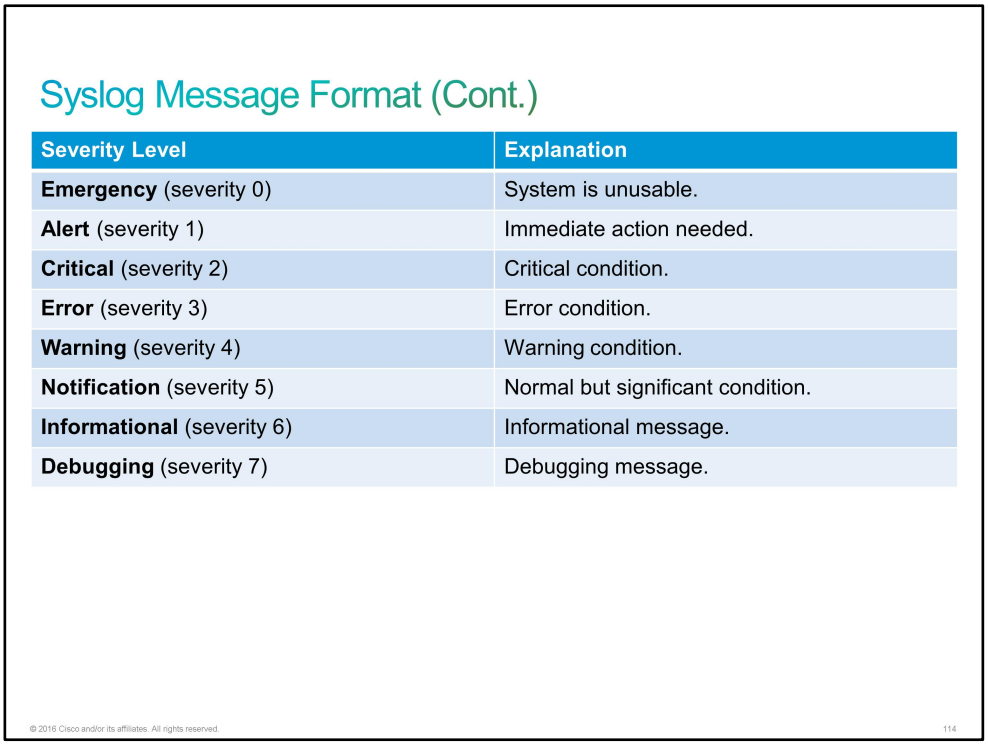
```
seq no:timestamp: %facility-severity-MNEMONIC:description
```



This table explains the items that Cisco IOS Software syslog message contains.



This table explains the eight message severity levels from the most severe level to the least severe level.

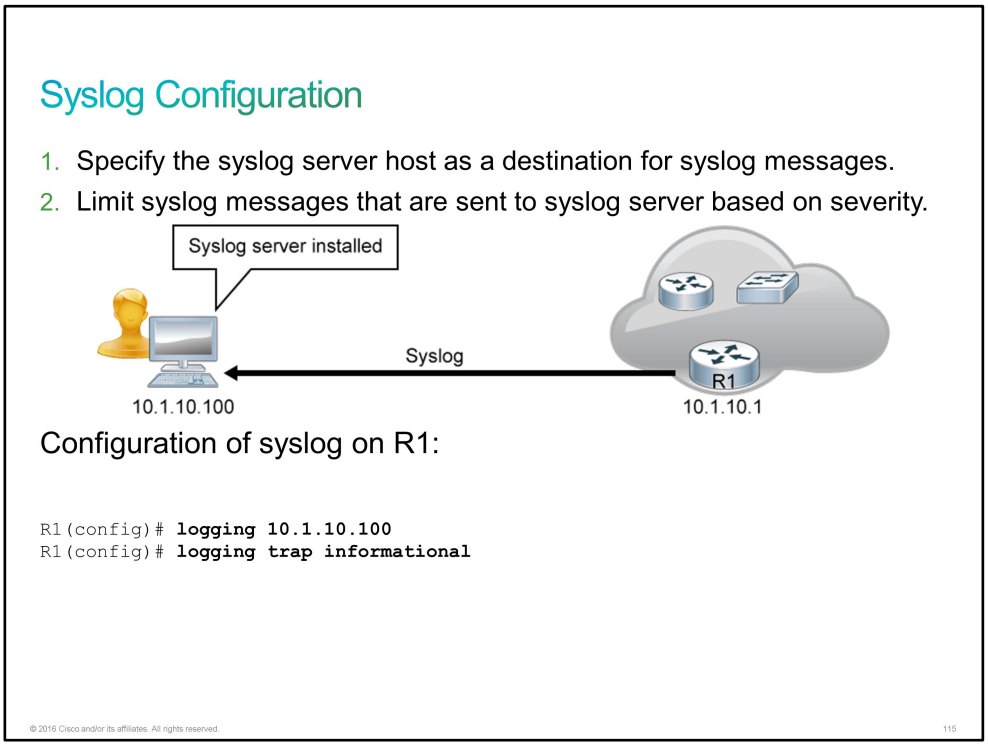


If severity level 0 is configured, it means that only emergency-level messages will be displayed. For example, if severity level 4 is configured, all messages with severity levels up to 4 will be displayed (**Emergency**, **Alert**, **Critical**, **Error**, and **Warning**).

The highest severity level is level 7, which is the debugging-level message. Much information can be displayed at this level, and it can even hamper the performance of your network. Use it with caution.

Syslog Configuration

To implement a syslog configuration, specify a syslog server host as a destination for syslog messages and limit the syslog messages that are sent to the syslog server based on the severity.



Configuration of syslog is based on the commands that the following table describes.

Command	Description
logging {*hostname* | *ip-address*}	Identifies a syslog server host to receive logging messages.
logging trap *severity*	Limits the syslog messages that are sent to the syslog server. It limits the messages based on severity.

The figure shows configurations for logging syslog messages to a syslog server with the IP address 10.1.10.100, where you can observe syslog messages.

The **logging** command identifies a syslog server host to receive logging messages. By issuing this command more than once, you build a list of syslog servers that receive logging messages. You can limit the syslog messages that are sent to the syslog server based on severity, using the **logging trap** command.

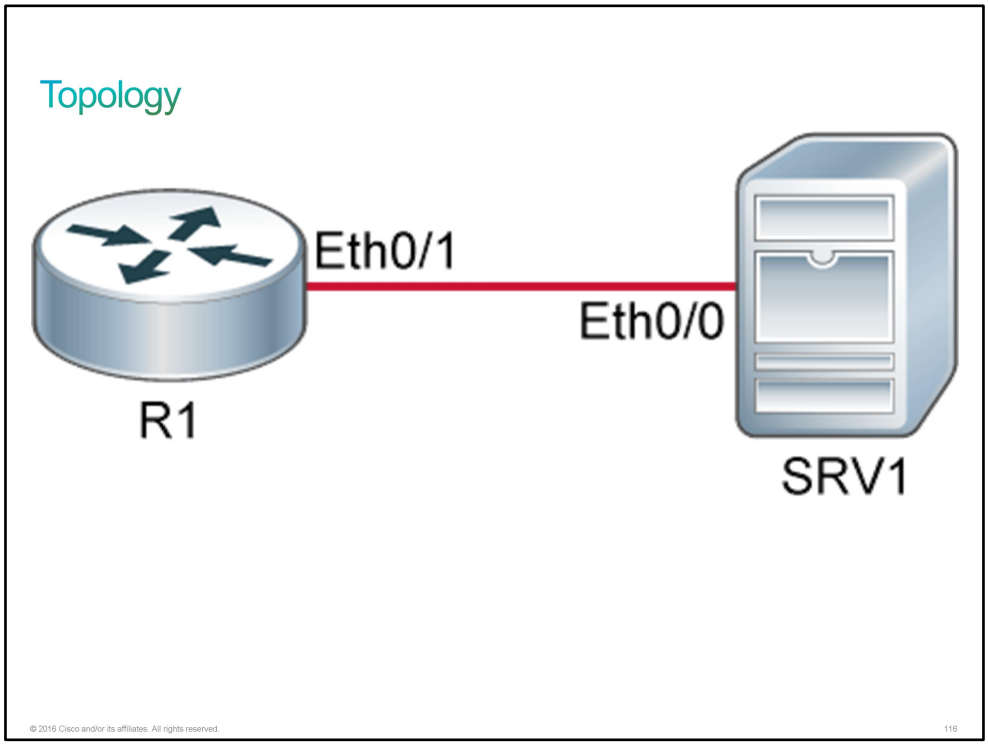
Discovery 26: Configure Syslog

Introduction

The objective of this discovery lab is to provide you with some experience with the syntax of basic syslog configuration to facilitate the management of Cisco IOS devices. This lab is prepared with the router and server that are represented in the topology diagram and the connectivity table. The devices have their basic configurations in place, including hostnames and IP addresses.

In the discovery lab, you will configure the syslog server address of the router and set the severity threshold for messages that are forwarded to the server. You will also use show commands to verify the syslog configuration and examine the syslog messages in the local logging buffer of the router.

Topology



Job Aid

Device Information

Device Information Table

Device	Characteristic	Value
SRV1	Hostname	SRV1
SRV1	IP address	10.1.1.10/24

Device	Characteristic	Value
R1	Hostname	R1
R1	Ethernet0/1 description	Link to SRV1
R1	Ethernet0/1 IP address	10.1.1.1/24

SRV in the virtual lab environment is simulated as router, so you should use Cisco IOS commands to configure it or make verifications.

Task 1: Configure Syslog

Activity

Step 1 Access the R1 console. Define SRV1 (10.1.1.10) as the R1 syslog server.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# logging 10.1.1.10
```

The most commonly used commands are abbreviated in this guided discovery. For example, you use **conf t** for **configure terminal**. If there is any confusion, you can perform tab completion of commands to see the full commands during the discovery execution. For example, **conf<tab> t<tab>** would expand to **configure terminal**.

Step 2 Set "informational" as the threshold for the minimum severity level for messages to send to syslog servers.

On R1, enter the following commands:

```
R1(config)# logging trap informational
R1(config)# end
R1#
*Dec  1 08:04:49.998: %SYS-5-CONFIG_I: Configured from console by console
R1#
*Dec  1 08:04:51.027: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 10.1.1.10
port 514 started - CLI initiated
```

There is a syslog message that is displayed to the console indicating that logging has started to the server at 10.1.1.10. The first message is of severity 5 (Notification), and the second message is of severity 6 (Informational). Setting the threshold to "informational" means that messages of severity 0 through 6 will be forwarded to the syslog server. Both these messages are forwarded.

Step 3 Enter the **show logging** command to display the syslog status and the local logging buffer.

On R1, enter the following command:


```
R1# show logging
Syslog logging: enabled (0 messages dropped, 3 messages rate-limited, 0
flushes, 0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.



No Inactive Message Discriminator.


    Console logging: level debugging, 32 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 32 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 35 message lines logged
        Logging to 10.1.1.10  (udp port 514, audit disabled,
              link up),
              2 message lines logged,
              0 message lines rate-limited,
              0 message lines dropped-by-MD,
              xml disabled, sequence number disabled
              filtering disabled
        Logging Source-Interface:       VRF Name:

Log Buffer (4096 bytes):

*Dec  1 07:49:59.944: %IFMGR-7-NO_IFINDEX_FILE: Unable to open nvram:/ifIndex-
table No such file or directory
<... output omitted ...>
*Dec  1 08:04:49.998: %SYS-5-CONFIG_I: Configured from console by console
*Dec  1 08:04:51.027: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 10.1.1.10
port 514 started - CLI initiated
```

The output indicates that R1 is now sending syslog messages to 10.1.1.10 with the minimum severity threshold set to "informational." The output also indicates that two messages have been sent to the syslog server. Syslog uses UDP for transport and is inherently not reliable. If these two messages are lost somewhere in the transport path, there is no mechanism to recognize the lost message or to request a retransmission.

There is a local logging buffer. It is in its default state, with a severity threshold of "debugging" (Severity 7) and sized at 4096 bytes. In the sample transcript, 32 messages have been logged in the local buffer. The end of the **show logging** command output displays the contents of the buffer. At this point in the discovery, the buffer is mostly filled with the messages that were produced when R1 booted. At the end of the buffer, however, are the two syslog messages that were produced as a result of the syslog configuration activity.

Step 4 The output of the **show logging** command documents that two messages were sent to 10.1.1.10. Initiate some activity that will generate more syslog messages on R1. Enter the configuration mode, enable the interface Ethernet0/3, then disable the interface back down, and leave configuration mode.

On R1, enter the following commands:

```
R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# int e 0/3
R1(config-if)# no shut
R1(config-if)#
*Dec  1 08:10:54.261: %LINK-3-UPDOWN: Interface Ethernet0/3, changed state to
up
*Dec  1 08:10:55.265: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/3, changed state to up
R1(config-if)# shutdown
R1(config-if)#
*Dec  1 08:11:02.057: %LINK-5-CHANGED: Interface Ethernet0/3, changed state to
administratively down
*Dec  1 08:11:03.061: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/3, changed state to down
R1(config-if)# end
R1#
*Dec  1 08:11:06.063: %SYS-5-CONFIG_I: Configured from console by console
R1#
```

This sample activity caused the generation of five syslog messages.

Step 5 Display the logging status and the local logging buffer.

On R1, enter the following command:

```
R1# show logging
<... output omitted ...>
    Console logging: level debugging, 37 messages logged, xml disabled,
                     filtering disabled
    Monitor logging: level debugging, 0 messages logged, xml disabled,
                     filtering disabled
    Buffer logging:  level debugging, 37 messages logged, xml disabled,
                    filtering disabled
    Exception Logging: size (4096 bytes)
    Count and timestamp logging messages: disabled
    Persistent logging: disabled

No active filter modules.

    Trap logging: level informational, 40 message lines logged
        Logging to 10.1.1.10  (udp port 514, audit disabled,
              link up),
<... output omitted ...>
*Dec  1 08:10:54.261: %LINK-3-UPDOWN: Interface Ethernet0/3, changed state to
up
*Dec  1 08:10:55.265: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/3, changed state to up
*Dec  1 08:11:02.057: %LINK-5-CHANGED: Interface Ethernet0/3, changed state to
administratively down
*Dec  1 08:11:03.061: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0/3, changed state to down
*Dec  1 08:11:06.063: %SYS-5-CONFIG_I: Configured from console by console
```

Additional messages were logged to 10.1.1.10.

The five syslog messages that were produced in response to your previous activity are at the end of the local logging buffer.

This is the end of the discovery lab.

Challenge

1. How can you access the syslog of a router?

   A. On a remote router that is receiving the syslog
   B. On a router that is placed between the router that is sending the syslog messages and a syslog server that is receiving the log messages
   C. On a syslog server that is receiving the syslog
   D. On a remote switch that is receiving the syslog

2. This is the format of a syslog message:

   seq no: timestamp: %facility-severity-MNEMONIC:description
   What is the MNEMONIC?

   A. The text string or code that uniquely describes the message.
   B. The text that is a full sentence-like description of the event.
   C. A way of remembering previous events.
   D. A number that is part event number and part MAC Address.

3. You want to control the severity of the event that determines when a syslog should be sent. Which command do you use?

   A. **logging *{hostname | ip address}***
   B. **logging trap *severity***
   C. **logging severity**
   D. **logging level *severity***

4. Which of the following severity levels is used when a system is unusable?

   A. Emergency
   B. Alert
   C. Critical
   D. Error

5. Severity level "Emergency" has which number assigned to it?

   A. 0
   B. 1
   C. 6
   D. 7

6. If you enter the l**ogging {*hostname | ip address*}** command more than once, the server that is specified in the command entered last will be the only one used. True or False?

   A. True
   B. False

7. What does the MNEMONIC "%LINEPROTO-5-UPDOWN" in a syslog message signify about the event being logged?

   A. An interface has changed its up/down status.
   B. An interface has been configured with an IP Address.
   C. A Routing protocol has just developed adjacency with a peer router
   D. The Router is either shutting down or has just booted up.

Answer Key

Challenge

1. C
2. A
3. B
4. A
5. A
6. B
7. A

Lesson 4: Managing Cisco Devices

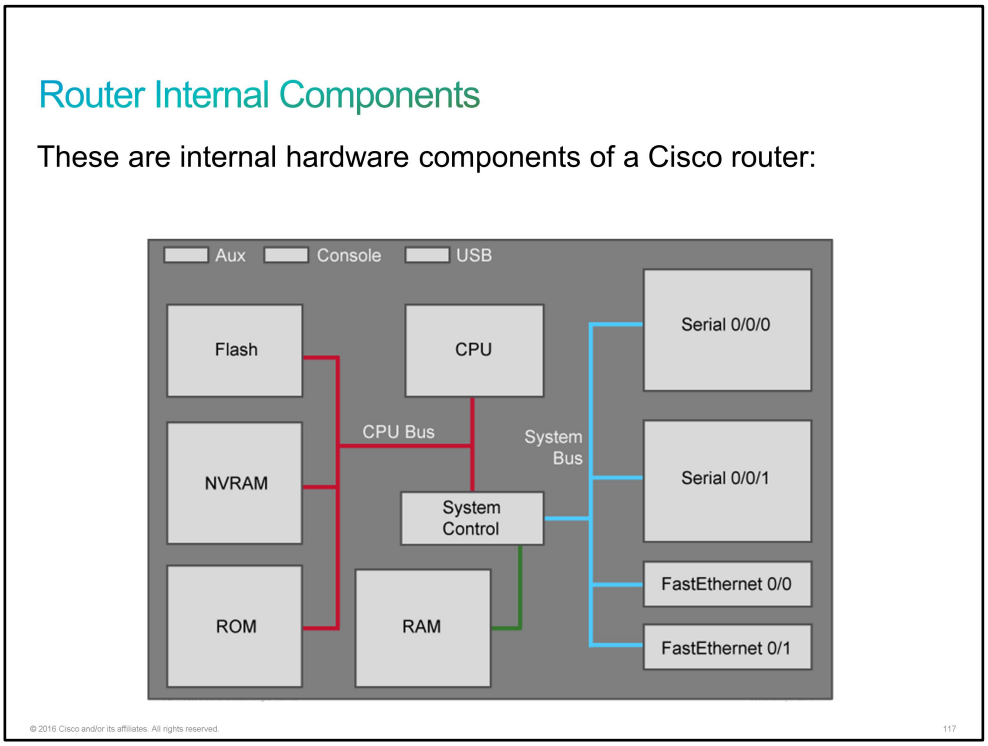
Introduction

When a Cisco router boots, it performs a series of steps in a particular order. At several points during the process, the router makes a decision about the next step to take. Knowledge of the boot sequence can be of great help when troubleshooting a Cisco router and also when adjusting its configuration. Carefully managing Cisco IOS images and configuration files reduces device downtime and maintains best practices. Cisco IOS image files contain the Cisco IOS Software that is required for a Cisco device to operate. The device configuration files contain a set of user-defined configuration commands that customize the functionality of a Cisco device.

Your boss sends you to your customer to explain how to manage Cisco devices.

Router Internal Components

The major internal components of a Cisco router include the CPU, interfaces, RAM, ROM, flash memory, and NVRAM.



A router is a computer, similar to a PC. Routers have several hardware and software components that you can find in other computers, including the following:

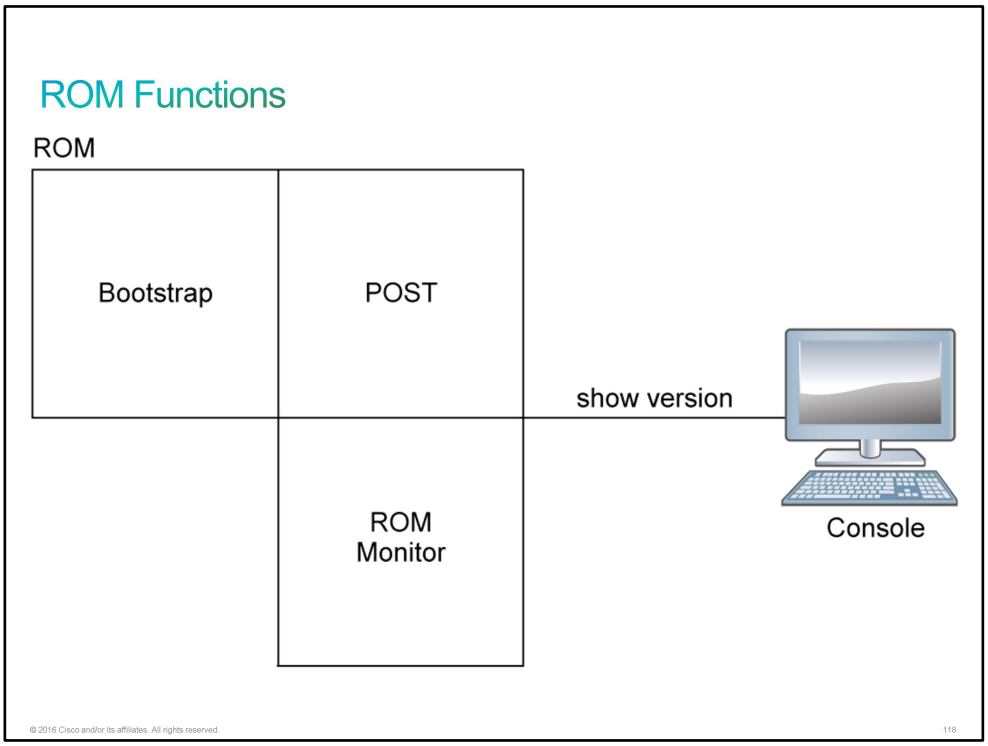
- **CPU:** The CPU executes operating system instructions such as system initialization, routing functions, and switching functions.
- **RAM:** RAM stores the instructions and data that the CPU needs to execute. This read/write memory contains the software and data structures that allow the router to function. RAM is volatile memory and loses its content when the router is powered down or restarted. However, the router also contains permanent storage areas such as ROM, Flash, and NVRAM. RAM is used to store the following components:
  - **Operating system:** Cisco IOS Software is copied into RAM during the boot process.
  - **Running configuration file:** This file stores the configuration commands that Cisco IOS Software is currently using on the router. With few exceptions, all commands that are configured on the router are stored in the running configuration file, which is also known as "running-config."
  - **IP routing table:** This file stores information about directly connected and remote networks. It is used to determine the best path to forward the packet.
  - **ARP cache:** ARP cache contains the IPv4 address to MAC address mappings, such as the ARP cache on a PC. The ARP cache is used on routers that have LAN interfaces such as Ethernet interfaces.
  - **Packet buffer:** Packets are temporarily stored in a buffer when they are received on an interface or before they exit an interface.

- **ROM:** ROM is a form of permanent storage. This type of memory contains microcode for basic functions to start and maintain the router. ROM contains the ROM monitor, which is used for router disaster recovery functions such as password recovery. ROM is nonvolatile, so it maintains the memory contents even when the power is turned off.
- **Flash memory:** Flash memory is nonvolatile computer memory that can be electrically stored and erased. Flash is used as permanent storage for the operating system. In most models of Cisco routers, Cisco IOS Software is permanently stored in the flash memory and copied into RAM during the boot process, where the CPU then executes it. Some older models of Cisco routers run Cisco IOS Software directly from flash. Flash consists of SIMMs or PCMCIA cards that can be upgraded to increase the amount of flash memory. The flash memory does not lose its contents when the router loses power or is restarted.
- **NVRAM:** NVRAM does not lose its information when the power is turned off. Cisco IOS Software uses NVRAM as permanent storage for the startup configuration file, which is also known as "startup-config." All configuration changes are stored in the running configuration file in RAM, and with few exceptions, Cisco IOS Software implements them immediately. To save these changes in case the router is restarted or loses power, the running configuration must be copied to NVRAM, where it is stored as the startup configuration file.
- **Configuration register:** The configuration register is used to control how the router boots. The configuration register value is stored in NVRAM.
- **Interfaces:** Interfaces are the physical connections to the external world for the router and include the following types, among others:
  - Ethernet, Fast Ethernet, and Gigabit Ethernet.
  - Asynchronous and synchronous serial.
  - USB interface, which can be used to add a USB flash drive to a router
  - Console and auxiliary ports. A console can have an RJ-45 or mini-USB connector.

Although there are several different types and models of routers, every router has the same general hardware components. Depending on the model, these components are located in different places inside the router.

ROM Functions

The ROM in a Cisco router contains microcode for basic router functions.



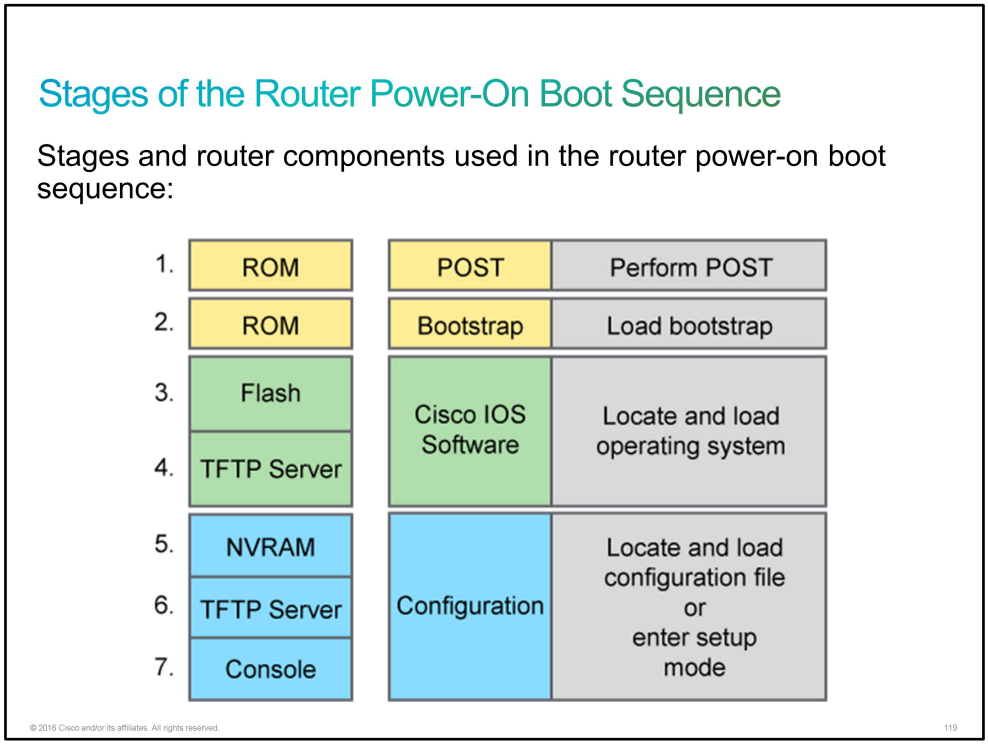
The figure shows three areas of the microcode that is generally contained in ROM:

- **Bootstrap code:** The bootstrap code is used to bring up the router during initialization. It reads the configuration register to determine how to boot and then, if instructed to do so, loads the Cisco IOS Software.
- **POST:** POST is the microcode that is used to test the basic functionality of the router hardware and determine which components are present.
- **ROM monitor:** The ROM monitor area includes a low-level operating system that is normally used for manufacturing, testing, troubleshooting, and password recovery. In ROM monitor mode, the router has no routing or IP capabilities.

Note	Depending on the specific Cisco router platform, the components that are listed may be stored in the flash memory or in the bootstrap memory to enable field upgrade to later versions.

Stages of the Router Power-On Boot Sequence

When a router boots, it performs a series of steps that include loading the Cisco IOS Software and the router configuration.



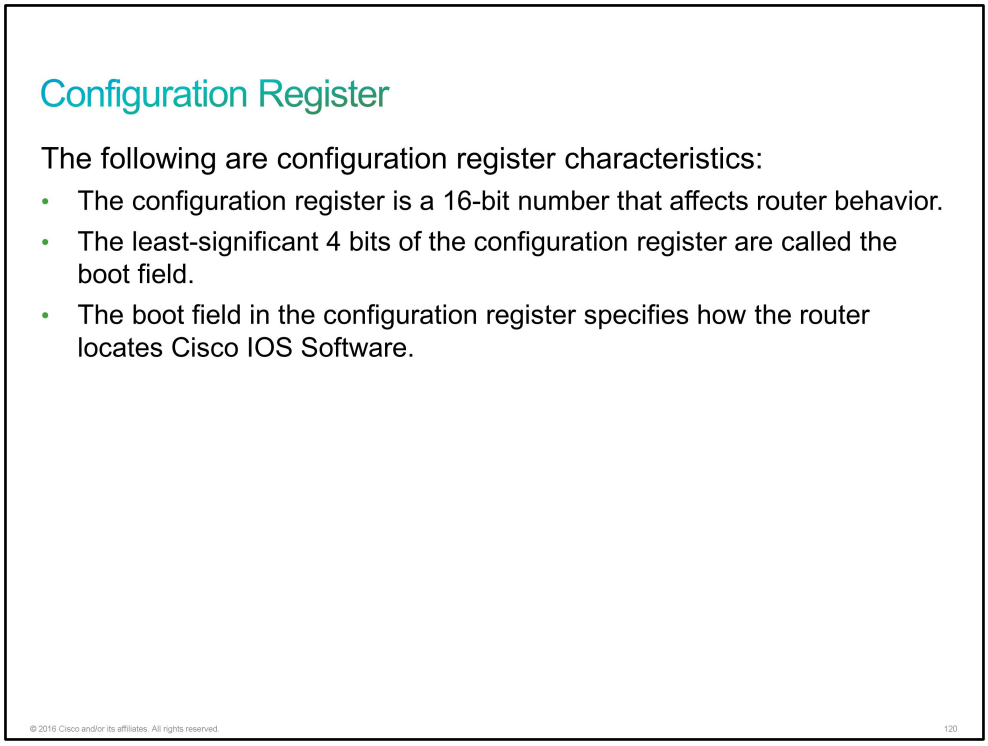
The sequence of events that occurs during the power-up (boot) of a router is explained in detail here. Understanding these events will help you accomplish operational tasks and troubleshoot router problems.

1. **Perform POST:** This event is a series of hardware tests that verifies if all components of a Cisco router are functional. During this test, the router also determines which hardware is present. POST executes from microcode that is resident in the system ROM.
2. **Load and run bootstrap code:** Bootstrap code is used to perform subsequent events such as locating the Cisco IOS Software, loading it into RAM, and running it. After the Cisco IOS Software is loaded and running, the bootstrap code is not used until the next time the router is reloaded or power-cycled.
3. **Find the Cisco IOS Software:** The bootstrap code determines the location of the Cisco IOS Software that will be run. Normally, the Cisco IOS Software image is located in the flash memory, but it can also be stored in other places such as a TFTP server. The configuration register and configuration file determine where the Cisco IOS Software images are located and which image file to use. If a complete Cisco IOS image cannot be located, a scaled-down version of the Cisco IOS Software is copied from ROM into RAM. This version of the Cisco IOS Software is used to help diagnose any problems and can be used to load a complete version of the Cisco IOS Software into RAM.
4. **Load the Cisco IOS Software:** After the bootstrap code has found the correct image, it loads this image into RAM and starts the Cisco IOS Software. Some older routers do not load the Cisco IOS Software image into RAM but execute it directly from flash memory instead.
5. **Find the configuration:** After the Cisco IOS Software is loaded, the bootstrap program searches for the startup configuration file in NVRAM.

6. **Load the configuration:** If a startup configuration file is found in NVRAM, the Cisco IOS Software loads it into RAM as the running configuration and executes the commands in the file, one line at a time. The running configuration file contains interface addresses, starts routing processes, configures router passwords, and defines other characteristics of the router. If no configuration exists, the router enters the setup utility or attempts an autoinstall to look for a configuration file from a TFTP server.
7. **Run the configured Cisco IOS Software:** When the prompt is displayed, the router is running the Cisco IOS Software with the current running configuration file. You can then begin using Cisco IOS commands on the router.

Configuration Register

The configuration register is a 16-bit number that resides in the NVRAM of a router.



Each bit has value 1 (on or set) or value 0 (off or clear), and each bit setting affects the router behavior upon the next reload power cycle.

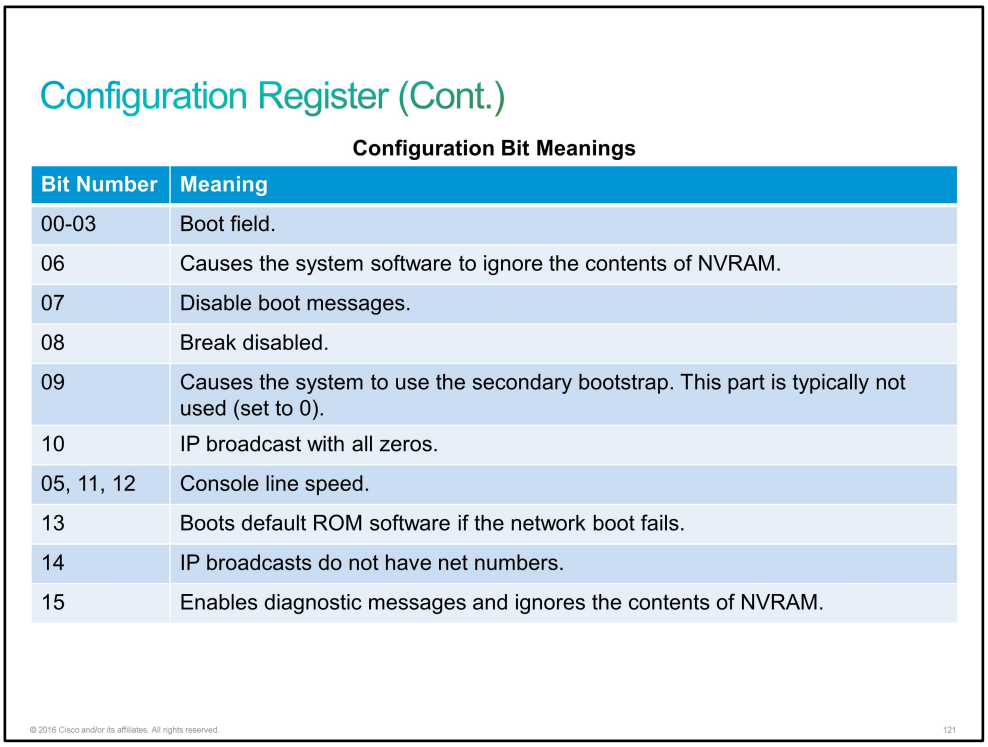
You can use the 16-bit configuration register to do the following:

- Force the router to boot into the ROM monitor.
- Select a boot source and default boot filename.
- Control broadcast addresses.
- Recover a lost password.
- Change the console line speed.

The lowest 4 bits (the rightmost number when displayed in hexadecimal) are called the boot field. The boot field specifies how the router locates Cisco IOS Software.

While the configuration register is a 16-bit number, it is displayed in hexadecimal. For example, when you issue the **show version** command, the value of the configuration register will appear something like 0x2102. The leading 0x simply indicates that the following value is displayed in hexadecimal format. As you know, each hexadecimal digit is 4 bits, which is why a 16-bit value can be displayed as a 4-digit hexadecimal number.

Here you can see the description of configuration register bit meanings, however they may differ between different platforms.



- Bit 8 controls the console Break key. Setting bit 8 (the factory default) causes the processor to ignore the console Break key. Clearing bit 8 causes the processor to interpret Break as a command to force the router into the bootstrap monitor, halting normal operation. Break can always be sent in the first 60 seconds while the router is rebooting, regardless of the configuration settings.
- Bit 9 controls the system boot. Clearing bit 9 (the factory default) causes the system to boot from flash memory. Clearing bit 9 causes the system to use the secondary bootstrap (netbooting), which is typically not used.
- Bit 10 controls the host portion of the IP broadcast address. Setting bit 10 causes the processor to use all zeros; clearing bit 10 (the factory default) causes the processor to use all ones. Bit 10 interacts with bit 14, which controls the network and subnet portions of the broadcast address. Table shows the combined effect of bit 10 and bit 14.

Configuration Register Settings for Broadcast Address Destination

Bit 10	Bit 14	Address (<net> <host>)
Off	Off	<ones> <ones>
On	Off	<zeros> <zeros>
On	On	<net> <zeros>
Off	On	<net> <ones>

- Bit 13 determines the router's response to a bootload failure. Setting bit 13 causes the router to load operating software from ROM after six unsuccessful attempts to load a boot file. Clearing bit 13 causes the router to continue indefinitely to attempt loading a boot file. By factory default, bit 13 is set to 0.

Configuration Register (Cont.)

Explanation of Boot Field Configuration Register Bits (00-03)

Boot Field	Meaning
0	Stays at the ROM monitor on a reload or power cycle
1	Boots the first image in flash memory as a system image
2-F	Enables default booting from flash memory Enables **boot system** commands that override default booting from flash memory

The boot field specifies a number in binary form. If you set the boot field value to 0, you must have console port access to boot the operating system manually. If you set the boot field to a value of 2 to F, and there is a valid **boot system** command that is stored in the configuration file, the router software processes each **boot** command in sequence until the process is successful or the end of the list is reached. If there are no **boot** commands in the configuration file, the router attempts to boot the first file in the flash memory.

Bit 5, bit 11, and bit 12 of the configuration register determine the baud rate of the console terminal. Table shows the bit settings for the eight available rates. The default baud rate is 9600 bps.

Configuration Register (Cont.)

Console Terminal Baud Rate Settings

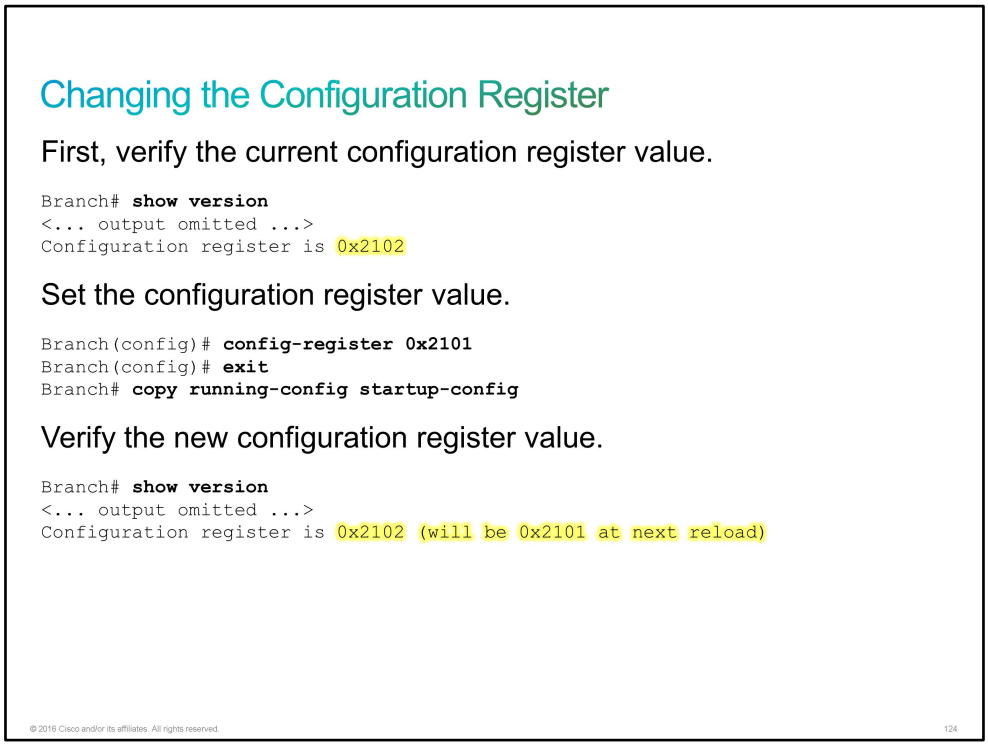
Baud	Bit 5	Bit 12	Bit 11
115200	1	1	1
57600	1	1	0
38400	1	0	1
19200	1	0	0
9600 (Default)	0	0	0
4800	0	0	1
2400	0	1	1
1200	0	1	0

Changing the Configuration Register

Before altering the configuration register, you should use the **show version** command to determine the current configuration register value. The last line of the **show version** command output shows the configuration register value.

Note	Record the configuration register setting, which is typically 0x2102, so you can change it back to the original setting if necessary.

You can use the **config-register** command in the global configuration mode to set the configuration register value. The syntax for this command is **config-register** *value*. The value argument is a hexadecimal number.



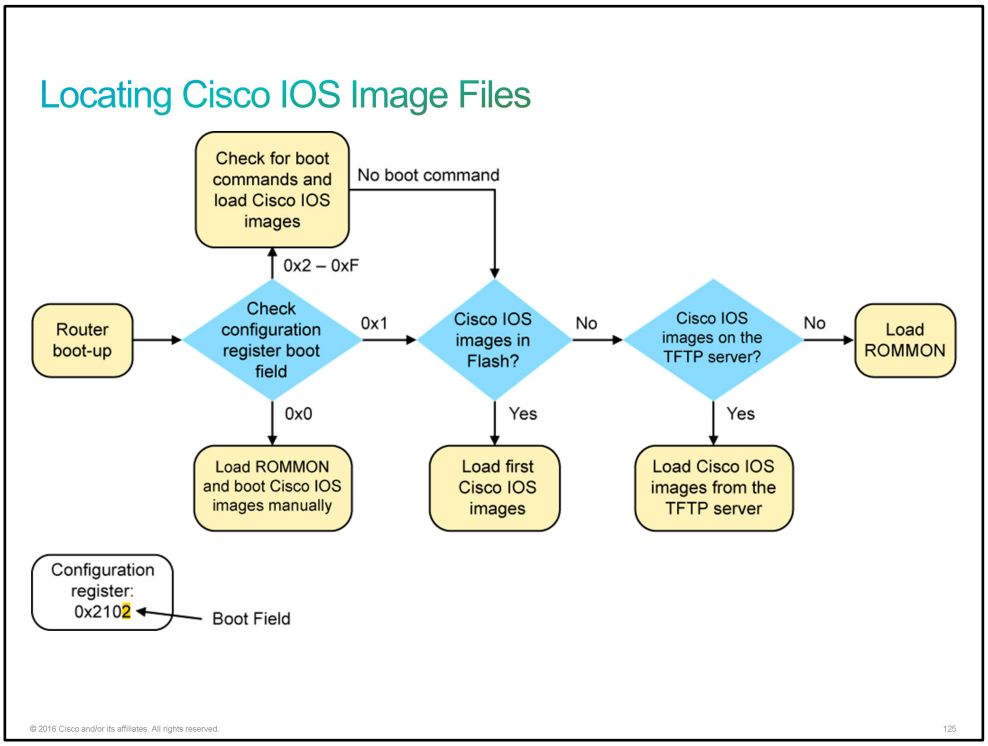
You should be careful when using the **config-register** command because the value argument sets all 16 bits of the configuration register. Only the lowest 4 bits of the configuration register (bits 3, 2, 1, and 0) form the boot field. For example, the default value of 0x2102 not only instructs the router to boot the system image from flash memory but also instructs the router to load the startup configuration with a console speed of 9600 baud (for most platforms), ignore the console Break key, and boot into ROM if the initial boot fails. If you modify the configuration register value, the change takes effect when the router reloads.

In the example, the configuration register value is changed from the default setting to 0x2101, and the configuration is saved to NVRAM. The new configuration register value will cause the router to load the bootstrap code.

If you issue the **show version** command again after changing the configuration register value, the command output shows both the currently configured value of the configuration register and the value that will be used at the next reload.

Locating Cisco IOS Image Files

When a Cisco router boots, it searches for the Cisco IOS image in a specific sequence. It searches for the location that is specified in the configuration register, flash memory, a TFTP server, and ROM.



The bootstrap code is responsible for locating the Cisco IOS Software. It searches for the Cisco IOS image in the following sequence:

1. The bootstrap code checks the boot field of the configuration register. The boot field tells the router how to boot up. The boot field can point to flash memory for the Cisco IOS image, the startup configuration file (if one exists) for commands that tell the router how to boot, or a remote TFTP server. Alternatively, the boot field can specify that no Cisco IOS image will be loaded, and the router should start a Cisco ROM monitor.
2. The bootstrap code executes the specifications of the configuration register boot field value as described in the following bullets. In a configuration register value, the "0x" indicates that the digits that follow are in hexadecimal notation. A configuration register value of 0x2102 has a boot field value of 0x2. The right-most digit in the register value is 2 and represents the lowest 4 bits of the register.
   - If the boot field value is 0x0, the router boots to the ROM monitor at the next power cycle or reload.
   - If the boot field value is 0x1, the router searches flash memory for Cisco IOS images.
   - If the boot field value is 0x2 to 0xF, at the next power cycle or reload, the bootstrap code parses the startup configuration file in NVRAM for **boot system** commands that specify the name and location of the Cisco IOS Software image to load. (Examples of **boot system** commands will follow.) If **boot system** commands are found, the router sequentially processes each **boot system** command in the configuration. If there are no **boot system** commands in the configuration, the router searches the flash memory for a Cisco IOS image.
3. If the router searches for and finds valid Cisco IOS images in flash memory, it loads the first valid image and runs it.

4. If it does not find a valid Cisco IOS image in flash memory, the router attempts to boot from a network TFTP server using the boot field value as part of the Cisco IOS image filename.
5. After six unsuccessful attempts at locating a TFTP server, the router loads the ROM monitor.

Note	The procedure for locating the Cisco IOS image depends on the Cisco router platform and default configuration register values. The procedure that is described here applies to the Cisco Integrated Services Routers.

Entering **boot system** commands in sequence in a router configuration can create a fault-tolerant boot plan. The **boot system** command is a global configuration command that allows you to specify the source for the Cisco IOS Software image to load. For example, the following command boots the system boot image file that is named c2900-universalk9-mz.SPA.152-4.M1.bin from the flash memory device:

```
Branch(config)#  boot system flash:c2900-universalk9-mz.SPA.152-4.M1.bin
```

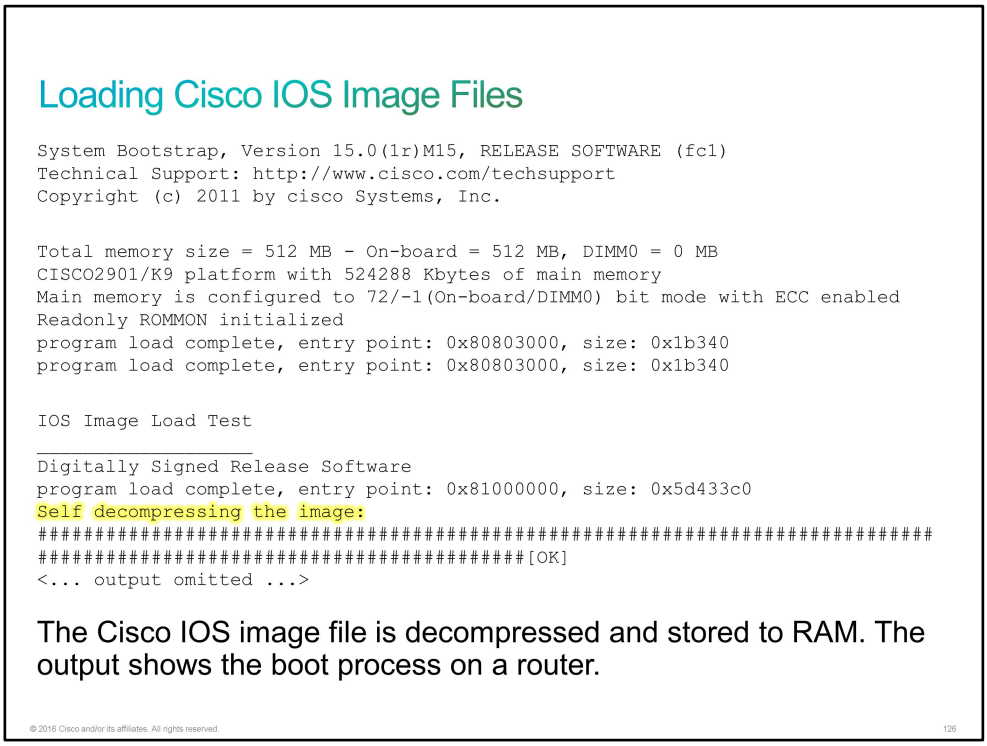
This next example specifies a TFTP server as a source of a Cisco IOS image, with a ROM monitor as the backup:

```
Branch(config)# boot system tftp://c2900-universalk9-mz.SPA.152-4.M1.bin
Branch(config)# boot system rom
```

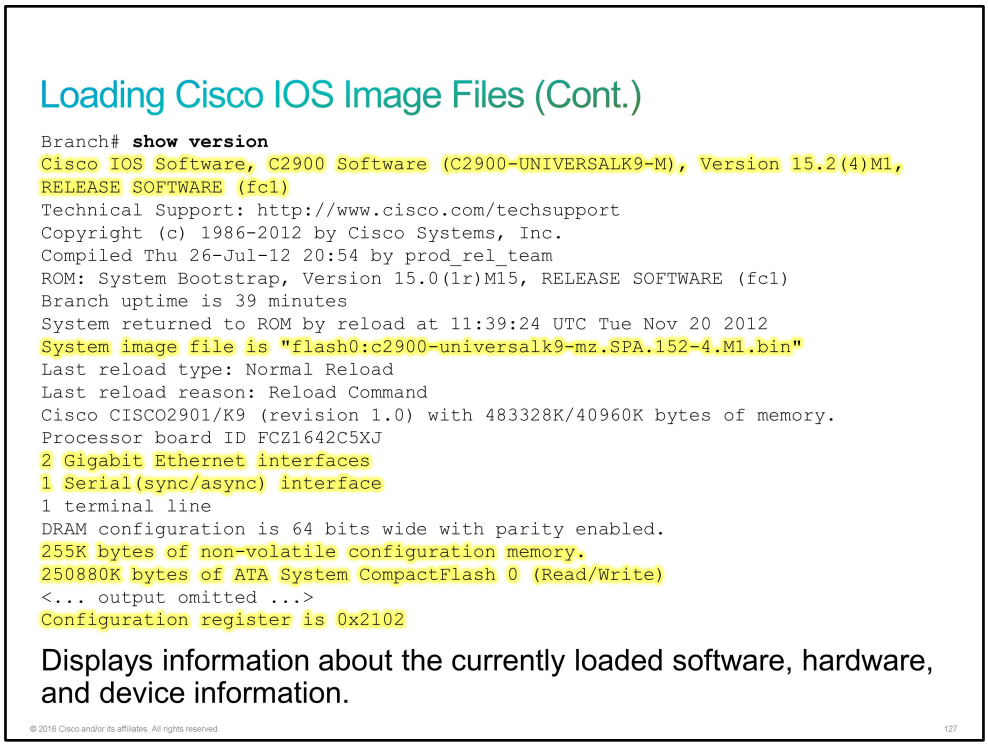
Loading Cisco IOS Image Files

When the router locates a valid Cisco IOS image file in the flash memory, the Cisco IOS image is normally loaded into RAM to run.

When the router locates a valid Cisco IOS image file in the flash memory, the Cisco IOS image is normally loaded into RAM to run. If the image needs to be loaded from the flash memory into RAM, it must first be decompressed. After the file is decompressed into RAM, it is started. When the Cisco IOS Software begins to load, you may see a string of pound signs (#), as shown in the figure, while the image decompresses.



The **show version** command can be used to help verify and troubleshoot some of the basic hardware and software components of the router. The **show version** command displays information about the version of the Cisco IOS Software that is currently running on the router, the version of the bootstrap program, and information about the hardware configuration, including the amount of system memory.



The output from the **show version** command includes the following:

- **Cisco IOS version**

```
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.2(4)M1, RELEASE
SOFTWARE (fc1)
```

This line from the example output shows the version of the Cisco IOS Software in RAM that the router is using.

- **ROM bootstrap program**

```
ROM: System Bootstrap, Version 15.0(1r)M15, RELEASE SOFTWARE (fc1)
```

This line from the example output shows the version of the system bootstrap software that is stored in ROM and was initially used to boot up the router.

- **Location of Cisco IOS image**

```
System image file is "flash0:c2900-universalk9-mz.SPA.152-4.M1.bin"
```

This line from the example output shows where the Cisco IOS image is located and loaded as well as its complete filename.

- **Interfaces**

```
2 Gigabit Ethernet interfaces

1 Serial (sync/async) interface
```

This section of the output displays the physical interfaces on the router. In this example, the Cisco 2901 router has two GigabitEthernet interfaces and one serial interface.

- **Amount of NVRAM**

```
255 KB of NVRAM
```

This line from the example output shows the amount of NVRAM on the router.

- **Amount of Flash**

```
250,880 KB of ATA System CompactFlash 0 (Read/Write)
```

This line from the example output shows the amount of flash memory on the router.

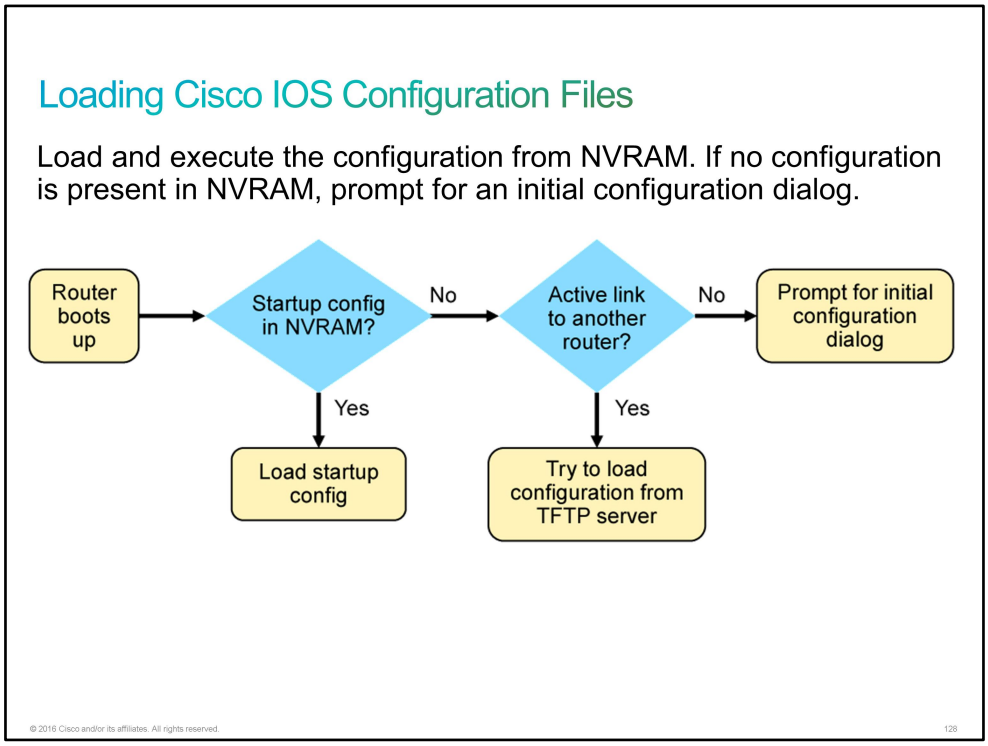
- **Configuration register**

```
Configuration register is 0x2102
```

The last line of the **show version** command displays the current configured value of the software configuration register in hexadecimal format. This value indicates that the router will attempt to load a Cisco IOS Software image from flash memory and load the startup configuration file from NVRAM.

Loading Cisco IOS Configuration Files

After the Cisco IOS Software image is loaded and started, the router must be configured to be useful. If there is an existing saved configuration file (startup-config) in NVRAM, it is executed. If there is no saved configuration file in NVRAM, the router either begins autoinstall or enters the setup utility.

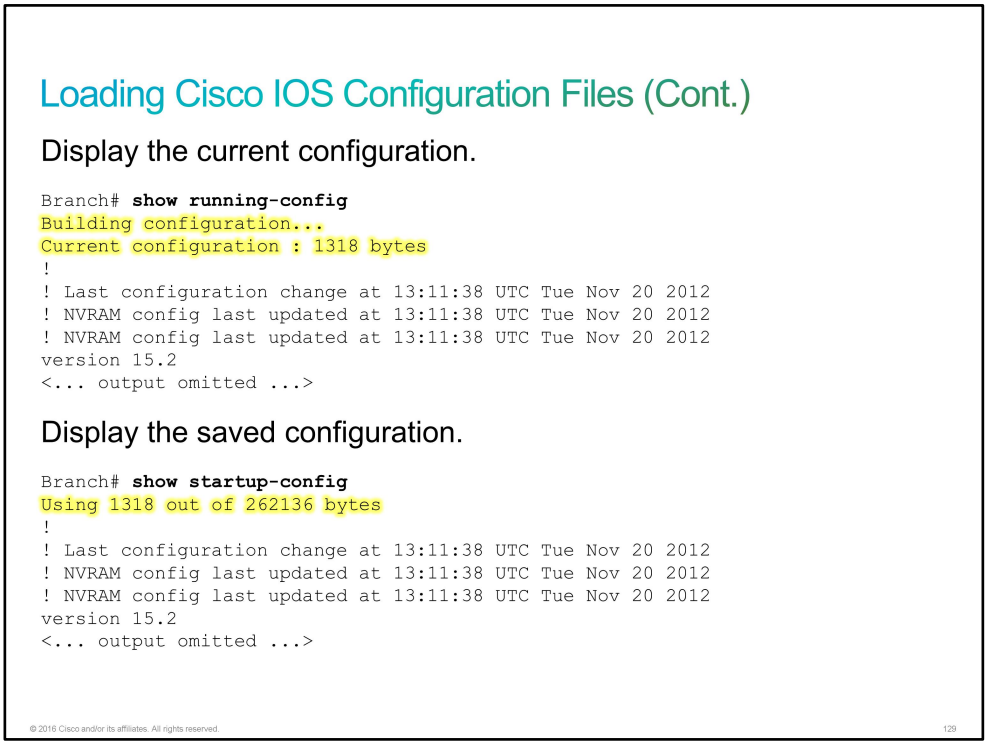


If the startup configuration file does not exist in NVRAM, the router may search for a TFTP server. If the router detects that it has an active link to another configured router, it sends a broadcast searching for a configuration file across the active link. This condition will cause the router to pause, but you will eventually see a console message such as the following:

```
%Error opening tftp://255.255.255.255/network-confg(Timed out)
%Error opening tftp://255.255.255.255/cisconet.cfg (Timed out)
```

The setup utility prompts the user at the console for specific configuration information to create a basic initial configuration on the router, as shown in this example:

```
<... output omitted ...>
If you require further assistance please contact us by sending email to
export@cisco.com.

Cisco CISCO2901/K9 (revision 1.0) with 483328K/40960K bytes of memory.
Processor board ID FCZ1642C5XJ
2 Gigabit Ethernet interfaces
1 Serial(sync/async) interface
1 terminal line
DRAM configuration is 64 bits wide with parity enabled.
255K bytes of non-volatile configuration memory.
250880K bytes of ATA System CompactFlash 0 (Read/Write)

         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:
```



The **show running-config** and **show startup-config** commands are among the most common Cisco IOS Software EXEC commands because they allow you to see the current running configuration in RAM on the router or the startup configuration commands in the startup configuration file in NVRAM that the router will use at the next restart.

If the words "**Current configuration**" are displayed, the active running configuration from RAM is being displayed.

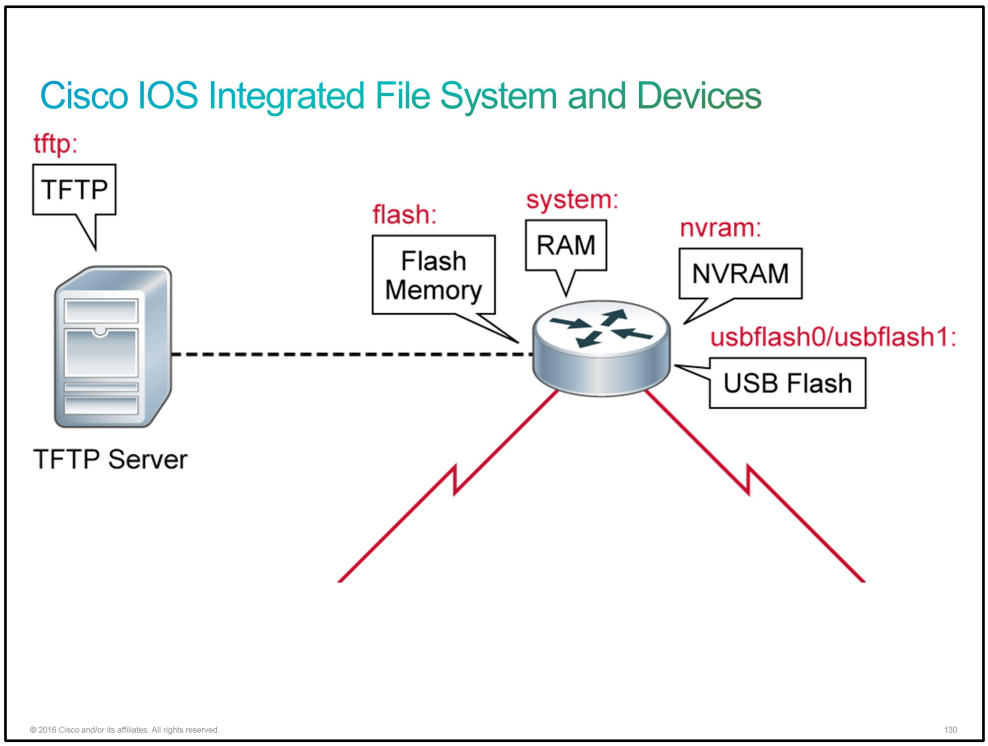
If there is a message at the top indicating how much nonvolatile memory is being used ("Using 1318 out of 262,136 B" in this example), the startup configuration file from NVRAM is being displayed.

Cisco IOS Integrated File System and Devices

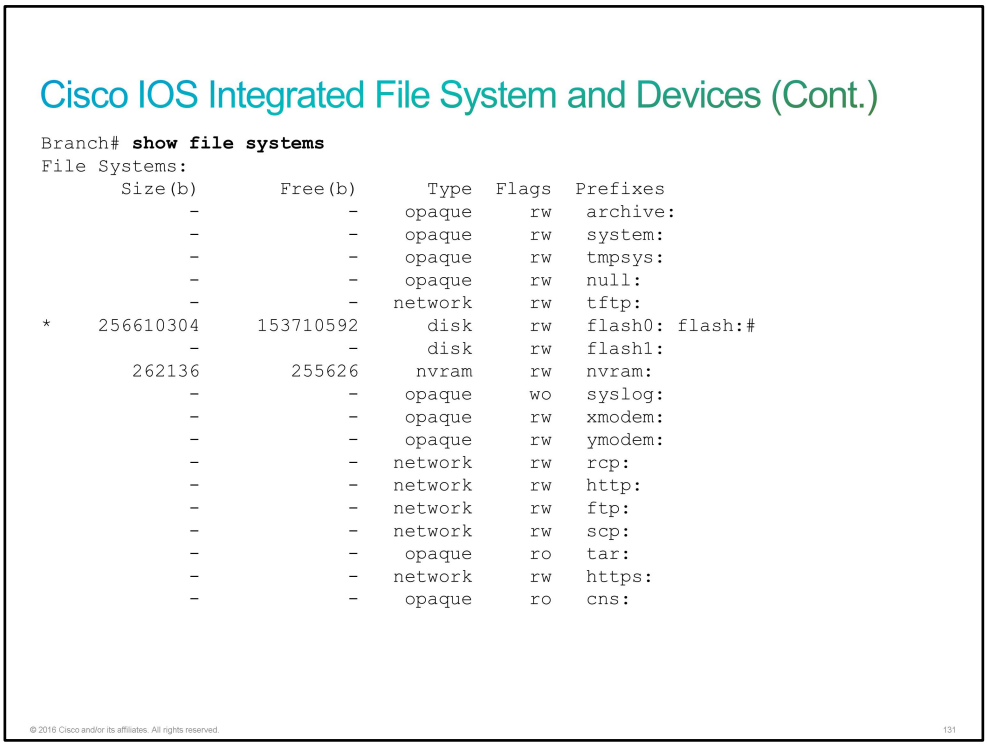
The availability of the network can be at risk if the configuration of a router or the operating system is compromised. Attackers who gain access to infrastructure devices can alter or delete configuration files. They can also upload incompatible Cisco IOS images or delete the Cisco IOS image. The changes are invoked automatically or invoked when the device is rebooted. To protect your network from these attacks, you have to be able to save, back up, and restore configurations and Cisco IOS images.

Cisco IOS devices provide a feature that is called the Cisco IOS Integrated File System (Cisco IFS). This system allows you to create, navigate, and manipulate directories on a Cisco device. The directories that are available depend on the platform. The Cisco IFS feature provides a single interface to all the file systems that a Cisco router uses, including the following:

- Flash memory file systems
- Network file systems such as TFTP, rcp, and FTP
- Any other endpoint for reading or writing data (such as NVRAM, the running configuration in RAM, and so on)



The next figure shows the output of the **show file systems** command, which lists all the available file systems on a Cisco 2901 Integrated Services Router. This command provides insightful information such as the amount of available and free memory and the type of file system and its permissions. Permissions include read only (as indicated by the "ro" flag), write only (as indicated by the "wo" flag), and read and write (as indicated by the "rw" flag). The FFS has an asterisk preceding it, which indicates the current default file system. The bootable Cisco IOS Software is located in the flash memory, so the pound symbol (#) that is appended to the Flash listing indicates a bootable disk.



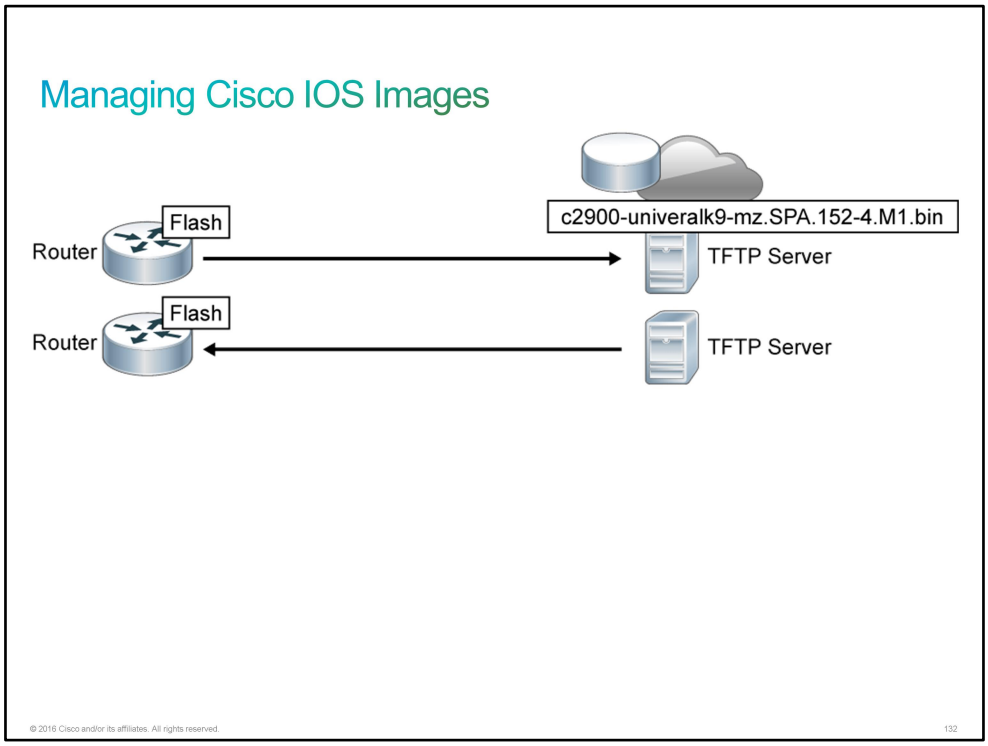
An important feature of the Cisco IFS is the use of the URL convention to specify files on network devices and the network. The URL prefix specifies the file system. The table contains some commonly used URL prefixes for Cisco network devices.

Prefix	Description
flash:	Flash memory. This prefix is available on all platforms. For platforms that do not have a device that is named Flash, the flash: prefix is aliased to slot0. Therefore, the flash: prefix can be used to refer to the main flash memory storage area on all platforms.
ftp:	FTP network server.
http:	HTTP network server.
nvram:	NVRAM.
rcp:	RCP network server.
system:	Contains the system memory, including the current running configuration.
tftp:	TFTP network server.
usbflash0, usbflash1	USB flash.

Managing Cisco IOS Images

As a network grows, storage of Cisco IOS Software images and configuration files on a central TFTP server enables control of the number and revision level of Cisco IOS images and configuration files that must be maintained.

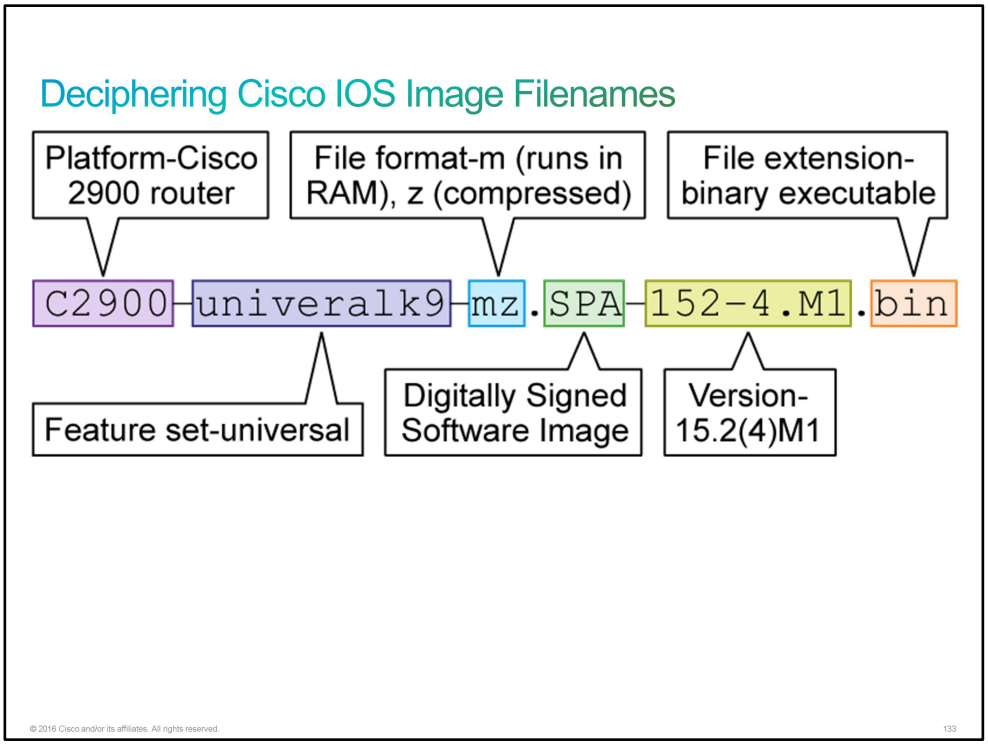
Production internetworks usually span wide areas and contain multiple routers. For any network, it is prudent to retain a backup copy of the Cisco IOS Software image in case the system image in the router becomes corrupted or accidentally erased.



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. Storage of Cisco IOS Software images and configuration files on a central TFTP server enables you to control the number and revision level of Cisco IOS images and configuration files that must be maintained. The network TFTP server can be another router, a workstation, or a host system.

Deciphering Cisco IOS Image Filenames

Before upgrading a Cisco IOS router, you must select a Cisco 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 that you understand this naming convention when upgrading and selecting Cisco IOS Software.



This list describes the parts of the example filename that is shown in the figure:

- The first part (c2900) identifies the platform on which the image runs. In this example, the platform is a Cisco 2900 Series Integrated Services Router.
- The second part (universal) specifies the feature set. In this case, "universal" refers to the universal, single image set that includes the IP base, security, unified communications, and data feature sets. Each router is activated for an IP base feature set. However, for other feature sets, software activation is needed.
- The third part (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.
- The fourth part (SPA) indicates that it is the file extension that Cisco software build process creates. Digitally signed Cisco IOS software is identified by a three-character extension in the image name.

File extension character	Character meaning
S (first character)	Stands for digitally signed software.
P or S (second character)	P and S stand for a production and special (development) image, respectively. A production (P) image is Cisco software that is approved for general release; a special (S) image is development software that is provided under special conditions for limited use.

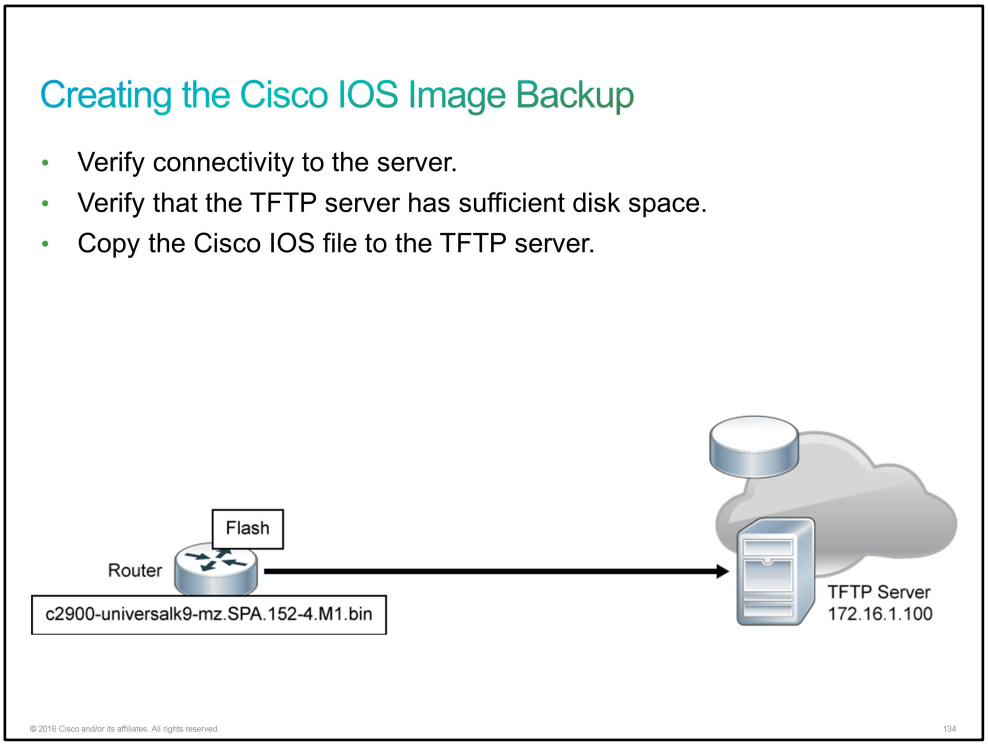
File extension character	Character meaning
A (third character)	Indicates the key version that is used to digitally sign the image. A key version is identified by an alphabetical character—for example, A, B, C, etc.

- The fifth part (15.2(4)M1) is the version number.
- The final part (bin) is the file extension. This extension indicates that this file is a binary executable file.

Note The Cisco IOS Software naming conventions, field meaning, image content, and other details are subject to change.

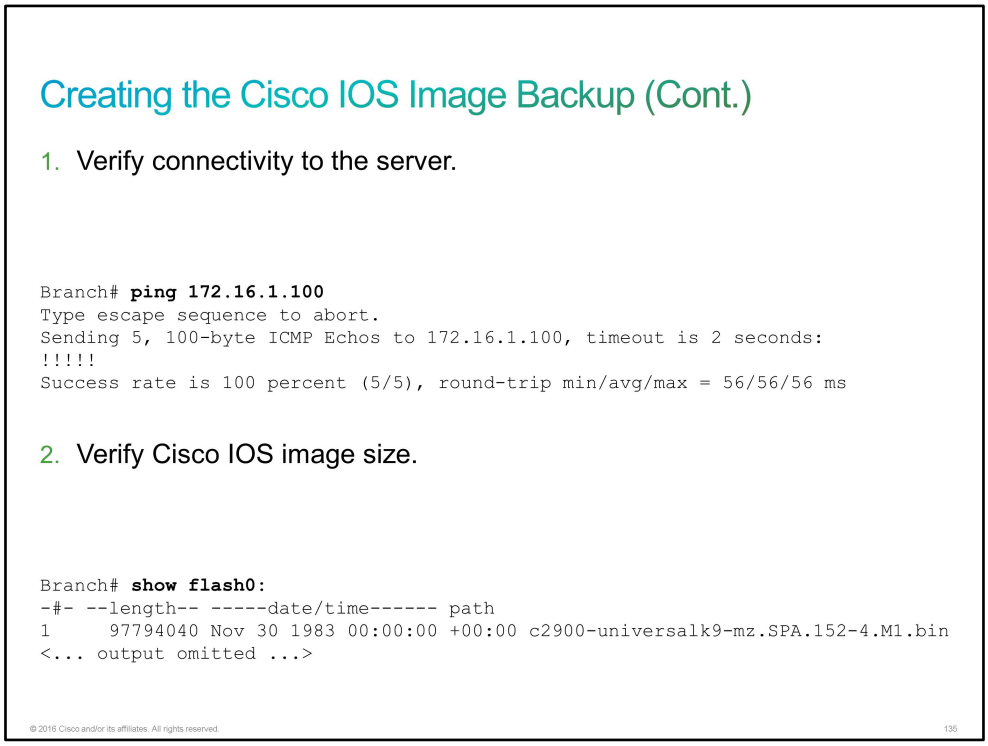
Creating the Cisco IOS Image Backup

To maintain network operations with minimum down time, it is necessary to have procedures in place for backing up Cisco IOS images. In this way, you can quickly copy an image back to a router in case of a corrupted or erased image on the router.



Follow these steps to create a backup of Cisco IOS images to the TFTP server:

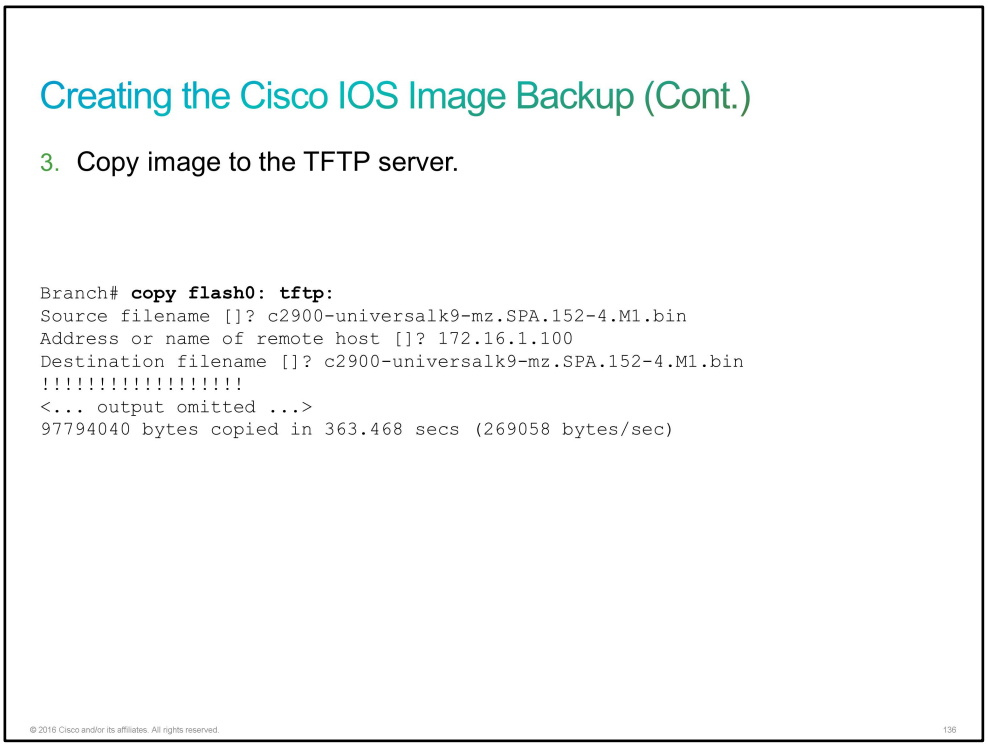
- Make sure that there is access to the network TFTP server. You can ping the TFTP server to test connectivity.
- Verify that the TFTP server has sufficient disk space to accommodate the Cisco IOS Software image. Use the **show flash0:** command on the router to determine the size of the Cisco IOS image file.
- Copy the image to the TFTP server using the **copy** command.



Before creating an image backup, verify connectivity to the TFTP server. You can verify connectivity by pinging the TFTP server from the router. In the example, the TFTP server is accessible from the router.

You should then make sure that you have sufficient disk space on the TFTP server to accommodate the Cisco IOS Software image. You can use the **show flash** command to verify the Cisco IOS Software image file size. The file in the example is 97,794,040 B (93 MB).

In this example, you will create a backup of the current image file on the router (c2900-universalk9-mz.SPA.152-4.M1.bin) to the TFTP server at 172.16.1.100.

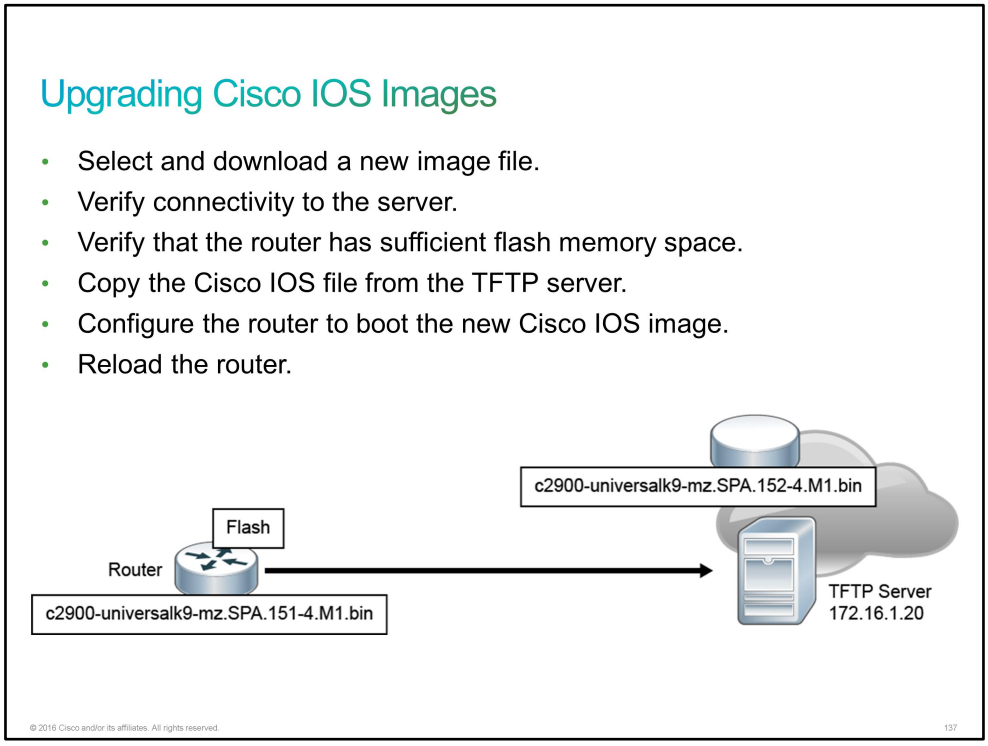


Finally, copy the Cisco IOS Software image file to the server, using the **copy** command. After you issue the command with specified source and destination URLs, you will be prompted for the source file name, IP address of the remote host, and destination filename. After you enter all this information, transfer of the file will occur. The table describes the command.

Command	Description
copy *source-url destination-url*	Copies any file from a source to a destination. The exact format of the source and destination URLs varies according to the file or directory location.

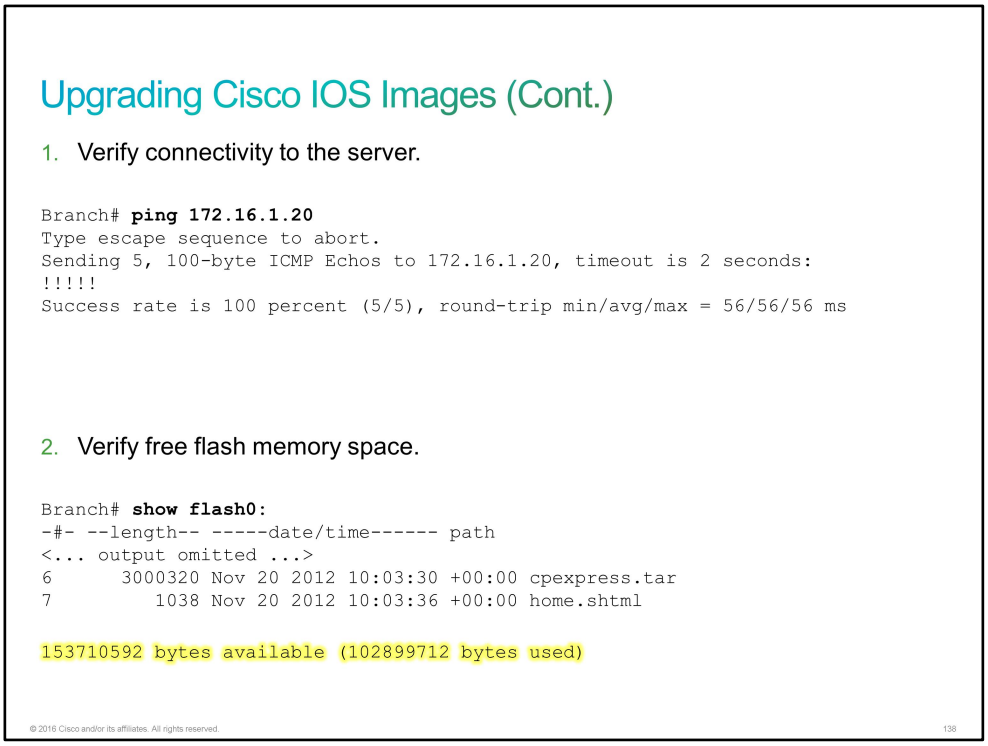
Upgrading Cisco IOS Images

Cisco constantly releases new Cisco IOS Software versions to resolve caveats (software defects) and provide new features.



If you decide to upgrade the software on the Cisco router, follow these steps:

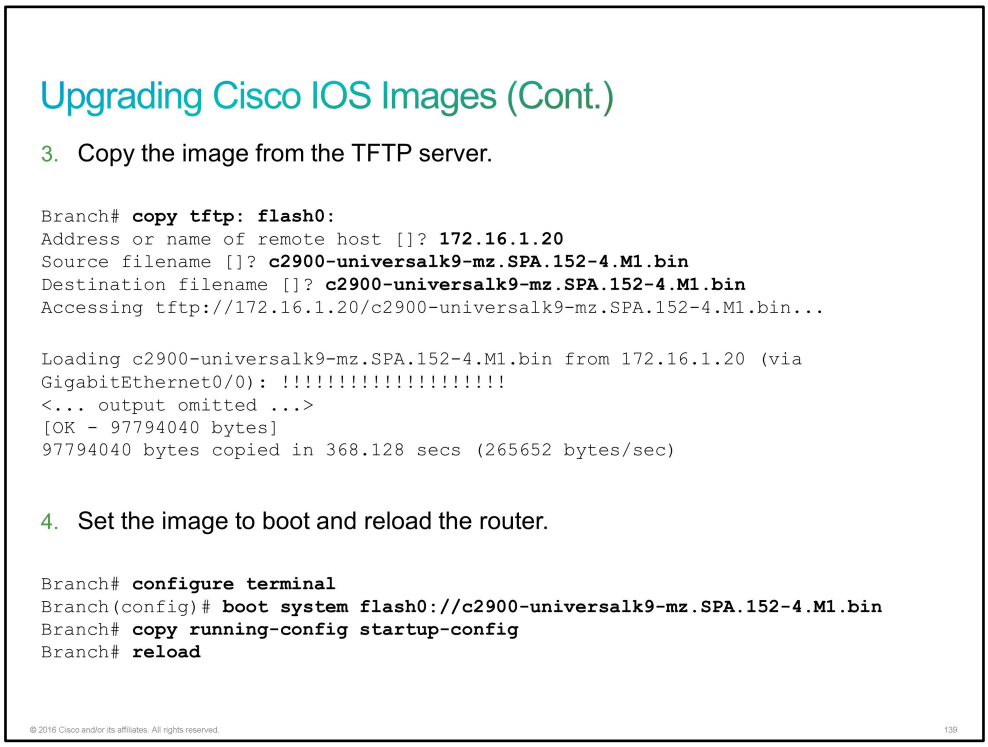
- Select a Cisco IOS image file that meets your requirements in terms of platform, features, and software defects. Download the file from http://www.cisco.com and transfer it to the TFTP server.
- Make sure that there is access to the network TFTP server. Ping the TFTP server to test connectivity.
- Make sure that there is sufficient flash memory space on the router that is being upgraded. You can verify the amount of free flash space by using the **show flash0:** command. Compare the free flash space with the new image file size.
- Copy the Cisco IOS image file from the TFTP server to the router using the **copy** command.
- When the image is saved on the router flash memory, you have to instruct the router to load the new image during the boot. Save the configuration.
- Finally, reload the router in order to boot the new image.



After you download the correct Cisco IOS image file and transfer it to the TFTP server, you should verify connectivity to the TFTP server by pinging the TFTP server from the router. In the example, the TFTP server is accessible from the router.

You should then make sure that you have sufficient disk space in the flash memory to accommodate the Cisco IOS image. You can use the **show flash** command to verify free the flash memory space. Free flash space in the example is 153,710,592 B.

In this example, you will load the new image file (c2900-universalk9-mz.SPA.152-4.M1.bin) from the TFTP server at 172.16.1.20 to the router.



Copy the Cisco IOS image file from the server to the router flash memory using the **copy** command. After you issue the command with specified source and destination URLs, you will be prompted for the IP address of the remote host, source file name, and destination file name. After you enter all the required information, the transfer of the file will begin.

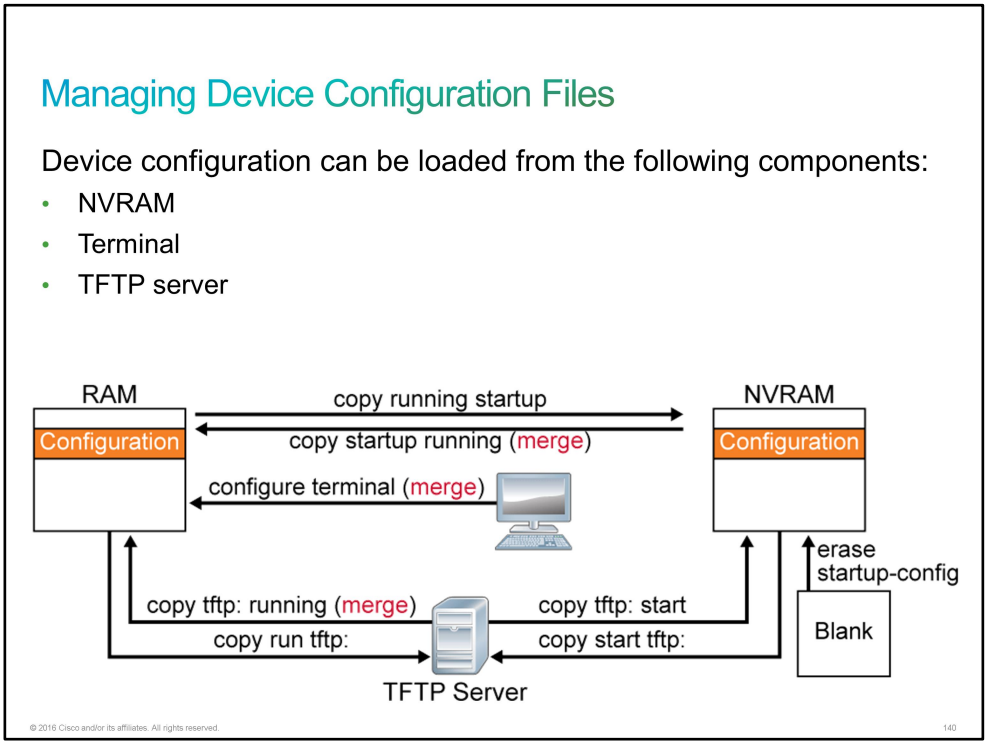
When the image file has been copied to the router, you have to instruct the router to boot the new image file. Use the **boot system** command to instruct the router to boot the specific file. Recall that the boot field in the configuration register has to be set to 0x2 to 0xF in order for the router to check the boot commands. Save the configuration.

Reload the router to boot the router with the new image. When the router has booted, you can verify if the new image has loaded using the **show version** command.

Command	Description
boot system *url*	Specify the system image that the router loads at startup.

Managing Device Configuration Files

Device configuration files contain a set of user-defined configuration commands that customize the functionality of a Cisco device.



Configuration files of a Cisco router are stored in the following locations:

- The running configuration is stored in RAM.
- The startup configuration is stored in NVRAM.

You can copy configuration files from the router to a file server using FTP or TFTP. For example, you can copy configuration files to back up a current configuration file to a server before changing its contents, therefore allowing the original configuration file to be restored from the server. The protocol that is used depends on which type of server is used.

You can copy configuration files from a server to the running configuration in RAM or to the startup configuration file in NVRAM of the router for one of the following reasons:

- To restore a backed-up configuration file.
- To use the configuration file for another router. For example, you may add another router to the network and want it to have a similar configuration as the original router. By copying the file to the network server and making the changes to reflect the configuration requirements of the new router, you can save time by not recreating the entire file.
- To load the same configuration commands onto all the routers in the network so that all the routers have similar configurations.
- To use the configuration file for another router. For example, you may add another router.

For example, in the **copy running-config tftp** command, the running configuration in RAM is copied to a TFTP server.

Use the **copy running-config startup-config** command after a configuration change is made in RAM and must be saved to the startup configuration file in NVRAM. Similarly, copy the startup configuration file in NVRAM back into RAM with the **copy startup running:** command (this is a merge, not a real copy). Notice that you can abbreviate the commands.

Similar commands exist for copying between a TFTP server and either NVRAM or RAM.

The following examples show common **copy** command usage. They list two methods to accomplish the same tasks. The first example is simple syntax, and the second example provides more explicit syntax.

- Copy the running configuration from RAM to the startup configuration in NVRAM, overwriting the existing file:

```
R2# copy running-config startup-config
R2# copy system:running-config nvram:startup-config
```

- Copy the running configuration from RAM to a remote location, overwriting the existing file:

```
R2# copy running-config tftp
R2# copy system:running-config tftp
```

- Copy a configuration from a remote source to the running configuration, merging the new content with the existing file:

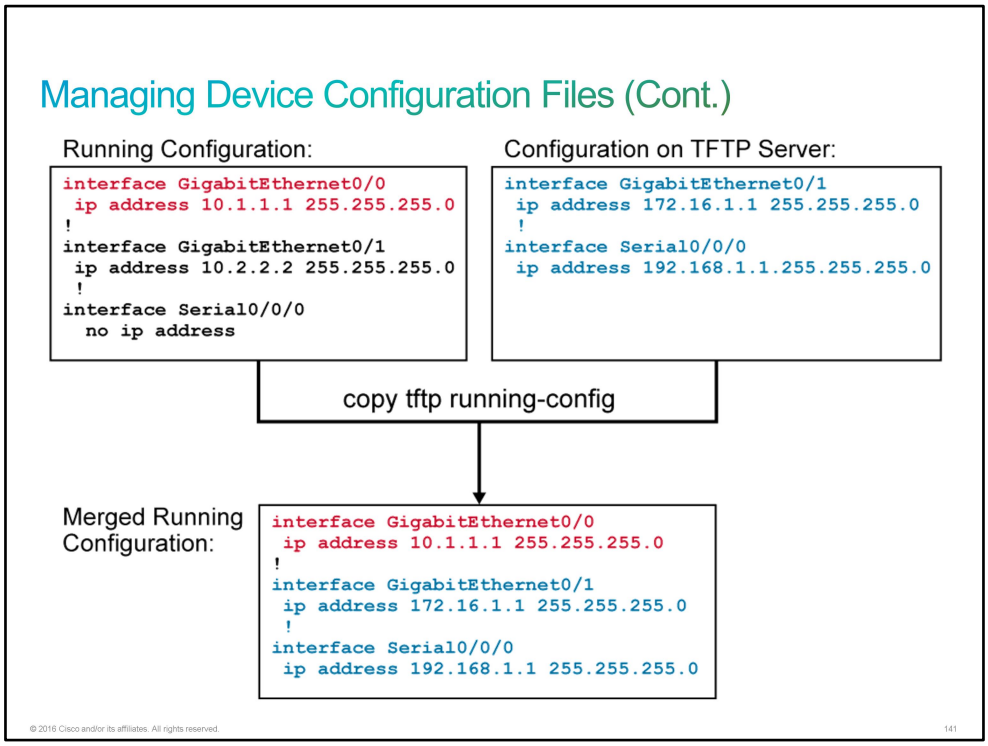
```
R2# copy tftp running-config
R2# copy tftp system:running-config
```

- Copy a configuration from a remote source to the startup configuration, overwriting the existing file:

```
R2# copy tftp startup-config
R2# copy tftp nvram:startup-config
```

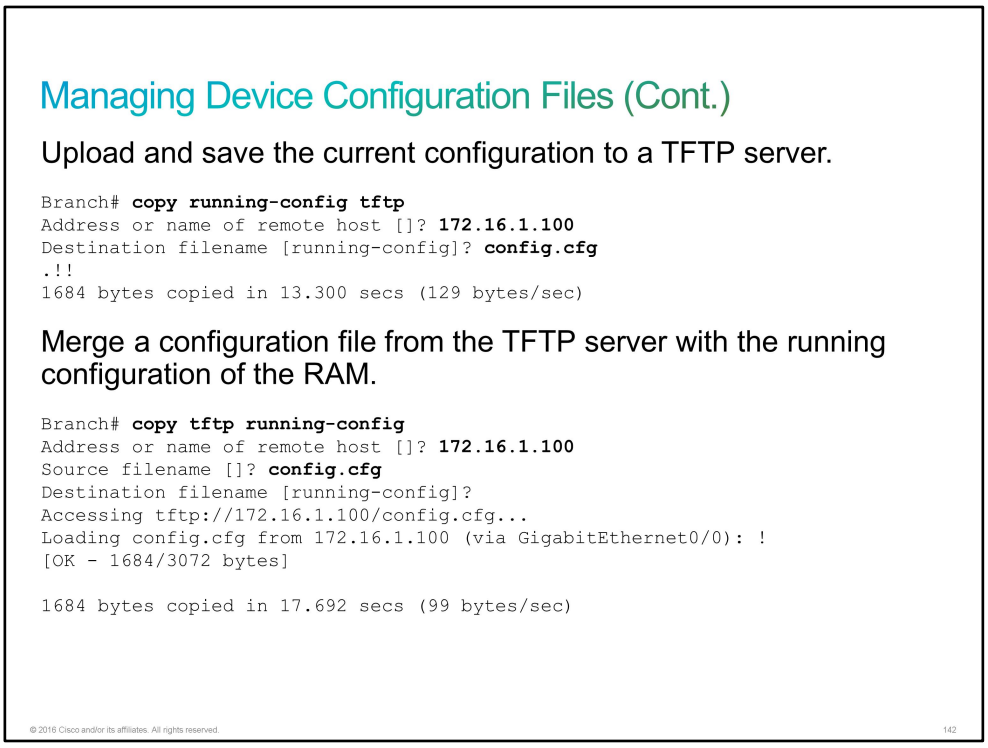
Use the **configure terminal** command to interactively create configurations in RAM from the console or remote terminal.

Use the **erase startup-config** command to delete the saved startup configuration file in NVRAM.



This figure shows an example of how to use the **copy tftp running-config** command to merge the running configuration in RAM with a saved configuration file on a TFTP server.

Note	When a configuration is copied into RAM from any source, the configuration merges with or overlays any existing configuration in RAM, rather than overwriting it. New configuration parameters are added, and changes to existing parameters overwrite the old parameters. Configuration commands that exist in RAM for which there are no corresponding commands in NVRAM remain unaffected. Copying the running configuration from RAM into the startup configuration file in NVRAM will overwrite the startup configuration file in NVRAM.



You can use the TFTP servers to store configurations in a central place, allowing centralized management and updating. Regardless of the size of the network, there should always be a copy of the current running configuration online as a backup.

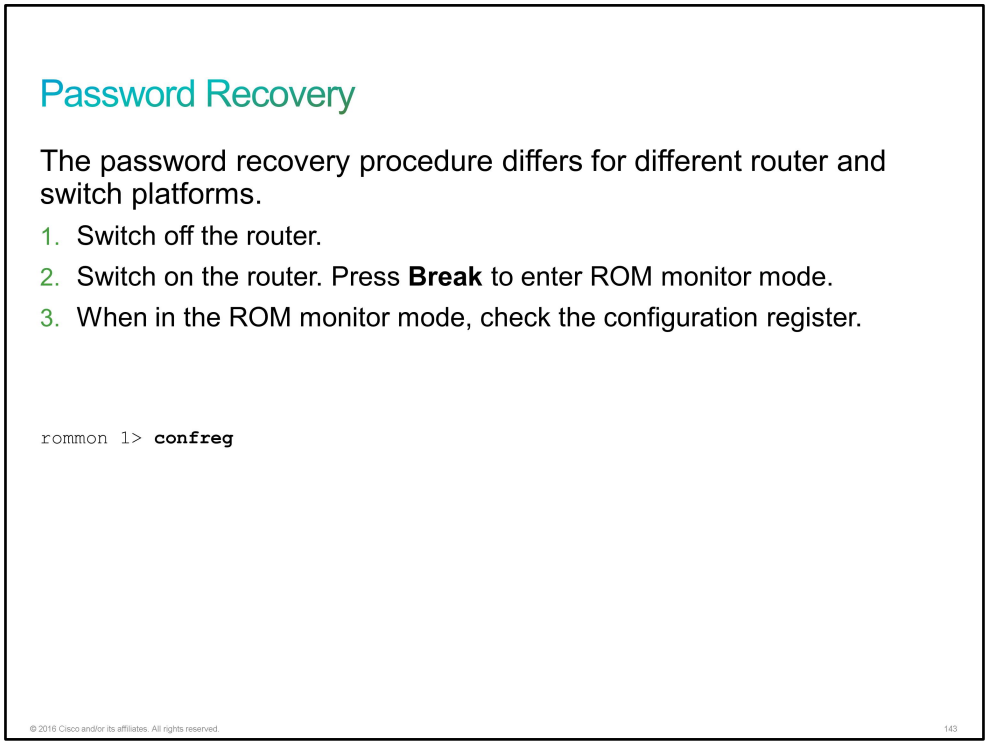
The **copy running-config tftp** command allows the current configuration to be uploaded and saved to a TFTP server. The IP address or name of the TFTP server and the destination filename must be supplied. A series of exclamation marks in the display shows the progress of the upload.

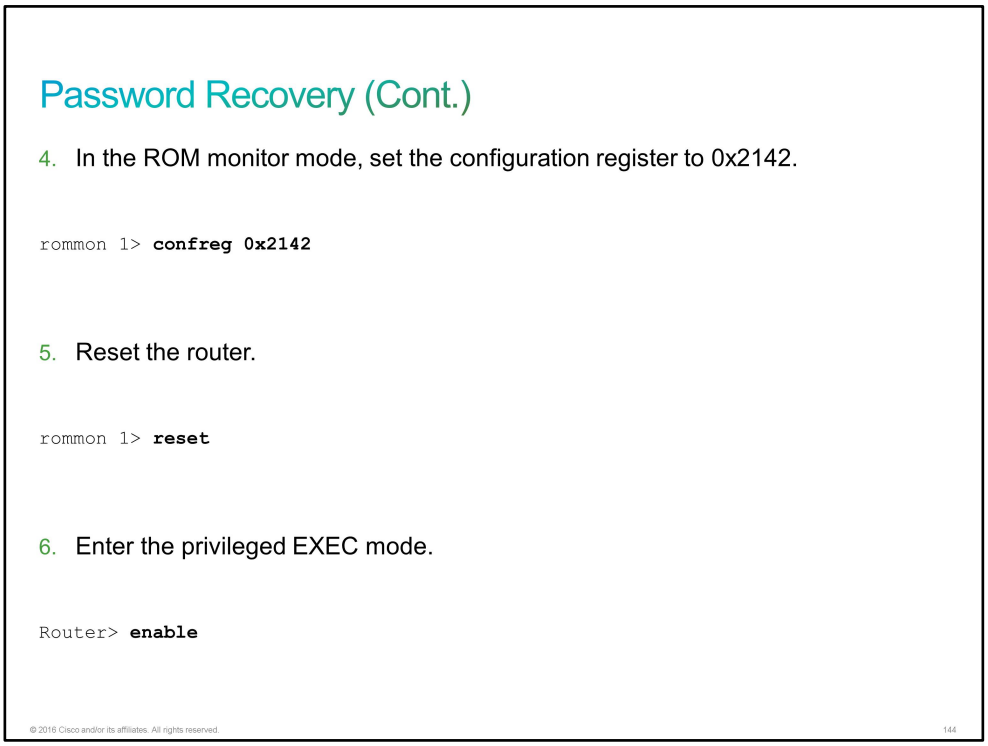
The **copy tftp running-config** command downloads a configuration file from the TFTP server to the running configuration of the RAM. Again, the address or name of the TFTP server and the source and destination filename must be supplied. In the example, IPv4 is used as a transport protocol. In this case, because you are copying the file to the running configuration, the destination filename should be running-config. This process is a merge process, not an overwrite process.

Password Recovery

If the password is mistyped or forgotten, access to the router privileged EXEC mode is not possible, and you must perform a password recovery procedure.

Different routers and switches may have different password recovery procedure. Refer to the http://www.cisco.com to find password recovery procedure for your router or switch. The following procedure describes password recovery for Cisco ISRs (Cisco Integrated Services Routers).

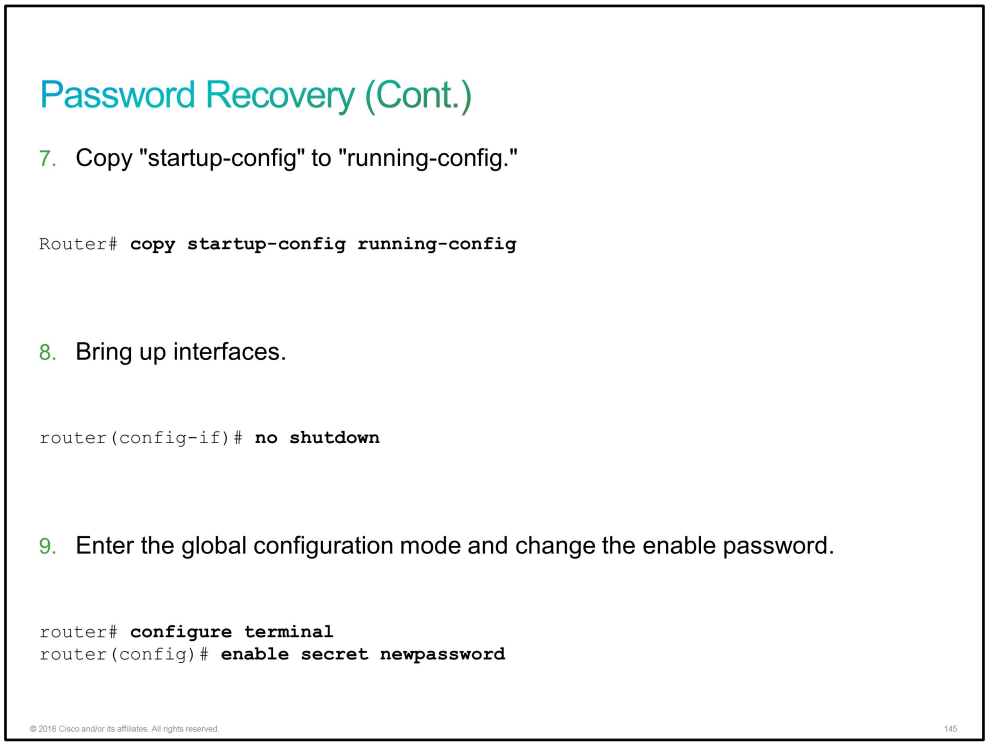




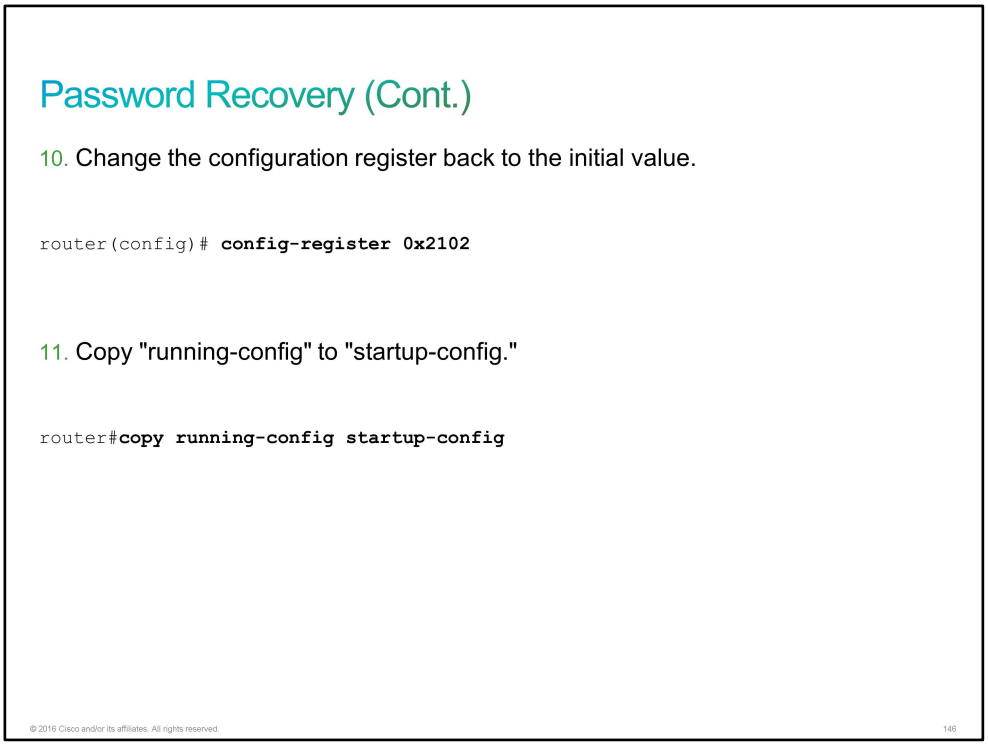
You can use the configuration register to perform the password recovery procedure. You have to set the configuration register value to a value that will instruct the router to ignore the startup configuration, which includes the forgotten enable password. Because a user cannot enter the privileged EXEC mode in order to change the configuration register, the register has to be changed in the ROM monitor. In order to enter the ROM monitor, reboot the router and press **Break** to interrupt the boot process to get into the ROM monitor.

Follow these steps to perform password recovery:

- Either switch off or shut down the router.
- Switch on the router. Press **Break** to interrupt the boot process to get into the ROM monitor.
- When the router is in the ROM monitor mode, check configuration register by using the **confreg** command.
- Set the configuration register to 0x2142. The hexadecimal number of 4 indicates that bit 6 is set and this will instruct the router to ignore the startup configuration at the next reload.
- Reset the router. The router reboots but ignores the saved configuration. Do not enter the interactive setup dialog.
- Enter the privileged EXEC mode. You should be able to do so because the saved configuration is ignored and the empty configuration without the enable password is loaded.



- Copy "startup-config" to "running-config" in order to load the saved configuration. After this step, all interfaces might be disabled. You have to enable the desired interfaces using the **no shutdown** command.
- Because the startup configuration merged into the running configuration, the interfaces will be shut down. Bring up the appropriate interfaces using the **no shutdown** command.
- Enter the global configuration mode and change the enable password to a new value. Do not forget or mistype the password this time.



- Change the configuration register back to the initial value. Change the value to the previously recorded value or to 0x2102. This action will instruct the router to not ignore the startup configuration at the next reload.
- Copy "running-config" to "startup-config" in order to save changes regarding the new enable password and the configuration register value.

Challenge

1. Which router component stores the IP routing table?

   A. NVRAM
   B. RAM
   C. ROM
   D. flash memory

2. Which microcode is used to test the basic functionality of router hardware and to determine which components are present?

   A. POST
   B. bootstrap
   C. ROM monitor
   D. ROM

3. From which two router components could the router obtain its operating system? (Choose two.)

   A. ROM
   B. flash
   C. TFTP server
   D. NVRAM
   E. console

4. From which three router components could the router obtain its configuration? (Choose three.)

   A. ROM
   B. flash
   C. TFTP server
   D. NVRAM
   E. console

5. Which Cisco IOS command can you use to examine the configuration register?

   A. **show running-config**
   B. **show startup-config**
   C. **show version**
   D. **show config-register**

6. Which Cisco IOS command do you use to verify the Cisco IOS image size?

   A. **show flash0:**
   B. **show running-config**
   C. **show startup-config**
   D. **show file systems**

7. When a configuration is copied into the startup configuration file in NVRAM from any source, the configuration merges with or overlays any existing configuration in NVRAM. True or false?

   A. true
   B. false

8. Put the commands in the order in which they should be used when upgrading a Cisco IOS image.

copy	1
show flash	2
ping	3
reload	4
boot system	5

Answer Key

Challenge

1. B
2. A
3. B, C
4. C, D, E
5. C
6. A
7. B
8.

ping	1
show flash	2
copy	3
boot system	4
reload	5

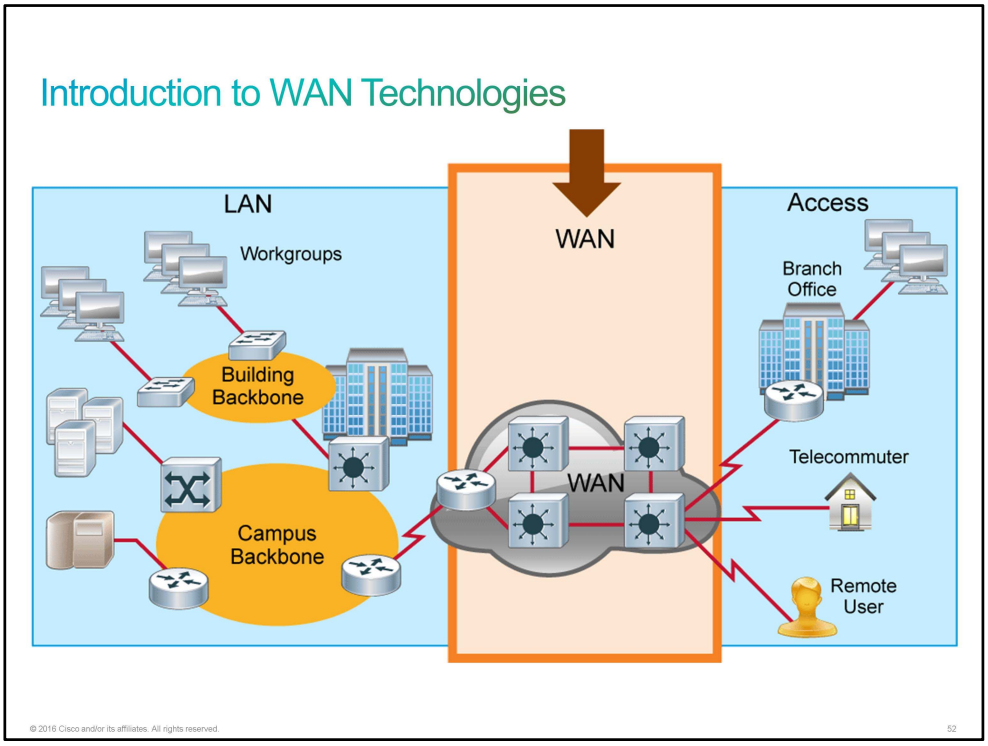
Lesson 1: Understanding WAN Technologies

Introduction

In order to continue to advance in your career, you have asked Bob if you can get more involved in WAN deployments. Although Bob is glad that you want to expand your skills and knowledge, he wants to assess your level of preparedness before taking you with him on WAN deployment jobs. To gauge your level of preparedness for WAN deployments, CCS provides a test. Bob tells you that the test will require you to demonstrate your knowledge of WAN devices, WAN cabling, WAN protocols, and WAN technologies.

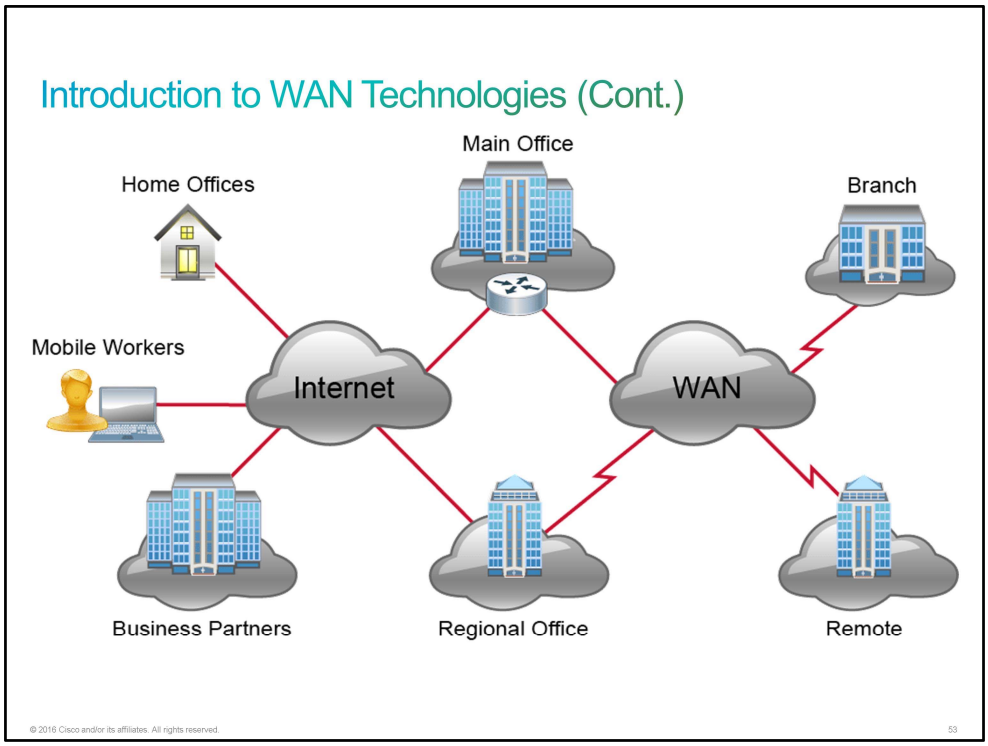
Introduction to WAN Technologies

A WAN is a data communications network that operates beyond the geographic scope of a LAN. WANs use facilities that a service provider or carrier, such as a telephone or cable company, provide. They connect the locations of an organization to each other, to locations of other organizations, to external services, and to remote users. WANs carry various traffic types such as voice, data, and video.



The following are three major characteristics of WANs:

- WANs generally connect devices that are separated by a broader geographic area than a LAN can serve.
- WANs use the services of carriers such as telcos, cable companies, satellite systems, and network providers.
- WANs use connections of various types to provide access to bandwidth over large geographic areas.



There are several reasons why WANs are necessary in a communications environment.

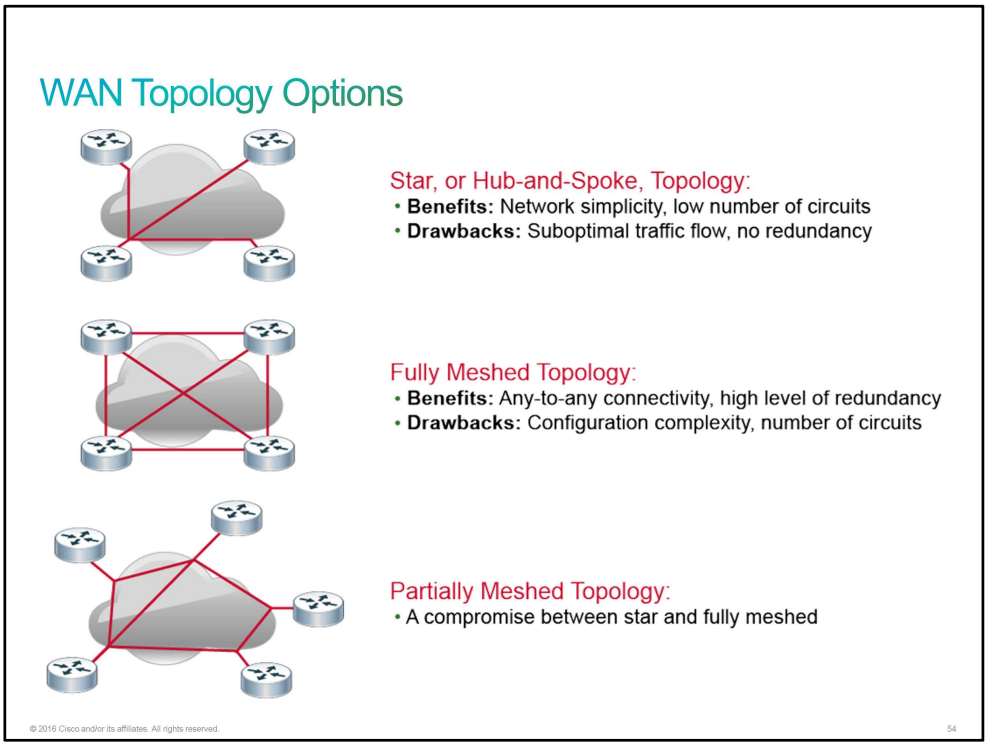
LAN technologies provide speed and cost efficiency for the transmission of data in organizations in relatively small geographic areas. You need WANs in a communications environment because some business needs require communication among remote sites for many reasons, including the following:

- People in the regional or branch offices of an organization need to be able to communicate and share data.
- Organizations often want to share information with other organizations across large distances.
- Employees who travel on company business frequently need to access information that resides on their corporate networks.

Because it is not feasible to connect computers across a country or around the world in the same way that computers are connected in a LAN environment with cables, different technologies have evolved to support this need. Increasingly, the Internet is being used as an inexpensive alternative to an enterprise WAN for some applications.

WAN Topology Options

A physical topology describes the physical arrangement of network devices that allow for data to move from a source to a destination network. There are three basic topologies for a WAN design.



Star or hub-and-spoke topology: This topology features a single hub (central router) that provides access from remote networks to a core router. All communication among the networks goes through the core router. The advantages of a star approach are simplified management and minimized tariff costs. However, the disadvantages are significant:

- The central router (hub) represents a single point of failure.
- The central router limits the overall performance for access to centralized resources. The central router is a single pipe that manages all traffic that is intended either for the centralized resources or for the other regional routers.

Fully meshed topology: In this topology, each routing node on the periphery of a given packet-switching network has a direct path to every other node on the cloud. The key rationale for creating a fully meshed environment is to provide a high level of redundancy. A fully meshed topology is not viable in large packet-switched networks. The following are the key issues of a fully meshed topology:

- Many virtual circuits are required (one for every connection between routers).
- Configuration is complex for routers without multicast support in nonbroadcast environments.

Partially meshed topology: This topology reduces the number of routers within a region that have direct connections to all other nodes in the region. All nodes are not connected to all other nodes. There are many forms of partially meshed topologies. In general, partially meshed approaches provide the best balance for regional topologies, which are based on the number of virtual circuits, redundancy, and performance.

Note	Large networks usually deploy a layered combination of these technologies—for example, a partial mesh in the network core, redundant hub-and-spoke for larger branches, and simple hub-and-spoke for noncritical remote locations.

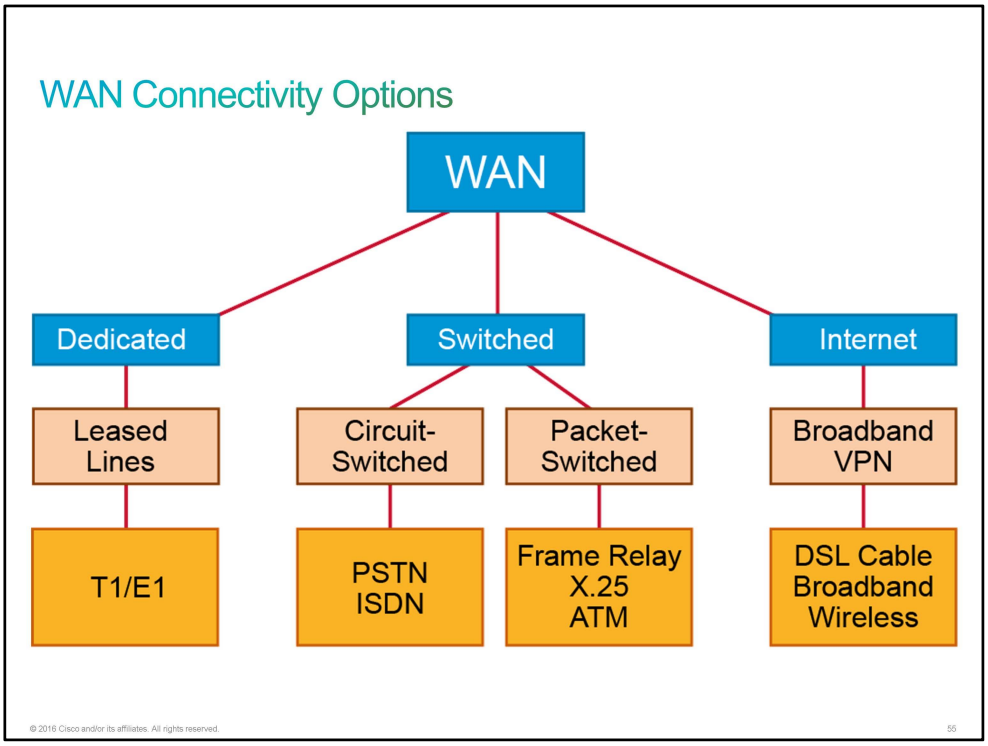
Network downtime can be very expensive in terms of decreased productivity and potential loss of revenue. To increase network availability, many organizations deploy a dual-carrier WAN design to increase redundancy and path diversity.

Single-carrier WANs are simpler and easier to support and manage. However, network outages can be catastrophic. You should perform an analysis of the downtime cost. You should make sure that there are adequate penalties in the contract with the service provider to cover the cost of downtime.

Dual-carrier WANs provide better path diversity with better fault isolation between providers. The cost of downtime to your organization usually exceeds the additional cost of the second provider and the complexity of managing redundancy.

WAN Connectivity Options

You have many options for implementing WAN solutions currently available. They differ in technology, speed, and cost. WAN connections can be either over a private infrastructure or over a public infrastructure such as the Internet.

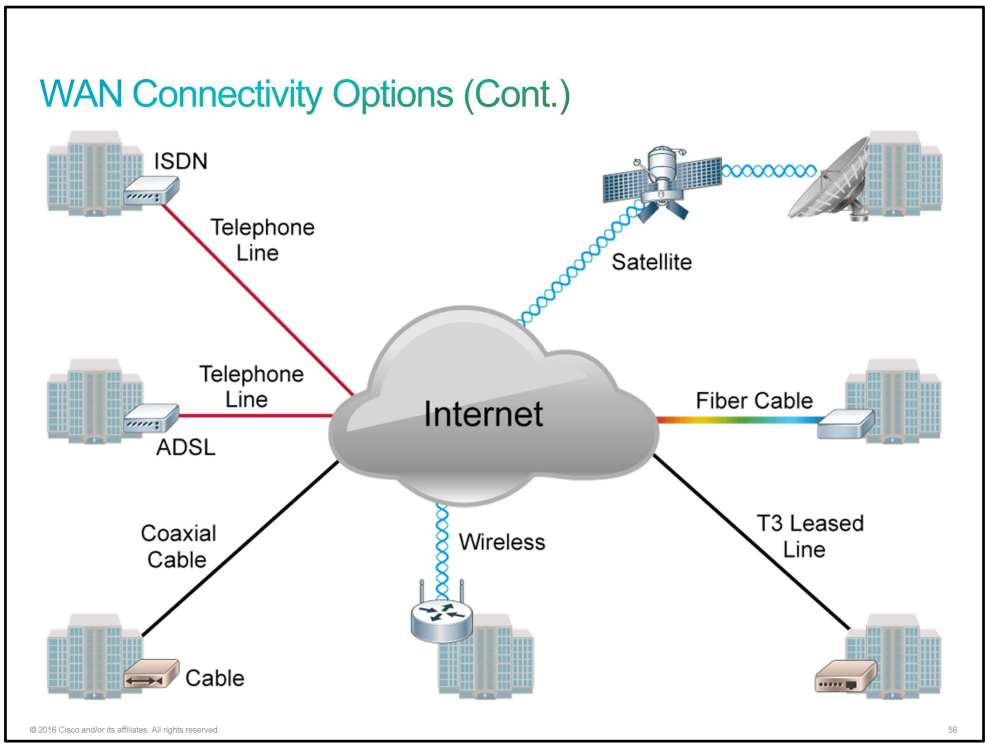


Private WAN connections include dedicated and switched communication link options:

- **Dedicated communication links:** When permanent dedicated connections are required, point-to-point lines are used with various capacities that are limited only by the underlying physical facilities and the willingness of users to pay for these dedicated lines. A point-to-point link provides a pre-established WAN communications path from the customer premises through the provider network to a remote destination. You usually lease point-to-point lines from a carrier, so they are also called leased lines. Leased lines were more popular in the past. Now companies rather use provider-managed VPN or enterprise-managed VPN over Internet. Companies prefer enterprise- or provider-managed VPNs because leased lines are by far the most expensive solution.
- **Switched communication links:** Switched communication links can be either circuit-switched or packet-switched.
  - **Circuit-switched communication links:** Circuit switching dynamically establishes a dedicated virtual connection for voice or data between a sender and a receiver. Before communication can start, the connection through the network of the service provider must be established. Examples of circuit-switched communication links are analog dialup (PSTN) and ISDN.
  - **Packet-switched communication links:** Many WAN users do not make efficient use of the fixed bandwidth that is available with dedicated, switched, or permanent circuits because the data flow fluctuates. Communications providers have data networks that are available to more appropriately service these users. In packet-switched networks, the data is transmitted in labeled frames, cells, or packets. Packet-switched communication links include Frame Relay, ATM, and X.25.

Public connections use the global Internet infrastructure. Now companies use provider-managed VPNs or enterprise-managed VPNs over Internet. Until recently, the Internet was not a viable networking option for many organizations because of the significant security risks and lack of adequate performance guarantees in an end-to-end Internet connection. With the development of the VPN technology, however, the Internet is now an inexpensive and secure option for connecting to teleworkers and remote offices where performance guarantees are not critical. Internet WAN connection links go through broadband services such as DSL, cable modem, and broadband wireless, and they are combined with VPN technologies (for example, DMVPN, GET VPN) to provide privacy across the Internet. Broadband connection options are typically used to connect telecommuting employees to a corporate site over the Internet.

Service providers build networks using different underlying technologies, the most popular being MPLS. Examples of provider-managed VPNs are Layer 3 MPLS VPN and Layer 2 MPLS VPNs (VPWS and VPLS). MPLS is an IETF standard that defines a packet label-based switching technique, which was originally devised to perform fast switching in the core of IP networks. This technique helped carriers and large enterprises scale their networks as increasingly large routing tables become more complex to manage. The industry began using MPLS over a decade ago as a way to allow enterprises to create end-to-end circuits across any type of transport medium using any available WAN technology.

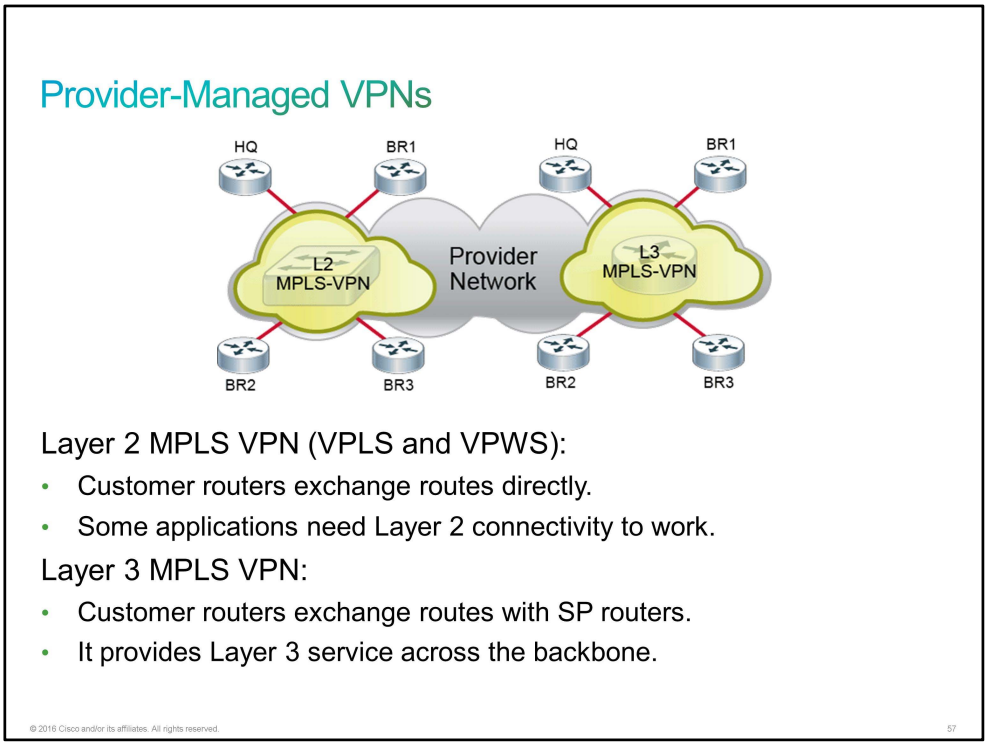


ISPs use several different WAN technologies to connect their subscribers. The connection type that is used on the local loop, or last mile, may not be the same as the WAN connection type that the ISP employs within the ISP network or between various ISPs.

Each of these technologies provides advantages and disadvantages for the customer. Not all technologies are available at all locations. When a service provider receives data, it must forward this data to other remote sites for final delivery to the recipient. These remote sites connect either to the ISP network or pass from ISP to ISP and to the recipient. Long-range communications are usually those connections between ISPs or among branch offices in very large companies.

Provider-Managed VPNs

Provider-managed VPNs can either offer Layer 2 or Layer 3 connectivity. MPLS is a technology that was designed to support efficient forwarding of packets across the network core that is based on a simplified header.



Layer 2 MPLS VPN is useful for customers who run their own Layer 3 infrastructure and require Layer 2 connectivity from the service provider. In this case, the customer manages its own routing information. One advantage that Layer 2 VPN has over its Layer 3 counterpart is that some applications do not work if nodes are not in the same Layer 2 network.

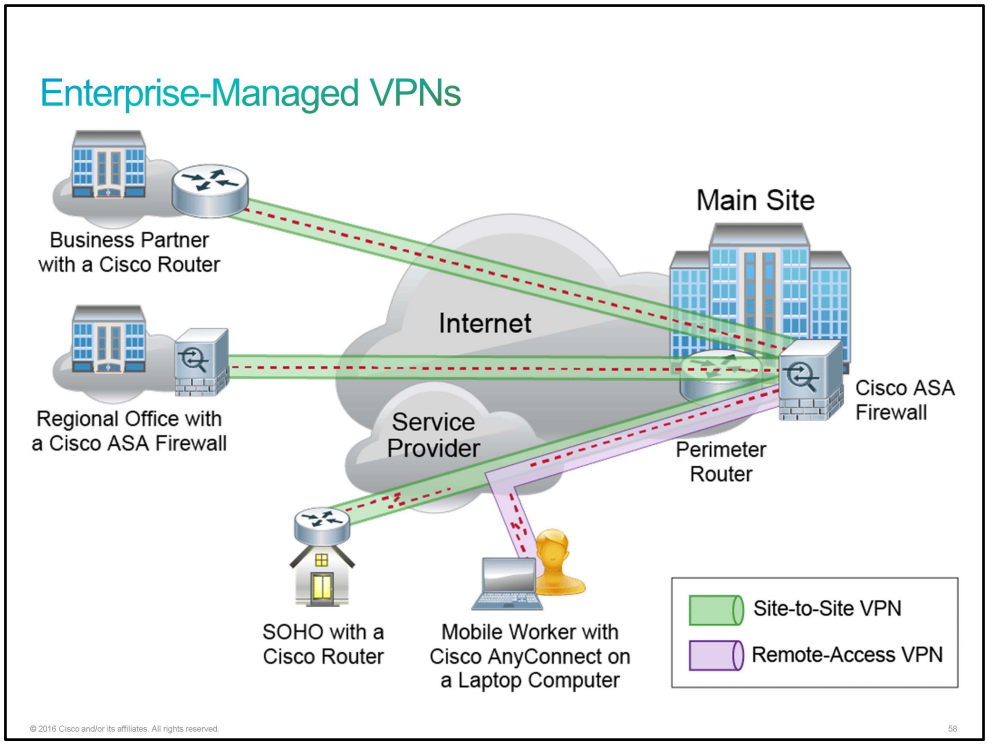
Some typical examples of Layer 2 VPN are VPLS and VPWS. If you look from the customer's perspective, with Layer 2 MPLS VPN, you can imagine a whole service provider network as one big virtual switch.

Layer 3 MPLS VPN provides Layer 3 service across the backbone. A separate IP subnet is used on each customer site. When you deploy a routing protocol over this VPN, the service provider needs to participate in the exchange of routes. Neighbor adjacency is established between your CE router and PE router (which the service provider owns). Within the service provider network, there are many P routers (service provider core routers). The job of P routers is to provide connectivity between PE routers. What this situation means is that the service provider becomes the backbone of your (customer) network.

Layer 3 VPN is appropriate for customers who prefer to outsource their routing to a service provider. The service provider maintains and manages routing for the customer sites. If you look from the customer's perspective, with Layer 3 MPLS VPN, you can imagine whole service provider network as one big virtual router.

Enterprise-Managed VPNs

Organizations need secure, reliable, and cost-effective ways to connect corporate headquarters, branch offices, and teleworkers working in home offices and other remote locations. A VPN is usually a bridge between two private networks. You build that bridge over a public network, typically the Internet. VPN enables headquarters and branch office devices to send and receive data as if they were directly connected.



A VPN is a virtual private network that is constructed within a public network infrastructure, such as the global Internet. VPNs provide an inexpensive alternative to private WAN connections. They are particularly helpful in organizations whose workforce is highly mobile and frequently needs to connect remotely to the corporate network and access sensitive data.

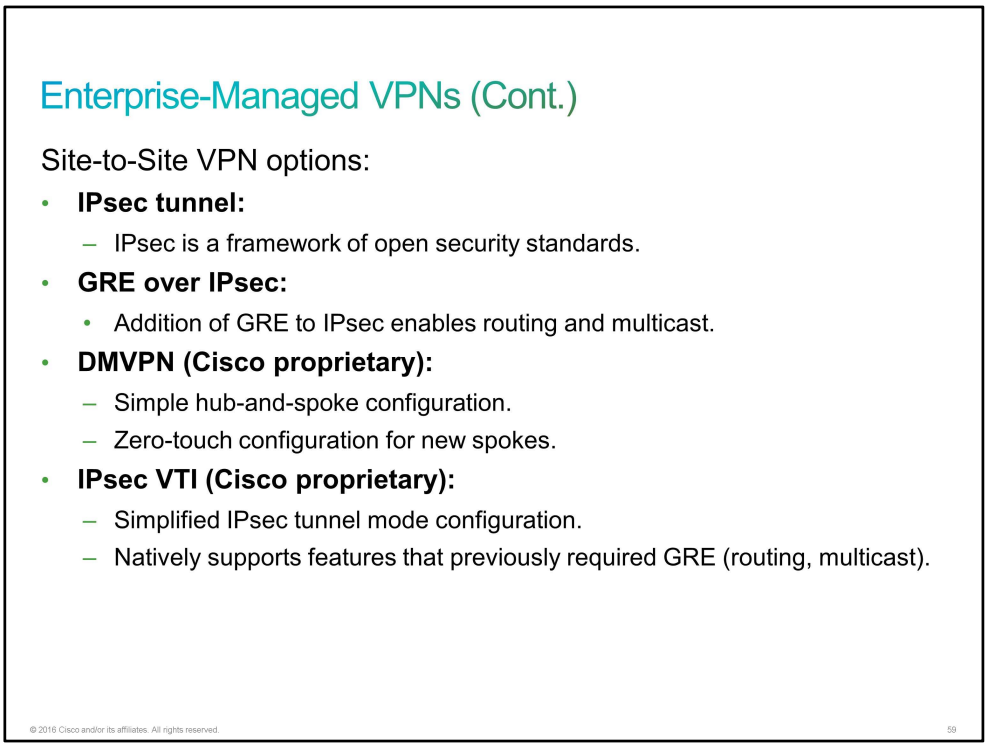
As shown in the figure, there are two types of VPN networks:

- **Site-to-site VPN:** A site-to-site VPN is an extension of a classic WAN network. End hosts send and receive traffic through a VPN device, which could be a router or Cisco Adaptive Security Appliance (Cisco ASA). This device is responsible for encapsulating and encrypting outbound traffic for all traffic from a particular site and sending it through a VPN tunnel over the Internet to a peer VPN device on the target site. Upon receipt, the peer VPN gateway strips the headers, decrypts the content if it was encrypted, and relays the packet toward the target host that is inside its private network. There are many site-to-site VPN options available.
- **Remote-access VPN:** Remote-access VPNs can support the needs of telecommuters, mobile users, and extranet, consumer-to-business traffic. In a remote-access VPN, each host typically has Cisco AnyConnect VPN Client software that is installed. Whenever the host tries to send any traffic, the Cisco AnyConnect VPN Client software encapsulates the traffic before sending it over the Internet to the VPN gateway at the edge of the target network. The VPN client may also encrypt the traffic before sending it over the Internet to the VPN gateway. Upon receipt, the VPN gateway behaves as it does for site-to-site VPNs.

VPNs provide the following benefits:

- **Cost savings:** VPNs enable organizations to use cost-effective, third-party Internet transport to connect remote offices and remote users to the main corporate site. The use of VPNs therefore eliminates expensive, dedicated WAN links. Furthermore, with the advent of cost-effective, high-bandwidth technologies such as DSL, organizations can use VPNs to reduce their connectivity costs while simultaneously increasing remote connection bandwidth.
- **Scalability:** VPNs enable corporations to use the Internet infrastructure, which makes new users easy to add. Therefore, corporations can add large amounts of capacity without adding significant infrastructure. For example, a corporation with an existing VPN between a branch office and the headquarters can securely connect new offices by simply making a few changes to the VPN configuration and ensuring that the new office has an Internet connection. Scalability is a major benefit of VPNs.
- **Compatibility with broadband technology:** VPNs allow mobile workers, telecommuters, and people who want to extend their work day to take advantage of high-speed, broadband connectivity, such as DSL and cable, to gain access to their corporate network. This ability provides workers with significant flexibility and efficiency. Furthermore, high-speed, broadband connections provide a cost-effective solution for connecting remote offices.
- **Security:** VPNs can provide the highest level of security by using advanced encryption and authentication protocols that protect data from unauthorized access. The two available options are IPsec and SSL.

There are many site-to-site VPN options available. However, each option is a little bit different than the other.



IPsec Tunnel

IPsec provides a tunnel mode of operation that enables you to use it as a standalone connection method. This option is the most fundamental IPsec VPN design model. IPsec provides four important security services:

- **Confidentiality (encryption):** The sender can encrypt the packets before transmitting them across a network. By doing so, nobody can eavesdrop on the communication. If another device intercepts the communication, it cannot read it.
- **Data integrity:** The receiver can verify that the data was transmitted through the path without being changed or altered in any way. IPsec ensures data integrity by using checksums, which is a simple redundancy check.
- **Authentication:** Authentication makes sure that the connection is made with the desired communication partner. The receiver can authenticate the source of the packet by guaranteeing and certifying the source of the information. IPsec uses IKE to authenticate users and devices that can carry out communication independently. IKE uses several types of authentication including username and password, one-time password, biometrics, PSKs, and digital certificates.
- **Antireplay protection:** Antireplay protection verifies that each packet is unique and not duplicated. IPsec packets are protected by comparing the sequence number of the received packets with a sliding window on the destination host. A packet that has a sequence number that is before the sliding window is considered either a late or duplicate packet. Late and duplicate packets are dropped.

GRE over IPsec

Although IPsec provides a secure method for tunneling data across an IP network, it has limitations. IPsec does not support IP broadcast or IP multicast, preventing the use of protocols that rely on these features, such as routing protocols. IPsec also does not support the use of the multiprotocol traffic. GRE is a protocol that can be used to "carry" other passenger protocols, such as IP broadcast or IP multicast, and non-IP protocols. Using GRE tunnels with IPsec will give you the ability to run a routing protocol, IP multicast, or multiprotocol traffic across the network between the head end or head ends and branch offices.

With a generic hub-and-spoke topology, you can typically implement static tunnels (typically GRE over IPsec) between the central hub and remote spokes. When you want to add a new spoke to the network, you need to configure it on the hub router. Also, the traffic between spokes has to traverse the hub, where it must exit one tunnel and enter another. Static tunnels may be an appropriate solution for small networks, but this solution becomes unacceptable as the number of spokes grows larger and larger.

Cisco DMVPN

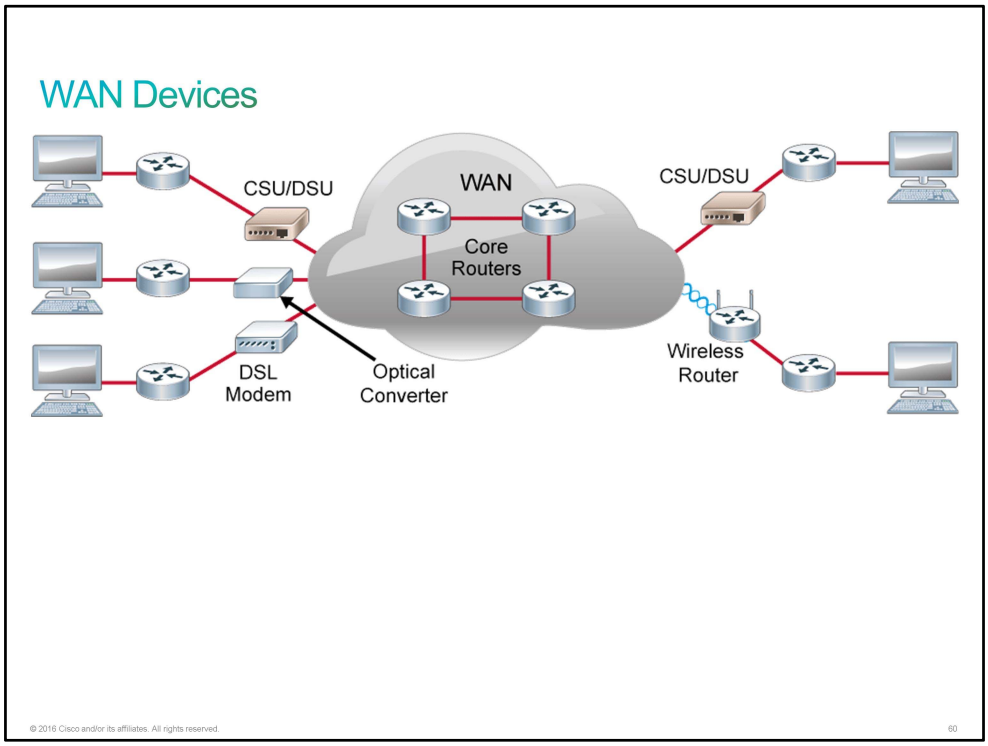
The Cisco Dynamic Multipoint Virtual Private Network (DMVPN) feature enables you to better scale large and small IPsec VPNs. The Cisco DMVPN feature provides simple provisioning of many VPN peers. It also easily supports dynamically addressed spoke routers by its design, if you use an appropriate peer authentication method, such as PKI-enabled peer authentication. The DMPVN feature enables you to configure a single mGRE tunnel interface and a single IPsec profile on the hub router to manage all spoke routers. Thus, the size of the configuration on the hub router remains constant even if you add more spoke routers to the network. DMVPN also allows IPsec to be immediately triggered to create point-to-point GRE tunnels without any IPsec peering configuration.

Cisco IPsec VTI

The VTI mode of an IPsec configuration simplifies a VPN configuration. There are two types of VTI—static and dynamic. With VTI, you implement the IPsec session as an interface. Simple configuration and routing adjacency directly over the virtual interface are great benefits. But keep in mind that all traffic is encrypted and that it supports, like standard IPsec, only one protocol (IPv4 or IPv6). The IPsec tunnel protects the routing protocol and multicast traffic, like with GRE over IPsec. The only difference is that with VTI, you do not need GRE and the overhead that it brings.

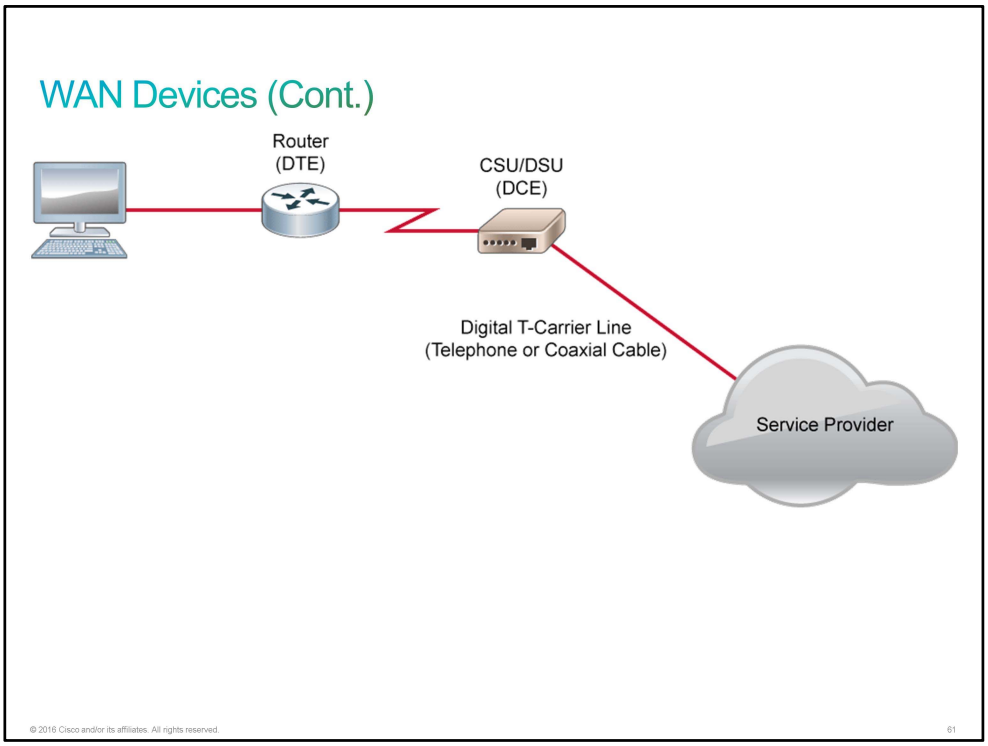
WAN Devices

Several types of devices are specific to WAN environments, including CSU/DSU devices, modems, and certain types of routers and switches.

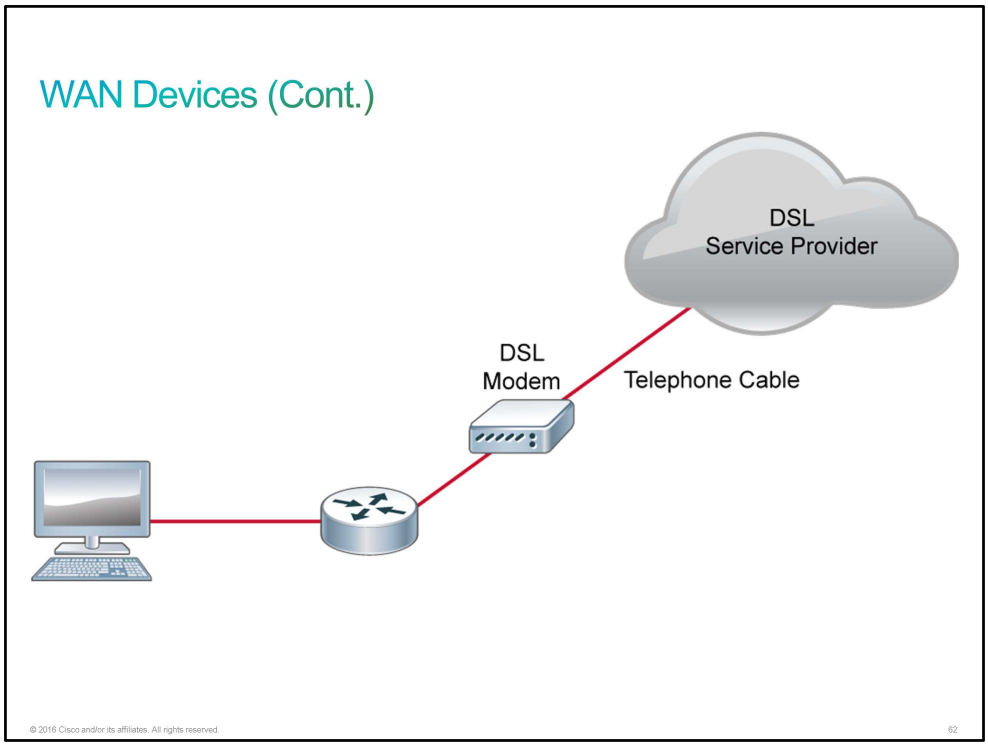


The following are common WAN devices and their descriptions.

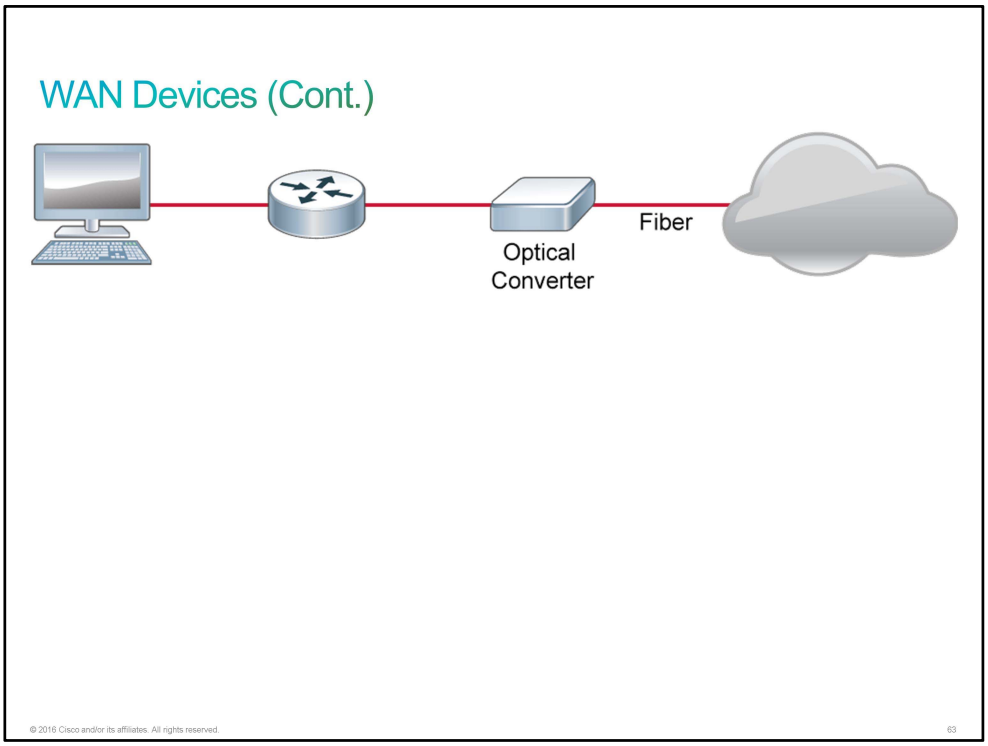
- **Router:** A router provides internetworking and WAN access interface ports that are used to connect to the service provider network. These interfaces may be serial connections or other WAN interfaces. With some types of WAN interfaces, you need an external device such as a CSU/DSU or modem (analog, cable, or DSL) to connect the router to the local POP of the service provider.
- **Core router:** A core router resides within the middle or backbone of the WAN, rather than at its periphery. To fulfill the role of core router, a router must be able to support multiple telecommunications interfaces of the highest speed in use in the WAN core. It must also be able to forward IP packets at wire speed on all these interfaces. The router must support the routing protocols that are being used in the core.
- **CPE:** Devices on subscriber premises are referred to as CPE. A subscriber to a service provider owns the CPE or leases the CPE from the service provider. A copper or fiber cable connects the CPE to the nearest exchange or CO of the service provider. This cabling is often called the local loop or "last mile." CSU/DSU devices, DSL modems, and optical fiber converters are just three of many WAN connection types.



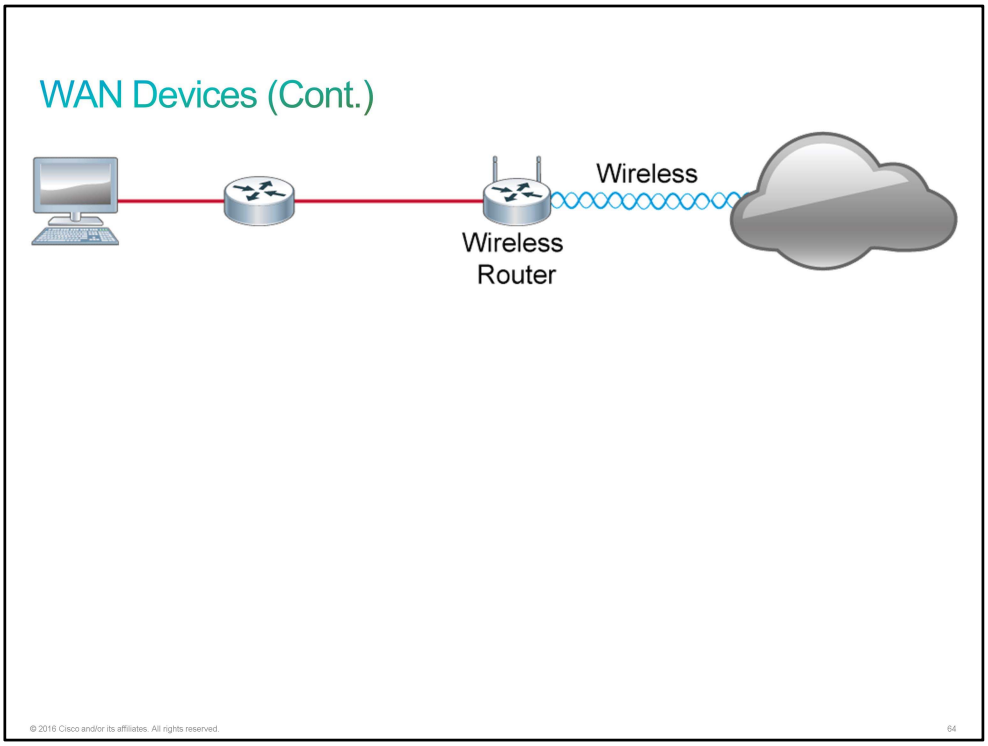
- **CSU/DSU:** A CSU/DSU is a device that is used to connect a DTE to a digital circuit, such as a T1 carrier line. A device is considered DTE if it is either a source or destination for digital data. Examples of DTE include PCs, servers, and routers. In the following figure, the router is considered DTE because it is passing data to the CSU/DSU, which will forward the data to the service provider. Although the CSU/DSU connects to the service provider infrastructure using a telephone or coaxial cable, such as a T1 or E1 line, it connects to the router with a serial cable. A CSU/DSU is actually two devices in one box. The CSU provides termination for the digital signal and ensures connection integrity through error correction and line monitoring. The DSU converts the T-carrier line frames into frames that the LAN can interpret and vice versa. You can also implement a CSU/DSU as a module within a router, in which case, a serial cable is not necessary. A CSU/DSU is sometimes referred to as a DCE because it provides a path for communication. DCE is a more general label for devices that provide interfaces for DTE into communication links on the WAN cloud. When the links are digital, the DCE is a CSU/DSU. When analog telephone lines are the communication media, the DCE is a modem.



- **Modem:** A modem is a device that interprets digital and analog signals, enabling data to be transmitted over voice-grade telephone lines. At the source, digital signals are converted to a form that is suitable for transmission over analog communication facilities. At the destination, these analog signals are returned to their digital form. There are various types of modems. In the following figure, a DSL modem (which is used in DSL broadband environments) is connected to a router with an Ethernet cable and is connected to the service provider network with a telephone cable. You can also implement a modem as a router module.



- **Optical fiber converters**: Optical fiber converters are used where a fiber-optic link terminates to convert optical signals into electrical signals and vice versa. You can also implement the converter as a router or switch module.



- **Wireless router**: Wireless routers are used when you are using wireless medium for WAN connectivity. You can also use an access point instead of wireless router.

Challenge

1. Which two statements about WANs are true? (Choose two.)

   A. WANs generally connect devices that are located over a broader geographical area.
   B. WANs generally connect devices that are close to each other.
   C. WAN stands for World Around Networks.
   D. WANs use connections of various types to provide access to bandwidth over large geographical areas.

2. Which WAN topology option provides the highest level of redundancy?

   A. hub-and-spoke
   B. partially meshed
   C. fully meshed
   D. point-to-point

3. Which two VPNs are examples of service provider-managed VPNs? (Choose two.)

   A. remote-access VPNs
   B. Layer 2 MPLS VPN
   C. Layer 3 MPLS VPN
   D. DMVPN

4. Which two technologies are examples of Layer 2 MPLS VPN technologies? (Choose two.)

   A. VPLS
   B. DMVPM
   C. GETVPN
   D. VPWS

5. Which protocol should be used with IPsec to give you the ability to run a routing protocol or IP multicast across the network between two site-to-site VPN peers?

   A. GRE
   B. IPsec tunnel
   C. WAN
   D. MPLS

6. Which protocol provides confidentiality, data integrity, authentication, and antireplay protection?

   A. GRE
   B. IPsec
   C. ISDN
   D. MPLS

7. Which service ensures that data being transmitted has not been changed or altered in any way?

   A. confidentiality
   B. data integrity
   C. authentication
   D. antireplay protection

Answer Key

Challenge

1. A, D
2. C
3. B, C
4. A, D
5. A
6. B
7. B